

*RobotScene:  
simulación/explotación de  
escenarios robóticos con  
capacidad multirobot*

**MANUAL DE USUARIO**



Escuela de  
Ingeniería y Arquitectura  
Universidad Zaragoza



Universidad  
Zaragoza

1542



El presente documento se corresponde con el manual de instrucciones de la versión multi-robot de RobotScene. Este documento esta basado en el manual de instrucciones de la versión mono-robot de RobotScene, que fue desarrollado por *Javier Fermín Hernández*. La actualización del antiguo manual de instrucciones para adecuarlo a las nuevas capacidades multi-robot de RobotScene ha sido desarrollada por *José Javier Pérez Rubio*.



## ÍNDICE DEL MANUAL

1	INTRODUCCIÓN.....	4
1.1	INTRODUCCIÓN A ROBOTSCENE .....	4
1.2	REQUISITOS MÍNIMOS .....	4
1.3	INSTALACIÓN .....	5
1.4	DESINSTALACIÓN.....	5
2	GENERALIDADES .....	6
3	EL CONSTRUCTOR DE OBJETOS.....	9
3.1	DESCRIPCIÓN DE LA INTERFAZ DE USUARIO .....	9
3.2	REFERENCIAS ABSOLUTAS Y RELATIVAS.....	12
3.3	MODOS DE OPERACIÓN E INTERACCIÓN CON EL TRIODO UNIVERSAL.....	13
3.4	CREACIÓN Y MODIFICACIÓN DE OBJETOS. TIPOS DE OBJETOS.....	18
3.5	FORMAS MÁS COMPLEJAS.....	23
3.6	MATERIALES .....	30
4	EL CONSTRUCTOR DE ROBOTS.....	36
4.1	DESCRIPCIÓN DE LA INTERFAZ DE USUARIO .....	36
4.2	METODOLOGÍA DE CREACIÓN DE UN ROBOT.....	38
4.3	IMPLEMENTACIÓN DE LA CINEMÁTICA INVERSA.....	42
5	EL CONSTRUCTOR DE ESCENARIOS.....	54
5.1	DESCRIPCIÓN DE LA INTERFAZ DE USUARIO .....	54
5.2	CREACIÓN DEL ESCENARIO .....	60
5.3	EXPLOTACIÓN DEL ESCENARIO.....	64
6	DESCRIPCIÓN DEL LENGUAJE DE PROGRAMACIÓN DE ROBOTS	78



## **1 INTRODUCCIÓN**

### **1.1 INTRODUCCIÓN A ROBOTSCENE**

Mediante la aplicación RobotScene el usuario será capaz de crear de una forma visual, rápida y muy intuitiva escenarios tridimensionales virtuales con robots industriales. Posteriormente se permitirá al usuario controlar a dichos robots de forma individual o simultánea, y simular aplicaciones reales, en las que los robots interactuarán con el entorno.

Para el correcto uso de RobotScene no es necesario que el usuario posea conocimientos de diseño gráfico en tres dimensiones, de hecho, el usuario comprobará que mediante la efectiva interfaz de usuario podrá crear objetos tridimensionales sin apenas esfuerzo y viendo en todo momento el resultado de las creaciones.

Sin embargo, sí que serán necesarios conocimientos generales de programación para poder aprovechar todas las características que ofrece esta aplicación, de cara a poder implementar el modelo inverso del robot teniendo en cuenta posibles configuraciones singulares que de otra forma pasarían desapercibidas.

También serán indudablemente necesarios ciertos conocimientos de robótica industrial, y en este sentido RobotScene resulta ideal para la docencia de dicha materia, puesto que el hecho de ir construyendo interactivamente un robot y tener acceso a todas sus variables puede ir afianzando muchos conceptos robóticos que serían más difíciles de ver exclusivamente sobre el papel. Y una vez construido nuestro robot se procede a su control, lo cual resulta de suma utilidad tanto para estudiantes de robótica como para profesionales de la materia que quieran ver sus robots simulados y en acción.

A diferencia de otras aplicaciones similares, en RobotScene el usuario podrá crear sin apenas esfuerzo cualquier robot industrial que desee, sin ninguna limitación en cuanto a su morfología o número de articulaciones, viendo siempre los resultados en la pantalla, y tan pronto como lo haya generado introducirlo en un entorno industrial virtual y proceder a su explotación de una manera interactiva mediante la inclusión de un programa robótico. Los escenarios robóticos permitirán la inclusión de varios robots en la escena, cada robot tendrá la capacidad de ejecutar su programa robótico de forma simultánea con otros robots.

### **1.2 REQUISITOS MÍNIMOS**

Debido al uso que se hace de los gráficos tridimensionales se exige que el equipo en el que va a ser ejecutado RobotScene cumpla unos requisitos mínimos, no excesivamente exigentes en los tiempos actuales, pero sí que se habrán de cumplir ciertas condiciones mínimas, a saber:

- Procesador Intel a 1,5 GHz o superior.



- Sistema operativo compatible: Windows 2000, Windows 98, Windows Me, Windows XP, Windows Vista o Windows 7.
- Tarjeta gráfica con controladores compatibles con OpenGL.

### **1.3 INSTALACIÓN**

Introduzca en su lector de CD-ROM el disco de RobotScene y ejecute el archivo setup.exe para proceder a la instalación de la aplicación en su ordenador.

Se ejecutará un instalador con la interfaz típica de un asistente, el cual le irá guiando por todas las posibles opciones de instalación, en el cual usted tendrá que elegir el directorio donde instalará la aplicación y se le pedirá si desea incluirlo en el menú inicio de windows y si desea un acceso directo en su escritorio.

### **1.4 DESINSTALACIÓN**

Para desinstalar RobotScene de su equipo pulse en la opción de desinstalar dentro del menú inicio de RobotScene, o bien mediante la opción de Windows, Panel de control, Agregar o quitar programas.



## 2 GENERALIDADES

Debido al gran número de acciones que se pueden realizar se ha habilitado una estructura con tres módulos, el módulo constructor de objetos, el módulo constructor de robots y el módulo constructor-explotador de escenarios.

Si bien cada uno de los módulos posee diferentes funcionalidades, hay ciertos aspectos básicos de la interfaz de usuario que son comunes a los tres módulos que componen RobotScene, tal y como se puede ver en la siguiente captura:

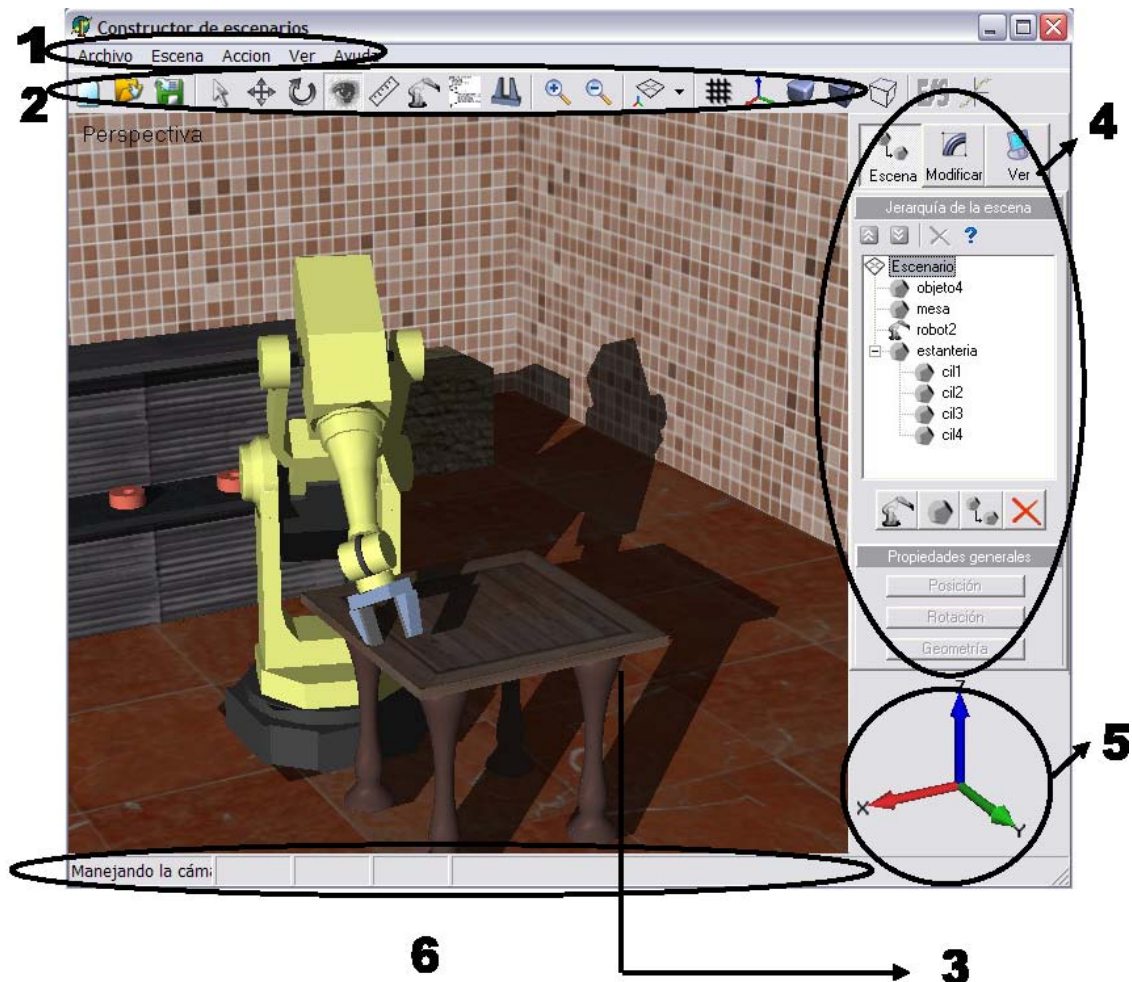


Figura 1. Interfaz de usuario del constructor de escenarios.

1. El menú principal. Se encuentra en la parte superior de la pantalla. En este menú se pueden encontrar menús típicos de cualquier aplicación, como Archivo, edición o Ayuda. Desde este menú se puede acceder a casi todas las funcionalidades del módulo y obviamente las opciones accesibles dependen del módulo en que nos encontremos.
2. Barra de Herramientas. Se encuentra justo debajo del menú principal y consiste en una serie de botones gráficos con las opciones más habituales del programa agrupadas por tipos (Salvaguarda de la información, acciones a realizar, visualización, etc).
3. Visor tridimensional. En este visor se realiza la representación gráfica en



- tres dimensiones. Existen varios tipos de representaciones que se podrán elegir, que serán vista en perspectiva o las correspondientes vistas de proyección ortogonal. Tal y como se explicara posteriormente, se puede actuar interactivamente en este visor mediante el ratón y se pueden ajustar una serie de opciones, como el color de fondo o la existencia o no de rejilla.
4. Panel de Control lateral. Este panel es fuertemente dependiente del módulo en el que nos encontremos, pero a través de dicho panel tendremos acceso a las opciones propias de creación y/o modificación de nuestro sólido – robot -escenario virtual, así como de la visualización.
  5. Triedro del espacio universal. Este ingenioso ‘icono’ tiene una doble funcionalidad, por un lado sirve para representar la orientación de los ejes principales de nuestra escena (el sistema de referencia al cual se refieren el resto de elementos virtuales) y por otro lado es utilizado para seleccionar direcciones, planos de movimiento y ejes de giro de una manera ciertamente intuitiva.
  6. Barra de estado. Aquí se nos muestra información relevante a lo que estamos construyendo o controlando. Dependerá fuertemente del módulo en el que estemos pero su finalidad es la misma, la salida por pantalla de información relativa a la acción que estamos realizando.

Tal y como se ha comentado existen tres módulos, cada uno de ellos con una funcionalidad diferente. El primer módulo (Constructor de Objetos) permite realizar el diseño de un sólido, independientemente del uso que se le vaya a dar. En este sentido, el sólido creado podrá formar parte de un robot o bien ser un elemento del escenario. Existen multitud de programas en el mercado que sirven para modelar objetos en tres dimensiones (3dStudio, Blender, Wings3D), aunque no es necesario haber tenido contacto con ellos, el usuario que los conozca encontrará ciertos aspectos bastante familiares. Si bien es cierto que nuestro módulo constructor de objetos afronta la creación de una manera mucho más básica, de manera que no hace falta tener extensos conocimientos de modelado tridimensional para utilizar correctamente este módulo y obtener resultados ciertamente vistosos. Para crear objetos se pueden emplear formas simples ya definidas como cajas, esferas o cilindros, o bien se permite al usuario la creación de formas con herramientas de modelado, como puede ser la generación de volúmenes a través del movimiento de una superficie dibujada por el usuario (extrusión de una superficie, volumen de revolución, etc). Otra parte de este módulo corresponde a la creación de materiales que recubrirán los objetos creados para propiciarles un aspecto más cercano a la realidad, definiendo colores, texturas, brillos o transparencias.

El segundo módulo lo constituye el Constructor de Robots, en el cual, como su nombre indica, se procederá al ensamblado de un robot virtual. Para ello, haremos uso de los sólidos modelados anteriormente, que representarán la geometría (la representación gráfica, la forma) de cada uno de los sólidos que componen la cadena cinemática que es el robot. Para la resolución del modelo cinemático directo del robot se ha utilizado la representación de Denavit Hartenberg (DH), que es la forma habitual de representación en robótica, de manera que el usuario deberá asignar un sistema de referencia a cada sólido de la cadena de acuerdo con el protocolo de asignación de Denavit Hartenberg (esto deberá ser tenido en cuenta a la hora de construir el sólido en el constructor de objetos), con lo que mediante la introducción de los cuatro parámetros de DH ( $\theta$ ,  $a$ ,  $\alpha$ ,  $d$ ) correspondientes a cada



articulación en una tabla quedará definida la geometría del robot y la relación entre sólidos, junto con parámetros adicionales como podrán ser el rango articular, el tipo de articulación o la velocidad y aceleración máximas.

Pero la construcción de un robot por medio de este módulo va más allá del mero ensamblado de sólidos. Con el fin de dotar al robot de una funcionalidad plena, deberá especificarse la cinemática inversa del mismo por medio de un script (código fuente que interpretaremos) específicamente creado para cada robot, por lo que se deberán obtener previamente las ecuaciones correspondientes para posteriormente proceder a su implementación programada. Todo ello exigirá al usuario unos conocimientos mínimos de programación.

También se posibilita en este módulo la creación de herramientas para el robot de una manera interactiva bastante gráfica.

Por último, a través del módulo Constructor de Escenarios procederemos a crear nuestro escenario virtual colocando nuestros robots en un entorno virtual. A continuación procederemos a su explotación, individual o simultánea, bien mediante un control que emula una botonera de guiado o bien programando directamente nuestro robot mediante un lenguaje robótico creado para el caso. La comunicación entre robots se realizará mediante una ventana interactiva que emula un cableado real en un entorno industrial.





### 3 EL CONSTRUCTOR DE OBJETOS

Como ya se ha explicado anteriormente en este módulo se realiza una construcción geométrica de un sólido, independientemente del uso que se le vaya a dar a dicho sólido, y una vez modelado el sólido se procederá a aplicarle un material mediante el editor de materiales incorporado. Así que el proceso necesario será el siguiente:

Primero modelar el sólido con las herramientas proporcionadas y después aplicarle un material que lo recubrirá propiciandno un aspecto más cercano a la realidad, definiendo colores, texturas, brillos o transparencias.

#### 3.1 DESCRIPCIÓN DE LA INTERFAZ DE USUARIO

Si bien ya ha sido descrita antes de forma genérica, veamos hagamos ahora una descripción exhaustiva de la interfaz que muestra este módulo y de sus características específicas con respecto a los siguientes módulos.

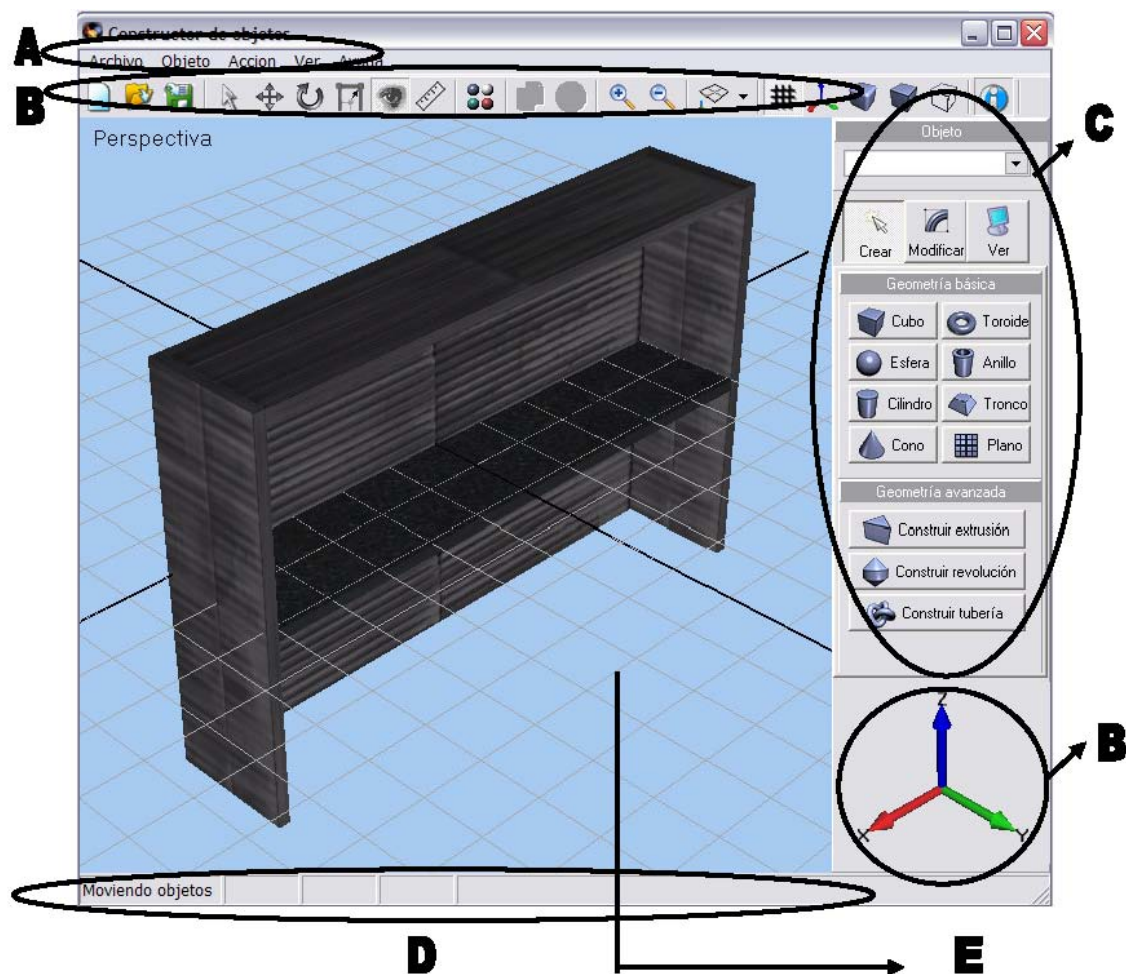


Figura 2. Intefaz de usuario del constructor de objetos



## A. Menú principal

Desde este menú se puede acceder a casi todas las funcionalidades del programa, contiene los siguientes elementos:

Archivo: Mediante este menú usted podrá guardar el sólido, abrir uno existente, empezar uno nuevo o bien abandonar la aplicación (estos conceptos serán explicados con más profundidad posteriormente)

Objeto: Mediante este menú usted podrá crear una nueva forma básica, o bien copiar o eliminar un objeto.

Acción: Desde este menú se pueden seleccionar todos los modos de operación posibles, así como activar el editor de materiales.

Ver: Desde este menú se tiene acceso a todas las opciones referentes a la visualización de la escena en tres dimensiones.

## B. Barra de herramientas

Se encuentra justo debajo del menú principal y consiste en una serie de botones gráficos con las opciones más habituales del programa agrupadas por tipos:



Los tres de la izquierda permiten crear un nuevo sólido, abrir uno existente o guardar el actual, respectivamente.

Los seis siguientes dan acceso a los diferentes modos de operación.

El siguiente icono activa el editor de materiales.

Los dos siguientes permiten copiar o eliminar el elemento seleccionado.

Los dos siguientes permiten alejar o acercar la cámara.

El siguiente permite seleccionar el tipo de vista.

El siguiente permite que se vean los ejes (el sistema de referencia) del objeto seleccionado.

Los siguientes seleccionan el tipo de representación de los objetos.



### C. Panel de control lateral

Desde este panel podremos crear elementos para nuestro sólido, aplicar modificadores a dichos elemento o bien controlar la representación visual.

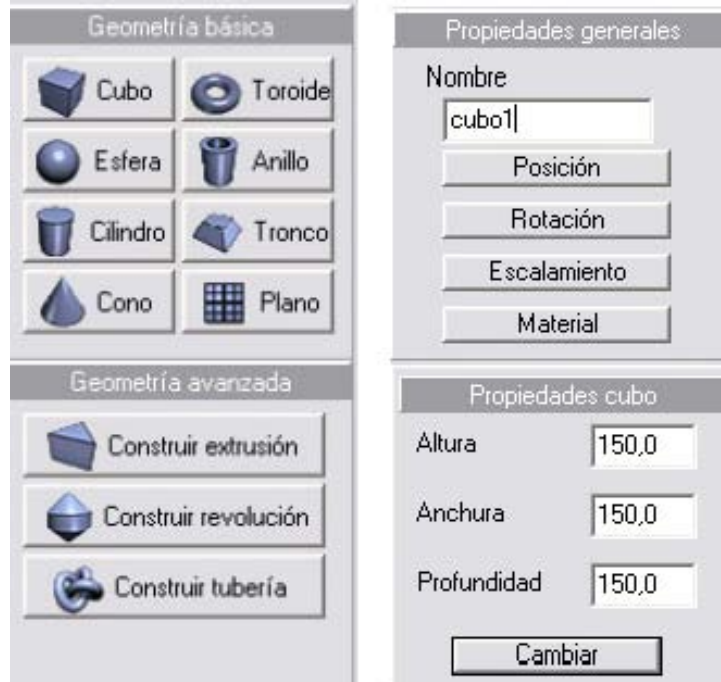
Existen tres zonas diferenciadas:

Lista de objetos: A través de esta lista desplegable tenemos acceso a todos los objetos que componen nuestro sólido, pudiendo seleccionar un objeto mediante el nombre.

Botones de selección de panel: A través de estos botones se selecciona el tipo de panel de control.

Panel de control: Según se haya seleccionado con los botones superiores puede haber tres paneles: el panel de creación, el panel de modificación o el panel de visualización.

En el panel de creación tendremos acceso a todas las herramientas para la creación de objetos. Tenemos dos partes, en la mitad superior podremos crear geometría básica, que son formas ya predefinidas, como el cubo o la esfera, mientras que en la mitad inferior crearemos geometría más compleja, ya sea mediante extrusión, mediante revolución o el sólido complejo que hemos pasado a llamar tubería.



En el panel de modificación tenemos todas las herramientas para proceder a la modificación de las propiedades de los objetos. Tenemos dos partes. En la mitad superior tenemos acceso a las propiedades generales de los objetos, comunes a todos ellos, como son la posición, la orientación, la escala o el material. En la mitad inferior tenemos acceso a las propiedades específicas del objeto seleccionado, que



como su nombre indica dependerán del tipo de objeto, como puede ser la anchura o altura en caso de una caja, el radio en caso de una esfera, etc. Este panel sólo estará visible cuando haya un objeto seleccionado y su forma cambiará con el tipo de objeto.

#### **D. Barra de estado**

En esta barra tenemos información siempre visible referente a la acción que estamos realizando. En el panel izquierdo tenemos el modo de operación en que nos encontramos, mientras que en los siguientes tenemos información referente al objeto seleccionado, como pueden ser sus coordenadas espaciales, su orientación o su escala.

#### **E. Zona de visualización en 3 dimensiones**

En esta parte de la pantalla se procede a la representación tridimensional de la escena que está siendo creada. Aparte de la escena propiamente dicha podemos observar la cuadrícula o rejilla y los ejes principales, representados por dos líneas más gruesas que se cruzan en el origen. La cuadrícula está formada por tres planos que dividen el espacio siguiendo los ejes de coordenadas. En la vista en perspectiva y en la cenital, la rejilla representa al plano horizontal, el plano XY, mientras que en resto de vistas se puede ver el plano correspondiente. La rejilla sirve como referencia para los ajustes a la hora de crear objetos y construir la escena, y sus divisiones son cada 10 cm., así que inicialmente la rejilla representa un espacio cuadrado de 2m de lado. La orientación y el sentido de los ejes principales vendrá siempre indicado mediante el triedro del espacio universal, que será explicado más adelante.

### **3.2 REFERENCIAS ABSOLUTAS Y RELATIVAS**

Como ya se ha explicado varias veces, la finalidad de este módulo es la construcción geométrica de un sólido, que posteriormente podremos utilizar para varios fines.

Para definir la posición de los objetos utilizamos un sistema cartesiano de referencia, cuyo origen se encuentra en la intersección de los ejes principales de la rejilla y cuya orientación viene siempre representada por el triedro del espacio universal. Ahora bien, como sabemos estamos construyendo un sólido, pero lo estamos construyendo mediante una serie de objetos elementales, así que para cada objeto elemental será necesario definir aparte de la posición, la orientación con respecto al sistema de referencia absoluto. Cada objeto lleva asociado un sistema de referencia propio para poder estudiar la relación espacial entre ambos sistemas, que vendrá dada por la posición y orientación del sistema asociado al objeto con respecto al sistema de referencia.

Así pues, si para la posición utilizamos tres parámetros, para la representación de la orientación utilizamos una notación compacta que nos permita definir la orientación con sólo tres parámetros, utilizamos los ángulos Roll-Pitch-Yaw, que representan tres rotaciones, la primera, el ángulo Yaw en torno al eje X absoluto, la segunda un ángulo Pitch en torno al eje Y absoluto y la última un ángulo Roll, en torno al eje Z absoluto.

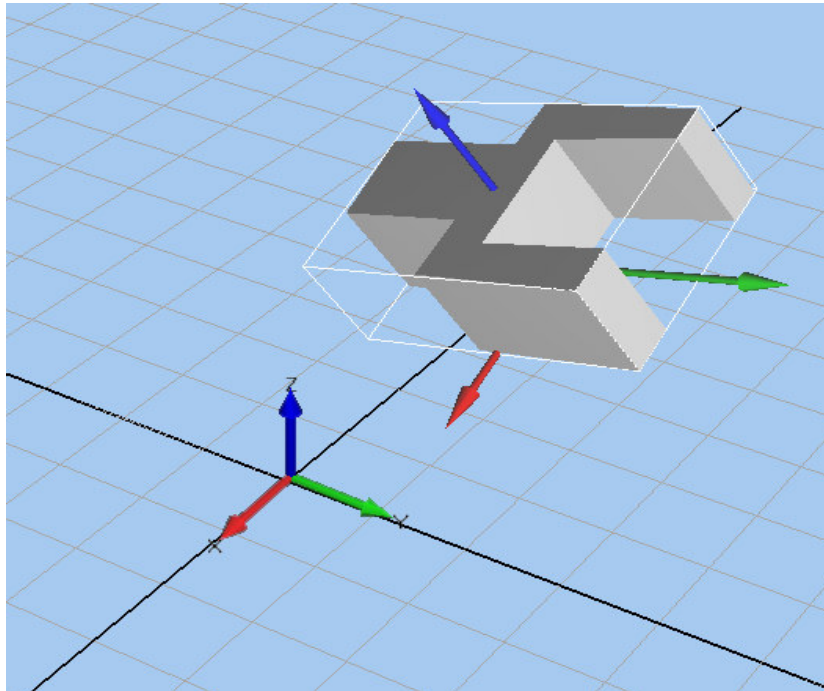


Figura 3. Referencia absoluta y referencia relativa de un objeto

De esta forma el sólido que hayamos construido constará de una serie de objetos elementales, cada uno con su propio sistema de referencia, pero de cara a módulos siguientes, el sólido construido será considerado como un solo sólido, independientemente de cómo esté constituido, y como tal deberá llevar asociado un sistema de referencia. Así pues, el sistema de referencia que nos ha servido para construir el sólido es el sistema de referencia asociado a dicho sólido, y así será usado de aquí en adelante.

### 3.3 MODOS DE OPERACIÓN E INTERACCIÓN CON EL TRIEDO UNIVERSAL

En este módulo tenemos seis modos de operación distintos, siendo un modo de operación el comportamiento que tiene la escena tridimensional frente a entradas provenientes del ratón. Se puede acceder a estos modos bien mediante el menú principal (Menú > Acción) o bien mediante la barra de herramientas.

Ha sido comentada antes la importancia del triedro universal, y ahora veremos su funcionamiento, puesto que cambia su forma y su funcionalidad dependiendo del modo de operación en que nos encontremos.

#### **Modo de selección de objetos.**

Se puede seleccionar este modo mediante Menú principal>Acción>Seleccionar Objetos o bien mediante el icono correspondiente de la barra de herramientas.

Lo primero que veremos al seleccionar este modo de operación es que la forma del triedro pasa a contener sólo los tres ejes principales (en caso de estar en vista tridimensional), que nos indicarán la orientación de los ejes del sistema de referencia absoluto en nuestra escena.

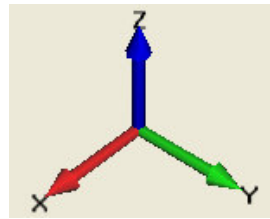


Figura 4. Forma que adopta el triedro en el modo de selección de objetos.

Cuando nos encontramos en este modo de operación seleccionaremos el objeto que queramos poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón. En ese momento el objeto que hayamos seleccionado quedará remarcado por un paralelepípedo de aristas blancas, lo que se conoce más genéricamente como 'bounding box'. Al seleccionar un objeto su nombre aparecerá en la lista desplegable situada en la parte superior del panel lateral. A través de esta lista desplegable, tal y como ya se ha comentado, también se puede proceder a la selección de un objeto en todo momento mediante su nombre, aunque no estemos en el modo de selección de objetos.

Al seleccionar un objeto, se habilitará el panel de propiedades generales, por lo que podremos acceder a ellas, y también aparecerá el panel de propiedades específicas correspondientes al tipo de objeto. Mientras exista un elemento seleccionado se podrá acceder a sus características mediante ambos paneles de control (situados en el panel lateral de Modificar).

Por último, al seleccionar un objeto tendremos también acceso a dos acciones nuevas, copiar (barra de herramientas > botón *copiar objeto* o menú principal > objeto > copiar objeto) lo que nos creará una copia del objeto seleccionado y eliminar objeto (barra de herramientas > botón *eliminar objeto* o bien Menú principal > Objeto > Eliminar objeto), con lo que eliminaremos el objeto de la escena.

### Modo de traslación de objetos.

Se puede seleccionar este modo mediante Menú principal>Acción>Trasladar Objetos o bien mediante el icono correspondiente de la barra de herramientas.

Lo primero que veremos al seleccionar este modo de operación es que la forma del triedro pasa a tener la forma siguiente:

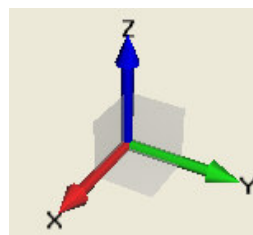


Figura 5. Forma que adopta el triedro en el modo de traslación de objetos

Aparte de los tres ejes principales han aparecido unos planos, que utilizaremos para seleccionar el plano de movimiento. Los ejes siguen indicando la orientación,



como anteriormente, pero ahora se utilizarán de forma interactiva.

Cuando nos encontramos en este modo de operación seleccionaremos el objeto que queramos poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón, de la misma forma que hacemos cuando estamos en el modo de selección de objetos, con la diferencia de que mientras tengamos el botón del ratón apretado podremos arrastrar nuestro objeto por el escenario.

Para trasladar un objeto deberemos indicar bien el plano por el que queremos que se mueva o bien la dirección de movimiento. Esta selección se realizará pulsando con el ratón en el triedro universal, así, si pulsamos sobre el eje X del triedro éste quedará resaltado, indicando que todos los objetos que ahora arrastremos, serán trasladados en el eje X, si ahora pulsamos sobre un objeto con el ratón y arrastramos comprobaremos que efectivamente con el ratón lo movemos en la dirección del eje X. Lo mismo pasa con el resto de los ejes y con los planos.

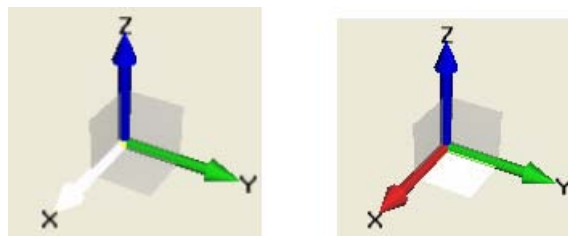


Figura 6. Selección de movimiento a lo largo del eje X (izquierda) y en el plano XY (derecha)

Cuando nos encontremos en un visor ortogonal hay direcciones de movimiento que pierden sentido, así que son deshabilitadas, así, por ejemplo, si nos situamos en el visor cenital (Visor XY), el movimiento a través del eje Z queda deshabilitado, puesto que en esta vista el eje Z es perpendicular al plano de proyección.

También se puede especificar la posición del objeto numéricamente mediante el panel lateral de propiedades generales (Panel lateral> Panel Modificar>Propiedades Generales>Posición).

### Modo de rotación de objetos.

Se puede seleccionar este modo mediante Menú principal>Acción>Rotar Objetos o bien mediante el icono correspondiente de la barra de herramientas.

Lo primero que veremos al seleccionar este modo de operación es que la forma del triedro pasa a tener la forma siguiente:



Figura 7. Forma que adopta el triedro en el modo de rotación de objetos

Aparte de los tres ejes principales han aparecido unos anillos alrededor de cada eje. Los ejes siguen indicando la orientación, como siempre, pero ahora se



usarán en conjunción con los anillos seleccionar el eje de rotación. La rotación de los objetos se hará en torno a ejes absolutos, La selección del eje de rotación se hará de una manera similar a como se hace a la hora de trasladar objetos, se pulsa con el ratón en el anillo del eje que queremos, quedando este remarcado y dicho eje pasará a ser el eje de rotación actual.

Cuando nos encontramos en este modo de operación seleccionaremos el objeto que queramos poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón, de la misma forma que hacemos cuando estamos en el modo de selección de objetos, con la diferencia de que mientras tengamos el botón del ratón apretado podremos rotar el objeto seleccionado moviendo el ratón hacia arriba o hacia abajo (rotaciones positivas y negativas, respectivamente).

De igual modo que antes, al pasar a vistas ortogonales hay giros que pierden sentido dependiendo de la vista, así, para cada tipo de vista ortogonal sólo se activa un eje de giro, el perpendicular al plano de la vista. Así, por ejemplo, al situarnos en vista cenital (Vista del plano XY), sólo podremos girar los objetos a través del eje Z.

También se puede especificar la orientación del objeto numéricamente mediante el panel lateral de propiedades generales (Panel lateral> Panel Modificar), en este caso la rotación vendrá definida por tres parámetros, roll, pitch y yaw, que explicaremos posteriormente.

### **Modo de escalado de objetos.**

Se puede seleccionar este modo mediante Menú principal>Acción>Escalar Objetos o bien mediante el icono correspondiente de la barra de herramientas.

Lo primero que veremos al seleccionar este modo de operación es que la forma del triedro pasa a contener sólo los tres ejes principales (en caso de estar en vista tridimensional), que nos indicarán la orientación de los ejes del sistema de referencia absoluto en nuestra escena.

Cuando nos encontramos en este modo de operación seleccionaremos el objeto que queramos poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón, de la misma forma que hemos venido haciendo hasta ahora, con la diferencia de que mientras tengamos el botón del ratón apretado podremos escalar uniformemente el objeto seleccionado moviendo el ratón hacia arriba, para incrementar su tamaño o hacia abajo para hacerlo más pequeño.

También se puede modificar el factor de escalamiento de forma numérica mediante el panel lateral de propiedades generales (Panel lateral> Panel Modificar) y además de esta forma se podrá aplicar una escala no uniforme al objeto seleccionado, siendo en este caso cada factor referido a los ejes del objeto, no a los ejes absolutos del sistema referencia absoluto como se había venido haciendo hasta ahora.

### **Modo de manejo de la cámara.**





Se puede seleccionar este modo mediante Menú principal>Acción>Manejar la cámara o bien mediante el icono correspondiente de la barra de herramientas.

Mediante este modo de operación no modificamos la escena, sino que mediante el uso del ratón pasamos a tener el control de la cámara.

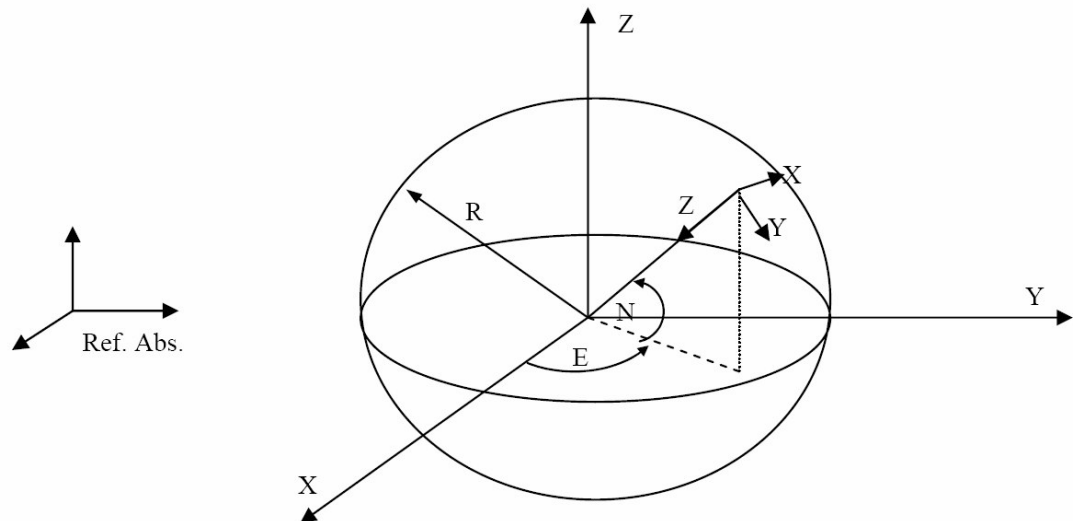


Figura 8. Determinación del punto de vista

Si estamos en la vista tridimensional, si pulsamos el botón izquierdo del ratón pasamos a mover la posición de la cámara, pero no del punto al que estamos mirando, seguimos teniendo el mismo objetivo. Si movemos el ratón hacia arriba o hacia abajo subiremos y bajaremos el punto de vista respectivamente, modificando en parámetro Norte (N en la figura) mientras que moviendo el ratón hacia izquierda y derecha giraremos en torno al punto al que estamos mirando, modificando el parámetro Este (E en la figura). Para tenerlo más claro, siempre estaremos situados en una esfera cuyo radio, que es la distancia al punto al que estamos mirando, es constante. Mientras nos vayamos moviendo observaremos que los ejes del triedro universal se van orientando de forma que nos indican la dirección de los ejes principales con respecto a nuestra posición.

Al pulsar el botón derecho del ratón pasaremos a mover el punto al cual estamos mirando a lo largo del plano horizontal, del plano XY de la referencia absoluta. El efecto en este caso es el de traslación, tanto nuestra como del objetivo por el plano horizontal. Mientras que si mantenemos pulsados ambos botones del ratón lo que haremos será modificar la altura del punto al cual miramos (a lo largo del eje Z de la referencia absoluta), lo que se traduce en términos coloquiales como mirar hacia arriba o hacia abajo, para entenderlo mejor.

Si por el contrario estamos en una vista ortogonal el único efecto que realizamos con el ratón es el de traslación de la cámara a través del plano de proyección.

Un último control que podemos realizar con la cámara, común a todas las representaciones es el de poder acercar o alejar la cámara mediante la rueda



central del ratón, lo que hace es disminuir el radio de la esfera y se conoce como Zoom. También se puede realizar este control mediante los iconos *acercar* y *alejarse* de la barra de tareas o mediante *Menú principal > Ver > Acercar* o *Menú principal > Ver > Alejar*

### **Modo de medida de distancias**

Se puede seleccionar este modo mediante *Menú principal > Acción > Medir distancias* o bien mediante el icono correspondiente de la barra de herramientas.

Siempre nos resultará útil conocer la distancia entre dos puntos, pero obviamente sólo se podrán realizar mediciones cuando estemos en los visores de proyección ortogonal, puesto que estaremos perpendiculares al plano en cual queramos realizar las mediciones.

Para proceder a la toma de medidas, tendremos que pulsar el botón izquierdo del ratón en el punto origen, mantener apretado el botón, momento en el cual veremos una recta que conecta el punto de origen con el puntero del ratón, y soltar en el punto que queramos hacer efectiva la medida. En ese momento se realizará la medición entre ambos puntos y se mostrará un mensaje con la longitud medida en mm.

## **3.4 CREACIÓN Y MODIFICACIÓN DE OBJETOS. TIPOS DE OBJETOS**

Como ya se ha explicado suficientemente, mediante este módulo construiremos un sólido a partir de formas geométricas predefinidas. Todos los objetos que componen el sólido tienen una serie de atributos comunes, como la posición, orientación, escala o material. Se puede acceder a estas propiedades a partir del panel lateral (Panel lateral > Panel Modificar > Propiedades Generales).

De cara a modificar la geometría del objeto no es necesario en ningún momento acceder a la estructura interna de vértices y caras de cada objeto puesto que todos los tipos de objetos que tenemos a nuestra disposición vienen convenientemente parametrizados así que, dependiendo del tipo de objeto que tengamos seleccionado tendremos acceso a diferentes propiedades específicas. Las propiedades específicas de cada objeto estarán accesibles desde el panel lateral.

### **La caja**

Podemos crear una esfera mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón caja) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Caja)

Es el objeto más sencillo y tiene tres características básicas: altura, anchura y profundidad, y podremos acceder a ellas desde el panel lateral (Panel lateral > Panel Modificar > Propiedades caja). Podemos apreciar el significado de dichas propiedades en la siguiente figura:

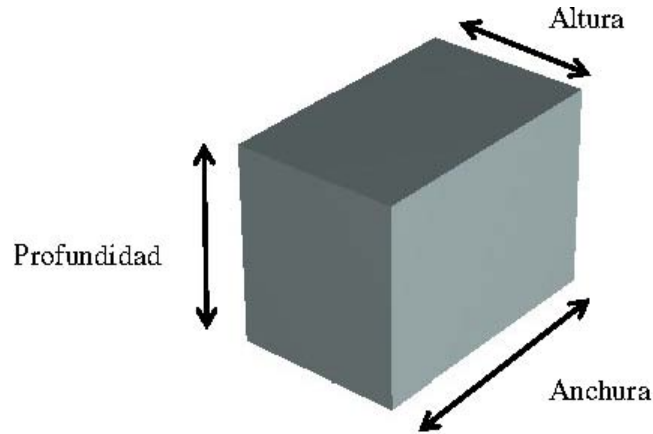


Figura 9. Objeto caja

### La esfera

Podemos crear una esfera mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón esfera) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Esfera)

Este objeto tiene tres propiedades específicas. La más importante es el radio, mientras que las otras dos hacen referencia al número de polígonos con los que se construye la esfera. Suponiendo que una esfera se construye mediante la revolución de una semicircunferencia a lo largo de un eje (el eje Y del objeto, en concreto), el término divisiones se refiere al número de segmentos de la semicircunferencia, mientras que el término rodajas se refiere al número de divisiones radiales a lo largo de la revolución. Se pueden observar estos parámetros en la siguiente imagen:

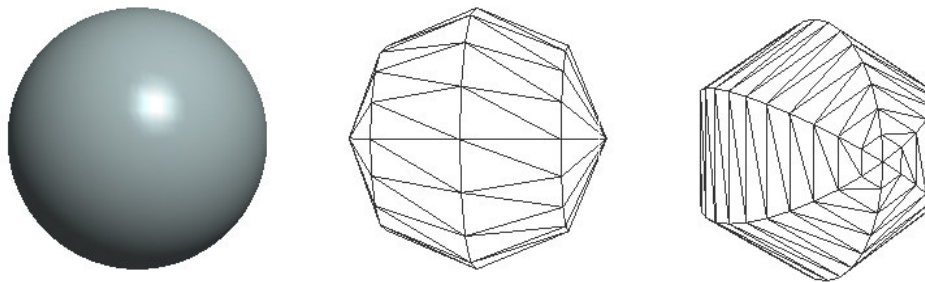


Figura 10. En la izquierda una esfera suavizada, en el centro una esfera con 4 divisiones y en la derecha una esfera con 6 rodajas.

### El cilindro

Podemos crear un cilindro mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón cilindro) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Cilindro).

Este objeto tiene las siguientes propiedades específicas: Altura, radio superior,



radio inferior y lados. La altura se mide a lo largo del eje Y, mientras que los radios superior e inferior pueden ser distintos, por lo que se pueden llegar a construir troncos de cono. Por otra parte, la propiedad lados se refiere al número de lados de la base circular. Podemos apreciar estos parámetros en la siguiente figura:

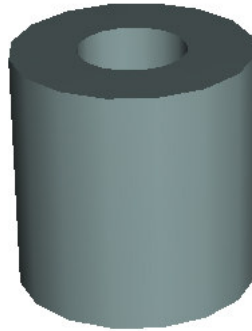


Figura 11. Objeto cilindro

### El cono

Podemos crear un cono mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón cono) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Cono).

Este objeto tiene las siguientes propiedades específicas: Radio, altura y lados. La altura mide la distancia desde el punto central de la base hasta el vértice superior, medida a lo largo del eje Y. El radio es el radio de la base mientras que la propiedad lados se refiere al número de lados de la base circular. Podemos apreciar estos parámetros en la siguiente figura:

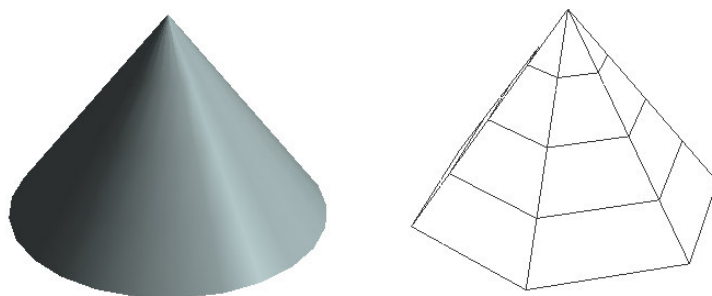


Figura 12. Objeto cono

### El toroide.

Podemos crear un toroide mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón Toroide) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Toroide).

Este objeto tiene las siguientes propiedades específicas: Radio interno, radio



externo y lados y segmentos. Si suponemos que un toroide ha sido creado mediante la revolución de una circunferencia a través de un eje (del eje Z del objeto en nuestro caso), el radio externo se refiere a la distancia desde el centro de la circunferencia que genera el volumen hasta el eje de giro, medida perpendicularmente. El radio interno se refiere al radio de la circunferencia que genera el volumen. El parámetro segmentos establece el número de divisiones radiales del toroide mientras que el parámetro segmentos establece el número de lados de cada sección del toroide. Podemos apreciar estos parámetros en la siguiente figura:

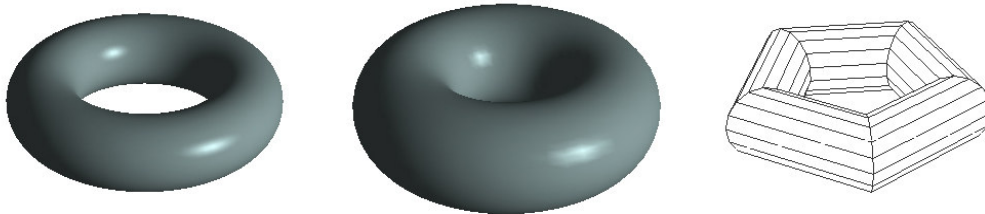


Figura 13. Distintos tipos de toroide

En los dos toroides de la izquierda de la figura se pueden observar los cambios que se producen al variar los parámetros de radio interno y radio externo, mientras que en el de la derecha, en modo alambres, el toroide tiene 5 segmentos y 15 lados.

### El anillo

Podemos crear un anillo mediante el panel lateral (Panel lateral > Panel Crear > Geometría básica > Botón anillo) o bien mediante el menú principal (Menú principal > Objeto > Nuevo > Geometría básica > Anillo).

Este objeto es similar al cilindro pero con agujero a través de su eje principal y tiene las siguientes propiedades específicas: Radio interno, radio externo y altura. Los parámetros radio externo y altura tienen exactamente el mismo significado que en el caso del cilindro, mientras que el parámetro del radio interno se refiere al radio del agujero que lo atraviesa. Podemos apreciar estos parámetros en la siguiente figura:

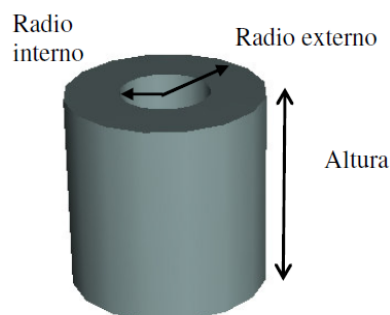


Figura 14. Parámetros de un anillo

### Tronco de pirámide



Podemos crear un tronco de pirámide mediante el panel lateral (Panel lateral> Panel Crear> Geometría básica> Botón tronco) o bien mediante el menú principal (Menú principal > Objeto > Nuevo> Geometría básica > Tronco de pirámide).

El tronco de pirámide consiste en una pirámide de base rectangular truncada a una determinada altura medida a lo largo del eje Y del objeto. Este objeto posee las siguientes propiedades específicas: Profundidad base, anchura base, altura y punto de fuga. La profundidad y la anchura de la base se refieren, respectivamente a las medidas de la base, a lo largo de los ejes Z y X (la altura se mide a lo largo del eje Y del objeto). El parámetro punto de fuga se refiere a la altura del vértice superior de la pirámide sin trincar, medido a lo largo del eje Y, y el parámetro altura se refiere a la distancia medida desde el centro de la base hasta el plano de truncamiento. Podemos apreciar estos parámetros en la siguiente figura:

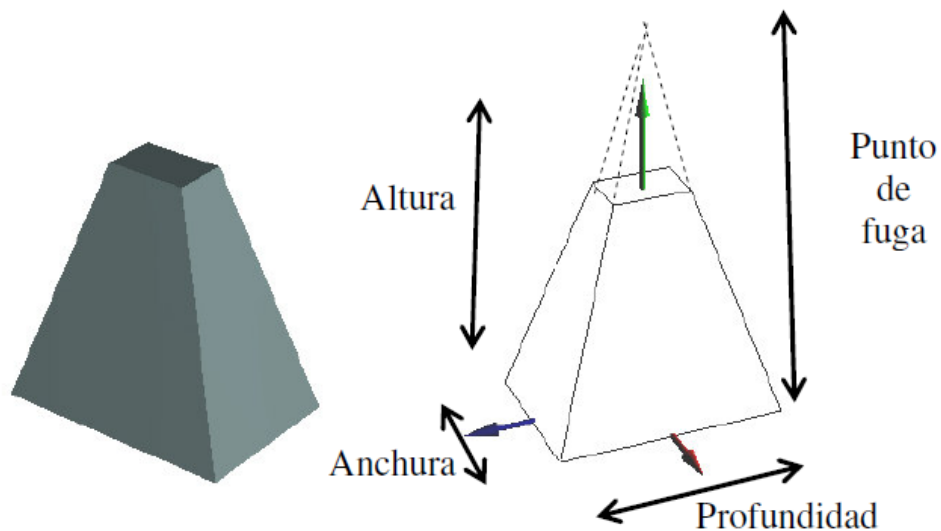


Figura 15. Parámetros del tronco de pirámide

## Plano

Podemos crear un plano mediante el panel lateral (Panel lateral> Panel Crear> Geometría básica> Botón plano) o bien mediante el menú principal (Menú principal > Objeto > Nuevo> Geometría básica > Plano).

El plano será creado en el plano XY y tiene las siguientes propiedades específicas: Anchura, profundidad, repeticiones en X y repeticiones en Y. La anchura y la profundidad se refieren al las dimensiones del plano medidas en los ejes X e Y respectivamente, mientras que los parámetros de las repeticiones serán válidos cuando tengamos un material con una textura asignado a nuestro plano, y representan el número de veces que se repetirá la textura a lo largo de cada eje. De todas formas, ya se tratará más adelante el tema de los materiales y del mapeado de las texturas.



Figura 16. Planos con texturas y repeticiones

### 3.5 FORMAS MÁS COMPLEJAS: EXTRUSIÓN Y REVOLUCIÓN

Aparte de las formas básicas explicadas, para la construcción de objetos de formas más complejas se ha habilitado la creación de volúmenes mediante 'desplazamiento' de superficies. De esta manera se deberá definir una superficie plana, bien mediante segmentos o mediante curvas 'spline', y una vez definida se construirá un volumen mediante el movimiento de esa superficie. Tenemos tres tipos de objetos que se construyen con este método: Objetos por extrusión, por revolución o el objeto que hemos denominado tubería, y los podemos ver en la siguiente imagen:

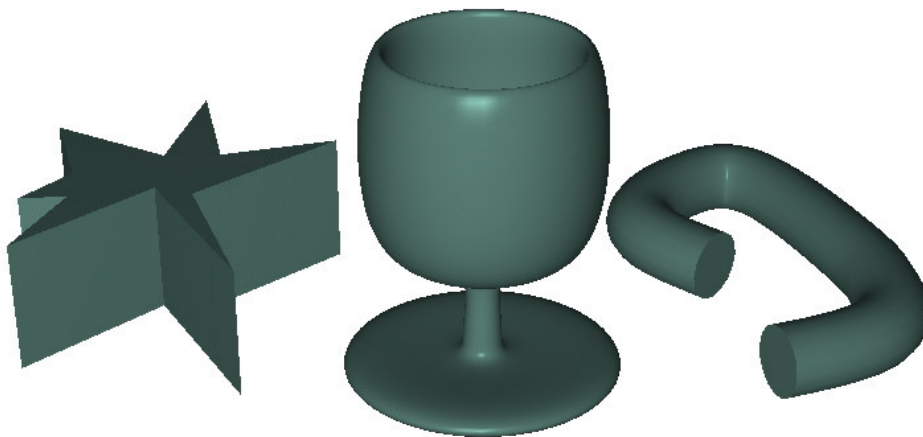


Figura 17. De izquierda a derecha extrusión, revolución y tubería.

Para la creación de volúmenes mediante esta técnica se ha habilitado una ventana especial, a la que podremos acceder mediante el panel lateral (Panel lateral> Panel Crear> Geometría avanzada> Botón construir extrusión) o bien mediante el menú principal (Menú principal > Objeto > Nuevo> Geometría avanzada > Mediante extrusión).

Para la construcción de dichos sólidos se presentan interfaces similares, y podemos ver un ejemplo en la siguiente imagen.

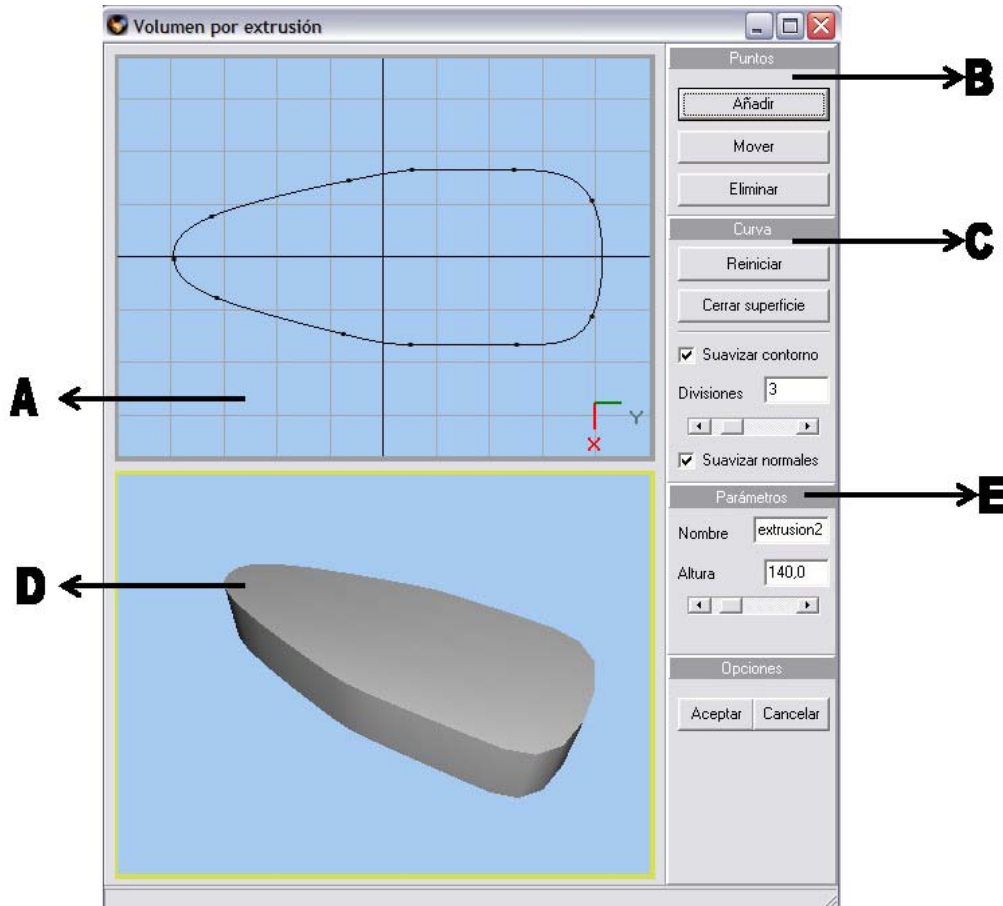


Figura 18. Interfaz para la creación de sólidos por extrusión.

Podemos distinguir las siguientes zonas en la interfaz:

**A) Zona de dibujo.** Este visor se utiliza para la definición de la curva que generará el volumen por extrusión. En este visor deberemos dibujar la curva con el ratón dependiendo del modo de dibujo adoptado mediante los controles laterales, podremos añadir puntos, mover puntos ya dibujados o eliminar puntos sobrantes.

**B) Zona de selección de modo de trabajo:** A través de los controles habilitados en esta zona podremos establecer el modo de dibujo. Tenemos tres modos, accesibles desde los botones Añadir, mover y eliminar respectivamente. Estos modos de dibujo nos definen cómo será la interacción del ratón con la zona de dibujo. En el primer caso nos encontraremos en el modo de añadir puntos y en este modo cada vez que pulsemos el botón izquierdo del ratón añadiremos un punto nuevo a nuestra curva en el punto en el que tengamos el puntero del ratón, para terminar de dibujar podremos hacerlo bien pulsando el botón derecho del ratón bien cerrando la superficie, esto es, colocar un punto donde hubiera ya otro punto anteriormente. En ese momento se nos preguntará si queremos cerrar la superficie. Si respondemos afirmativamente la superficie se cerrará y dejaremos de dibujar mientras que si la respuesta es negativa seguiremos dibujando. En el modo de mover puntos deberemos seleccionar con el ratón el punto del cual queremos cambiar su posición, poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón. Una vez seleccionado el punto, sin soltar el botón izquierdo del ratón arrastraremos el punto por la zona de dibujo hasta la posición deseada. Para





dejar de arrastrar sólo hay que soltar el botón. En el tercer modo procederemos de manera similar a como lo hemos hecho en el modo anterior, con la diferencia de que el punto que seleccionemos con el ratón será eliminado.

**C) Zona de control de la curva.** Mediante los controles habilitados en esta zona controlaremos todos los parámetros de la curva. Mediante los dos primeros controles (botón Cerrar superficie y botón Reiniciar) podremos cerrar la superficie o reiniciar el dibujo. Al pulsar el botón de cerrar superficie lo que se hace es juntar el primer punto dibujado con el último, con lo que pasamos a tener una superficie cerrada, mientras que si pulsamos el botón de reiniciar, como su propio nombre indica reiniciaremos la curva, se perderán todos los datos y procederemos a dibujar una curva totalmente nueva. Tenemos también en esta zona la posibilidad de suavizar el contorno. Si activamos esta posibilidad en lugar de ser segmentos rectos los que unen los puntos serán curvas polinómicas de tercer grado, más conocidas como splines cúbicos, así, la curva resultante pasará por cada uno de los puntos pero seguirá una trayectoria suave, sin cambios bruscos, esto es, sin aristas, tal y como se puede ver en la siguiente imagen:

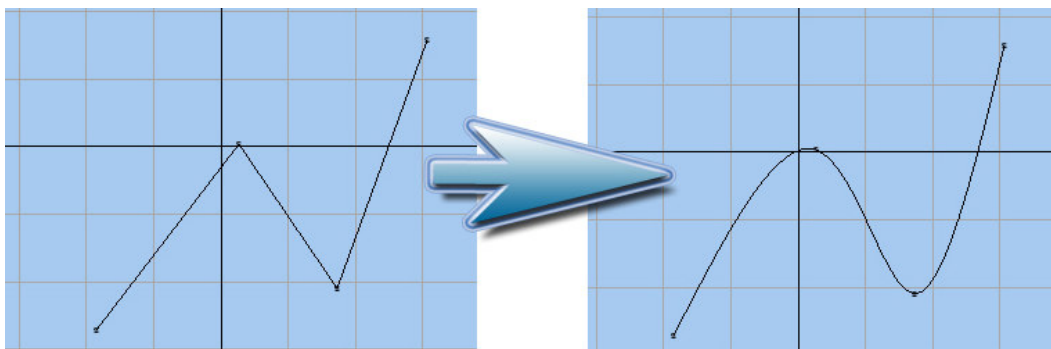


Figura 19. Efecto de la aplicación de splines cúbicos

El siguiente control, denominado divisiones, se refiere al número de veces que son muestreadas las curvas polinómicas, por lo que este parámetro sólo tiene sentido cuando aplicamos el suavizado de contorno. Aunque el contorno nos venga definido mediante polinomios, dicha curva será muestreada y convertida a pequeños segmentos, a cuyo número hacemos referencia mediante el parámetro divisiones. Obviamente, a mayor número de divisiones mayor fidelidad a la curva pretendida, pero también incurriremos en una mayor carga poligonal, lo cual no podría llegar a perjudicar en la velocidad de nuestra aplicación. Íntimamente ligado al control de divisiones tenemos la opción de suavizado de normales, este control afectará al sólido generado. Si este control está desactivado se considerará cada segmento como una cara independiente en cuanto a su representación gráfica se refiere, mientras que si suavizamos las normales se suavizará la representación de la superficie, tal y como se puede ver en la siguiente imagen:

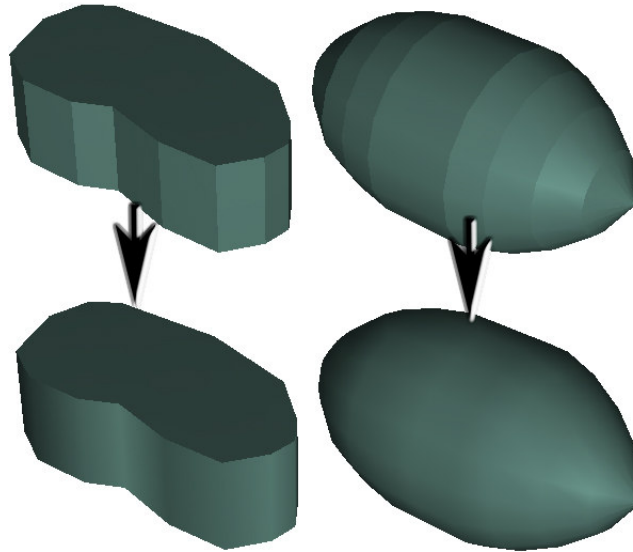


Figura 20. Ejemplos de suavizado en una extrusión (izquierda) y una revolución (derecha)

Generalmente, cuando se proceda a suavizar el contorno se deberían suavizar también las normales, y cuando no suavicemos el contorno no deberíamos suavizar tampoco las normales para evitar posibles efectos no deseados en el sólido.

**D) Visor tridimensional.** En este visor podemos observar como está quedando nuestro volumen en todo momento. Podemos girar la cámara con el ratón para observarlo desde cualquier ángulo, así como acercarlo o alejarlo mediante la rueda del ratón, de forma análoga a como lo hacemos en el visor principal de la aplicación.

**E) Zona de definición del volumen:** Una vez creada la superficie, el volumen generado queda definido mediante una serie de parámetros accesibles desde esta zona. Todas las partes de la interfaz comentadas hasta ahora son comunes para todo tipo de sólidos mientras que esta zona será específica del tipo de sólido generado y las opciones accesibles desde este panel describen cómo se crea el sólido a partir de la curva que hayamos definido anteriormente.

### Procedimiento para dibujar polilíneas

Este proceso es el mismo para los tres tipos de volúmenes, y aquí explicaremos su funcionamiento. Como ya hemos explicado tenemos tres modos de trabajo, modo de añadir puntos (Panel lateral> botón añadir), modo modificar la posición de los puntos (Panel lateral> botón mover) y modo de eliminar puntos (Panel lateral> botón eliminar).

Para empezar a dibujar deberemos activar el modo de añadir puntos, y en el visor bidimensional ir pulsando con el botón izquierdo del ratón donde queramos un nuevo punto. Cuando no queramos añadir más puntos dejaremos de dibujar tal y como hemos explicado antes (botón derecho del ratón o creando un punto donde ya había otro). Este procedimiento queda ilustrado en la siguiente figura.

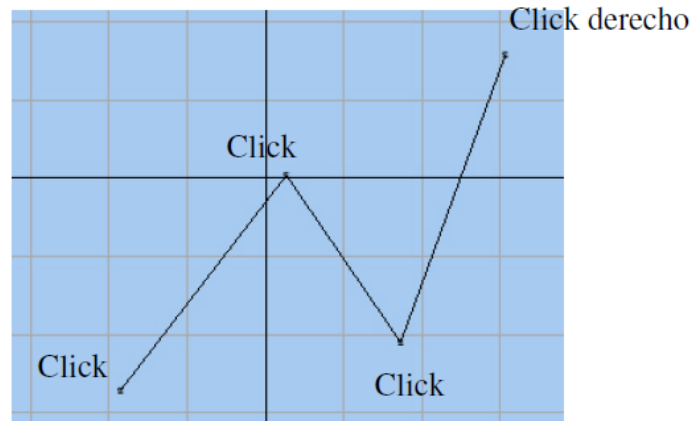


Figura 21. Procedimiento de dibujo de polilíneas

Siempre que queramos podemos activar el modo de modificar la posición de los puntos hasta que tengamos la superficie como deseemos. Si queremos suavizar el contorno activaremos la casilla correspondiente (Panel lateral >casilla suavizar contorno)

### **Sólido por extrusión.**

La construcción de este tipo de sólido se produce mediante la traslación a lo largo del eje Z de una superficie cerrada definida en el plano XY. Para este tipo de sólidos la interfaz tomará el aspecto de la figura XX (la de muestra).

La generación de este tipo de sólidos viene definida por una serie de parámetros específicos situados en el panel lateral. Estamos ante el caso más simple, ya que sólo contamos con dos parámetros, el primero es el nombre, que no tiene ninguna repercusión en la geometría, y el segundo es la altura, que es la distancia en milímetros que se ha trasladado la superficie a lo largo del eje Z.

Por regla general, no suele hacer falta subir a más de 4 el número de segmentos si se suavizan las normales, la única consecuencia de reducir el número de segmentos podría darse si el material tuviera brillo, que podría quedar ligeramente deformado en el caso de reducir excesivamente el número de segmentos suavizando las normales, aunque esto es extrapolable al resto de elementos que posean suavizado de normales, desde la esfera hasta la revolución.

### **Sólido por revolución**

La generación de este sólido se produce mediante la revolución en torno al eje Y de una curva definida en el plano XY. Podemos conseguir mediante este método de generación un gran abanico de sólidos. Podemos ver en la siguiente figura un ejemplo de los sólidos que se pueden crear mediante esta técnica.

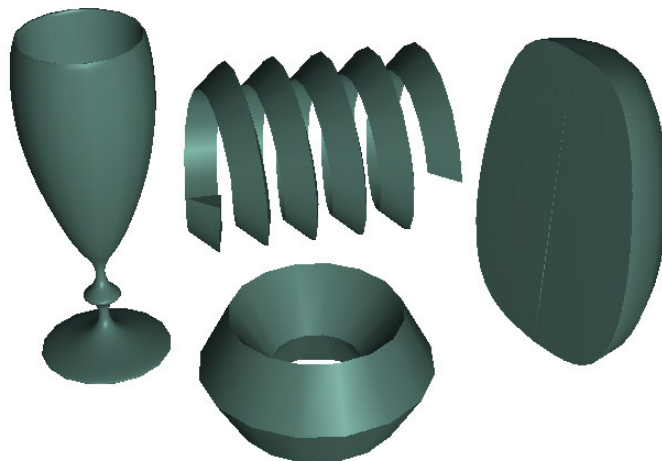


Figura 22. Sólidos creados por revolución.

Para este tipo de sólidos la interfaz toma la siguiente forma:

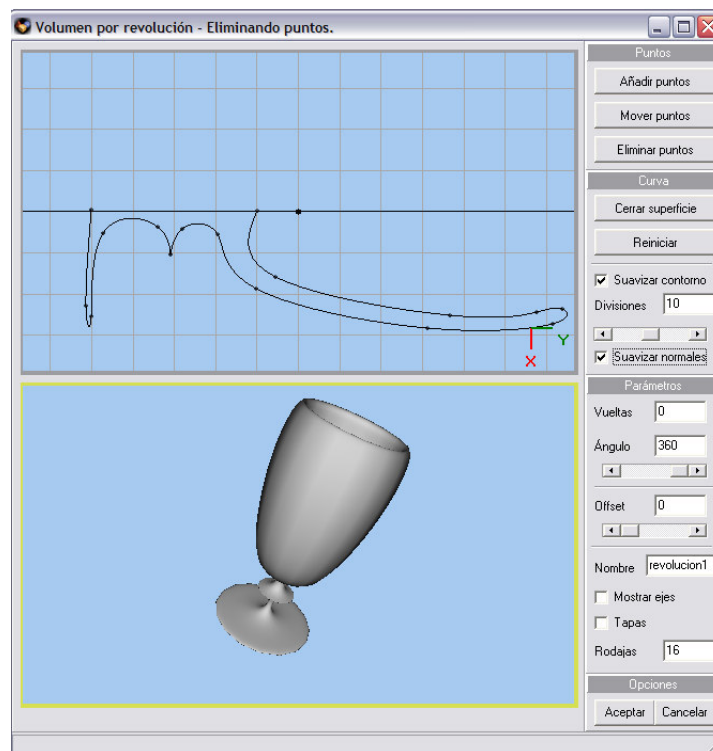


Figura 23. Interfaz de creación de sólidos por revolución

La generación de este tipo de sólidos viene definida por una serie de parámetros específicos situados en el panel lateral, en la parte *parámetros*. Estamos ante el caso más complejo y más versátil, un gran tipo de sólidos pueden describirse de este modo. El parámetro vueltas se refiere al número de veces que la superficie gira una vuelta entera en torno al eje Y. A menos que no cambiemos el offset este parámetro debería ser 0. El parámetro ángulo se refiere al ángulo girado, si es menor de 360 el giro no estará completo y se producirá un efecto de segmentación.



En este caso será útil la opción de ponerle tapas a nuestro sólido (Panel lateral > casilla Tapas), aunque sólo tendrá sentido si la superficie que genera el volumen es una superficie cerrada.

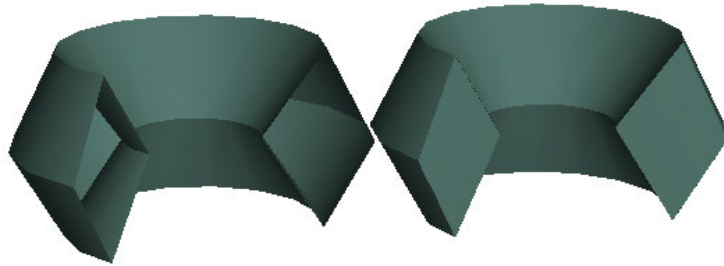


Figura 24. Efecto de las tapas para un ángulo menor de  $360^\circ$

El parámetro offset permite que en cada revolución la superficie avance una determinada longitud a lo largo del eje Y. Si se combina esta propiedad con un número de vueltas no nulo tendremos objetos de forma helicoidal:

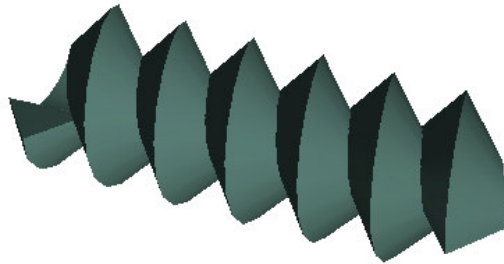


Figura 25. Sólido helicoidal con offset no nulo

Por último el parámetro rodajas es análogo al utilizado en objetos como el cilindro o la esfera y se refiere al número de divisiones radiales de la revolución.

### Tubería

La generación de éste tipo de sólido se produce mediante el desplazamiento a lo largo de una curva contenida en el plano XY de una sección circular. En este caso, y a diferencia de los dos tipos de sólidos anteriores, la superficie que genera el volumen es una superficie circular obligatoriamente mientras que la curva que definamos representará el recorrido de dicha superficie circular. Para este tipo de sólidos la interfaz tomará el siguiente aspecto.

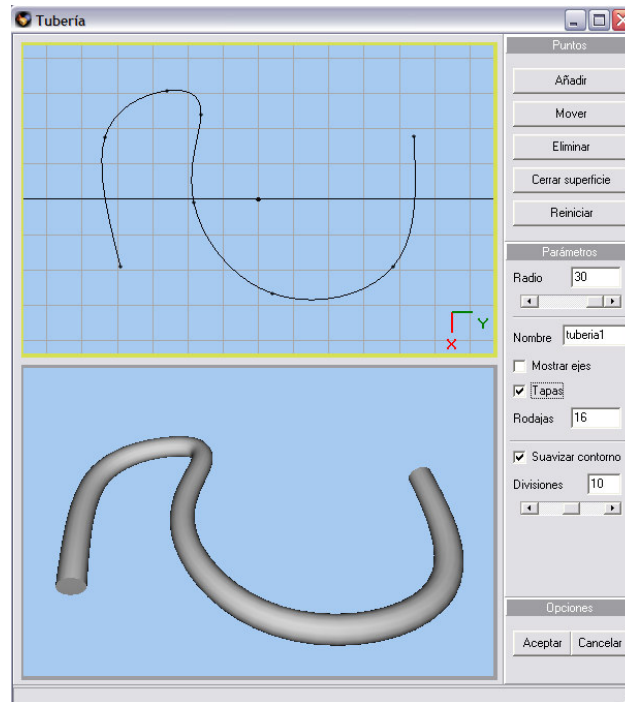


Figura 26. Interfaz para la creación de sólidos de tubería

Controlaremos la generación de este volumen mediante una serie de parámetros específicos situados en el panel lateral *parámetros*, a saber: El parámetro radio se refiere al radio en milímetros de la sección circular que generará el volumen mientras que el parámetro rodajas se refiere al número de segmentos de dicha superficie circular. También tendremos la opción de ponerle tapas de inicio y fin al volumen (*Panel lateral > casilla Tapas*).

### 3.6 MATERIALES

Una vez creada la geometría con la que representaremos a nuestro objeto en posteriores módulos llega el momento de 'vestir' los objetos con las superficies adecuadas. Un material describe el comportamiento de la superficie de un objeto frente a la luz, como refleja o como transmite luz. Se ofrece también la posibilidad de aplicar texturas que se adaptarán a la superficie de un objeto. Al aplicar materiales a un objeto se consigue que presente un aspecto realista, podemos tener una superficie lisa, rugosa, brillante o que presente diversos colores o diseños, en definitiva todo aquello que permite recrear los materiales reales.

#### Librería de materiales

Para permitir al usuario una reutilización de materiales se ha creado el concepto de librería de materiales. De esta forma, cada uno de los objetos de nuestra escena tiene asociado un material, pero dicho material se almacena en la librería de materiales, de manera que un número ilimitado de objetos podrán usar dicho material, evitándonos así tener que definirlo para cada objeto que necesite usar dicho material. Una librería de materiales no es más que un conjunto de materiales, cada uno con sus características específicas. Cada escena creada con el constructor de objetos (lo que será un sólido en los posteriores módulos) tiene asociada una librería de materiales, y cada uno de los objetos de dicha escena tiene



asociado un material de dicha librería de materiales, puede haber materiales que no usa ningún objeto o por el contrario podemos encontrarnos con materiales que usan uno o varios objetos de la escena. Para una mayor reutilización de la información cada librería de materiales se guarda en un fichero específico, de manera que una librería puede ser utilizada las veces que queramos, algo que será de suma utilidad, por ejemplo a la hora de definir cada uno de los sólidos de un robot, sólo tendremos que definir los materiales utilizados una vez.

### Acceso a la librería de materiales: El editor de materiales.

Se puede acceder al editor de materiales de tres maneras distintas: mediante el menú principal (Menú principal > Acción > Activar el editor de materiales), mediante el icono correspondiente de la barra de herramientas o mediante la propiedad material del objeto (Panel lateral > Panel Modificar > Propiedades Generales > Botón Material).

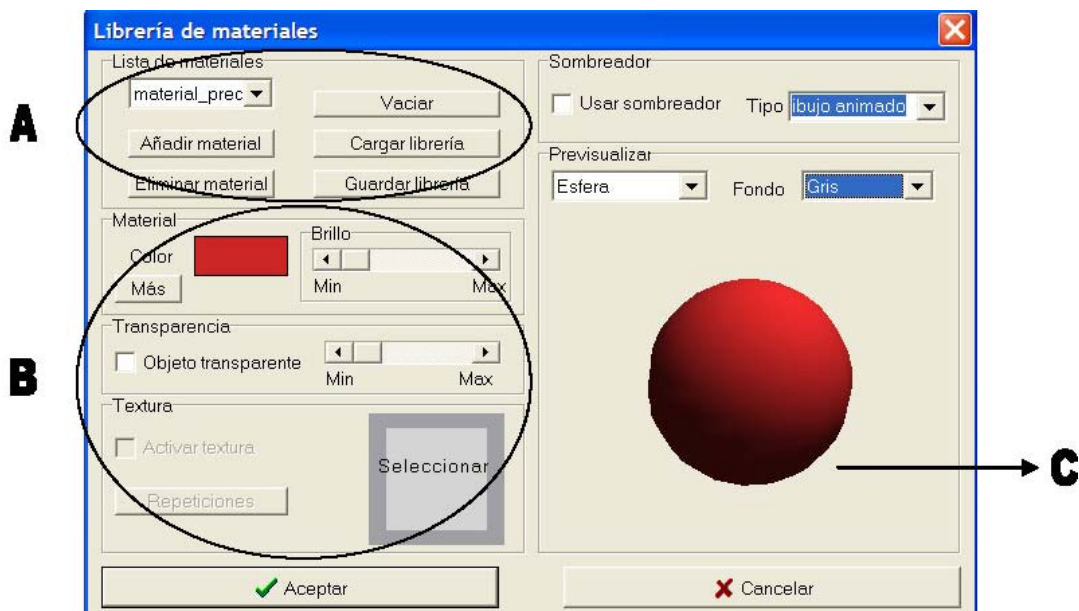


Figura 27. Interfaz del editor de materiales

Se pueden apreciar las siguientes zonas claramente diferenciadas en la interfaz:

- A. Zona de navegación por la librería. En esta zona podremos navegar por los diferentes materiales que componen nuestra librería, añadir nuevos materiales, eliminar materiales o bien guardar en disco la librería o abrir una ya existente. Las librerías se guardan en formato \*.lib, un formato binario que incluirá las texturas en caso de que las hubiera, lo cual es una ventaja a la hora de transportar nuestros archivos, puesto que no tendremos que transportar los archivos de imagen.
- B. Zona de edición del material. Una vez seleccionado un material de todos los de la librería en esta zona se procederá a la definición propiamente dicha del material. Tenemos dos partes la parte superior, en la que definiremos los parámetros básicos del material y la parte inferior, en la cual procederemos a la aplicación de mapas de textura al objeto.



- C. Zona de previsualización. En esta zona podemos ver como está quedando nuestro material. Disponemos de tres tipos de objetos en los que comprobar nuestro material, el cubo, la esfera o la taza de té (objeto ciertamente habitual en este tipo de programas). Asimismo también podremos escoger el tipo de fondo que queramos para la previsualización de nuestro material, pudiendo elegir incluso fondos multicolor, que resultarán de máxima utilidad cuando definamos objetos con transparencias.

### **Propiedades básicas de los materiales.**

Como acabamos de comentar accederemos a las propiedades del material mediante la zona de edición del material de la ventana de la librería de materiales.

Los **materiales básicos** son aquellos que presentan un color uniforme, y se distinguen de los materiales con mapa que tienen superficies con diseño. A la hora de definir materiales básicos podemos distinguir cuatro componentes de color:

Color ambiental: es el color de la sombra del objeto, es la parte iluminada por la luz que rebota en el entorno, por la luz ambiente, a diferencia de luz que incide directamente en el objeto.

Color difuso: es el color 'real' del objeto, lo que nosotros encendemos coloquialmente como color de un objeto y representa el color en las zonas en las que incide directamente la luz emitida por una fuente de luz.

Color especular: es el color que presentas las partes más brillantes del objeto, es el color del brillo del objeto, por lo que va íntimamente ligado con la propiedad brillo del objeto. Así, un objeto que tenga el color especular nulo (esto es, negro) presentará un aspecto totalmente mate, aunque aumentemos la propiedad del brillo. Mientras que un color distinto del negro, en conjunción con el uso del brillo nos permitirá definir el tipo de reflejo que queremos. Es altamente recomendable que el color especular del objeto pertenezca a la misma gama de colores que el difuso y el ambiente, en caso contrario se puede llegar a tener materiales irreales, o cuanto menos, psicodélicos.

Color emitido: es el color que emite el objeto, de forma que parece que está auto iluminado. Este color es nulo por defecto (negro), con lo que no obtendremos ningún efecto de auto iluminación a menos que no cambiemos el color emitido a uno no nulo.

Se pueden apreciar distintas combinaciones de colores en la siguiente imagen:



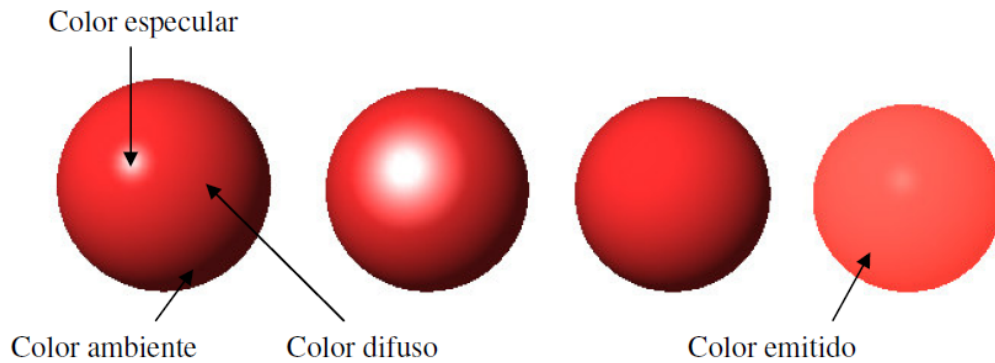


Figura 28. Diferentes combinaciones de colores para un mismo objeto

En la figura mostrada el color difuso es el mismo para todas las esferas. Las dos esferas de la izquierda tienen un color especular blanco, la diferencia entre ellas está en la propiedad brillo. En la primera está casi al mínimo mientras que en la segunda es bastante elevado. En la tercera esfera el color especular es nulo (negro), así que aquí la propiedad brillo resulta irrelevante, dando un acabado mate al objeto. Por último, la esfera de la derecha posee un color emitido no nulo, por lo que el efecto conseguido es el de que el objeto brille con luz propia (de hecho, en algunos programas de diseño gráfico, como 3DStudio, esta propiedad se define aparte y se llama autoiluminación).

En principio en nuestra interfaz sólo tenemos acceso a un color, el color difuso, y el resto de colores se calcularán en consecuencia. Si queremos personalizar más nuestro material tenemos acceso a todas las componentes del color mediante el botón 'más'. Para cambiar un color sólo tenemos que pulsar con el ratón en el recuadro coloreado y se nos abrirá el diálogo típico de Windows de selección de color, pudiendo elegir entre colores básicos, colores personalizados o pudiendo construir nosotros el color en todo el espectro que se nos ofrece.

Podemos aplicar transparencias a un objeto (*Panel transparencia > casilla Objeto transparente*), una vez que convertimos nuestro objeto en transparente tendremos que indicar el grado de transparencia del objeto. Para este fin resulta muy útil cambiar el fondo de la muestra a un fondo multicolor.

Los **materiales con mapas** o texturas permiten que los materiales no sólo presenten un color sólido sino imágenes que simulen materiales reales, dando aspecto de materiales rugosos, metálicos, etc. Si los materiales básicos equivalen a pintar un objeto, con las texturas el objeto se 'empapela' con una imagen, como se muestra en la siguiente imagen:



Figura 29. Efecto de aplicar una textura a un objeto

El uso de esta propiedad de los materiales se realizará a través del panel textura, situado abajo a la izquierda de la ventana. La textura actual del material se mostrará en vista en miniatura en el recuadro de la derecha del panel. Si el material seleccionado no tuviera textura en el recuadro aparecerá la palabra seleccionar. Si pulsamos este recuadro con el ratón tendremos acceso al correspondiente diálogo de Windows de selección de imagen, en el cual se han habilitado las imágenes en formato \*.bmp y en formato \*.jpg, y podremos escoger cualquier imagen de cualquiera de ambos formatos. Dentro del directorio predeterminado *Texturas* se incluyen una gran variedad de imágenes que podremos utilizar para dotar de apariencia más realista a nuestros materiales, si bien podremos abrir cualquier imagen que tengamos en el ordenador siempre que sea jpg o bmp. Es recomendable, aunque no imprescindible que la textura que asociemos, sus dimensiones sean potencia de 2, porque internamente así lo necesita OpenGL. Si no lo hiciéramos se convertirá internamente, y puede que el rendimiento sea peor. Normalmente se utilizan texturas cuadradas de tamaños estándar, como 128\*128, 256\*256 o 512\*512.

Una vez seleccionada una imagen tendremos que proceder a la activación de la textura (casilla Activar textura) para que el mapeado de la textura resulte efectivo, y veremos el resultado en la muestra de la previsualización.

Tenemos acceso, mediante el botón repeticiones a las repeticiones del mapa en la superficie. Si las repeticiones, tanto en el eje X como en el Y son la unidad una sola imagen 'empapelará' a todo el objeto. Si el número es menor que la unidad la textura será ampliada, mientras que si el número de repeticiones mayor que la unidad la textura se repetirá varias veces. Podemos ver un ejemplo en la siguiente imagen:



Figura 30. En el objeto de la derecha hay dos repeticiones en x



Recordemos que esta propiedad se encontraba ya disponible en los objetos de tipo plano, pero ahora ya se puede utilizar con cualquier tipo de objeto.

Si ya sabemos editar materiales ahora el único paso que nos queda es el asociarlos a los objetos que componen nuestra escena. Como ya hemos comentado existen tres formas de acceder al editor de materiales. Si accedemos de las dos primeras formas comentadas (menú principal y barra de tareas) lo que hacemos es simplemente acceder a la librería, con lo que podremos acceder a cualquier material y cambiar sus propiedades, en definitiva, todo lo comentado hasta ahora, mientras que si accedemos al material mediante el panel lateral accederemos a la misma interfaz, pudiendo realizar las mismas acciones, pero en el momento en el que le demos a aceptar en objeto seleccionado, el cual queremos cambiar su material, adoptará el material que hubiera activo en el editor de materiales.



## 4 EL CONSTRUCTOR DE ROBOTS

Tal y como se ha comentado en el apartado de Generalidades, la finalidad de este módulo es la construcción de un robot industrial a partir de los sólidos construidos con en módulo constructor de objetos, para posteriormente insertarlo en un entorno virtual y proceder a su explotación mediante el módulo constructor-explotador de escenarios.

En este módulo se procederá de la siguiente manera: Se construirá el robot añadiendo el número de sólidos necesarios. Se definirá cada articulación mediante una serie de parámetros como podrán ser el tipo, el rango o los parámetros de Denavit y Hartenberg, una vez construido correctamente el robot podremos guiarlo, así comprobaremos en todo momento el resultado. Tendremos también la opción más que recomendada de implementar el modelo inverso para nuestro robot, el cual programaremos mediante un script. Resumiendo, deberemos tener el robot resuelto en el papel para proceder con este módulo, con la asignación de referencias y la tabla de parámetros DH ya resueltas, así como las ecuaciones del modelo inverso, con lo que la implementación en el simulador será sencilla y rápida.

### 4.1 DESCRIPCIÓN DE LA INTERFAZ DE USUARIO

Si bien ya ha sido descrita antes de forma genérica, veamos hagamos ahora una descripción exhaustiva de la interfaz que muestra este módulo y sus características específicas con respecto a los siguientes módulos.

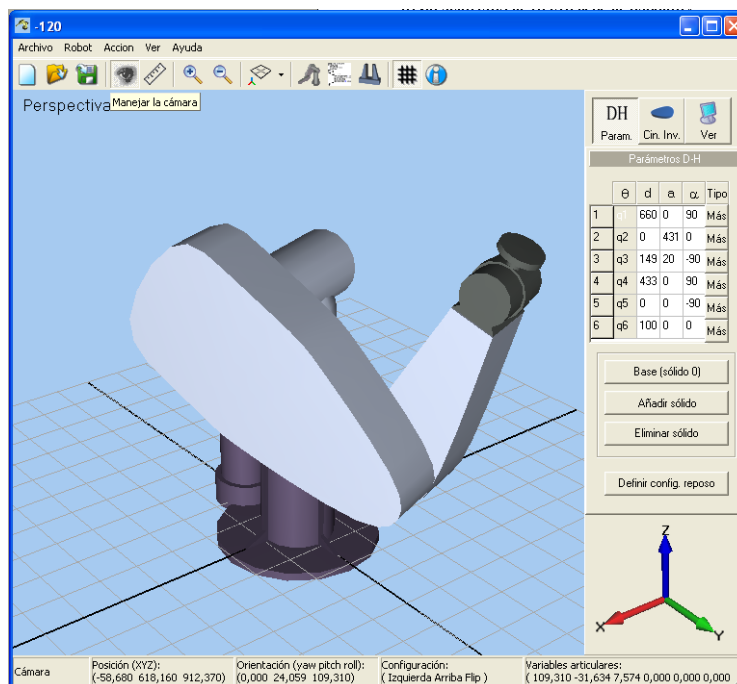


Figura 31. Interfaz de usuario del constructor de robots



## A. Menú principal

Desde este menú se puede acceder a casi todas las funcionalidades del programa, contiene los siguientes elementos:

Archivo: Mediante este menú se permite guardar el robot, abrir uno existente, empezar uno nuevo o bien abandonar la aplicación (estos conceptos serán explicados con más profundidad posteriormente)

Robot: Este menú es el más importante de este módulo puesto que mediante este menú se tiene el control del robot que está siendo construido, se pueden añadir o eliminar sólidos, editar la configuración, modificar la posición de reposo, llamar al editor de cinemática inversa, crear una nueva herramienta o proceder al guiado del robot.

Acción: Desde este menú se pueden seleccionar todos los modos de operación posibles, así como activar el editor de materiales.

Ver: Desde este menú se tiene acceso a todas las opciones referentes a la visualización de la escena en tres dimensiones.

## B. Barra de herramientas

Se encuentra justo debajo del menú principal y consiste en una serie de botones gráficos con las opciones más habituales del programa agrupadas por tipos:



Los tres de la izquierda permiten crear un nuevo sólido, abrir uno existente o guardar el actual, respectivamente.

Los tres siguientes dan acceso a los diferentes modos de operación.

Los dos siguientes permiten alejar o acercar la cámara.

El siguiente permite seleccionar el tipo de vista.

El siguiente permite activar la ventana de guiado del robot

El siguiente activa el editor de cinemática inversa

El siguiente activa el constructor de herramientas.

Los siguientes seleccionan el tipo de representación de los objetos.

## C. Panel de control lateral

De manera similar a como nos ocurría en el módulo anterior, este panel lateral se compone de tres subpaneles, que serán activados mediante los tres botones de control.

El primer panel, al cual accedemos mediante el botón *DH*, es el encargado de la cinemática directa de nuestro robot. A través de este panel podremos añadir o



eliminar sólidos, y tendremos acceso a la matriz de parámetros de DH, que podremos cambiar en todo momento y ver las consecuencias de dichos cambios. También tendremos acceso al resto de propiedades asociadas a cada articulación mediante una ventana específica.

En el segundo panel, accesible desde el botón *CI* tendremos acceso a todo lo relacionado con la cinemática inversa de nuestro robot, así como otras propiedades como la definición de la posición de reposo o la posibilidad de añadirle una herramienta.

#### **D. Barra de estado**

En esta parte de la interfaz tendremos acceso visual a todas las variables del robot, a saber: la posición del elemento terminal, la orientación a través de los ángulos de roll, pitch yaw, la configuración actual y el valor de las variables articulares del robot. Todas las medidas en milímetros/grados.

#### **E. Zona de visualización en tres dimensiones**

En esta parte de la pantalla se procede a la representación tridimensional del robot que está siendo creado. Aparte del robot podemos observar la cuadrícula o rejilla y los ejes principales, representados por dos líneas más gruesas que se cruzan en el origen, de manera idéntica a como se hace en el constructor de objetos.

## **4.2 METODOLOGIA DE CREACIÓN DE UN ROBOT**

El primer paso para la construcción de nuestro robot pasa por definir la cadena cinemática que representa nuestro robot, esto es, definir todos los parámetros necesarios de las articulaciones, como el tipo de articulación (rotacional o prismática) la geometría, el rango articular o la velocidad máxima. Posteriormente procederemos a la implementación del modelo inverso.

### **Geometría asociada a un sólido**

Cada componente de la cadena cinemática lleva asociada una determinada geometría, que habremos generado previamente mediante el constructor de objetos. Cada sólido del robot (y cada sólido de la escena, todo lo explicado ahora vale también para el siguiente módulo) tiene asociado un sistema de referencia. En nuestro caso, el sistema de referencia asociado a cada sólido se deberá asignar mediante el protocolo de asignación de Denavit Hartenberg, y esto lo deberemos de tener en cuenta cuando estemos construyendo el sólido. Tal y como se ha comentado antes, mientras construíamos un sólido en el constructor de objetos, lo hacíamos con respecto a un sistema de referencia al que denominábamos referencia absoluta. Ahora bien esa referencia absoluta ahora pasa a ser el sistema de referencia asociado al sólido, así que tendremos que haberlo tenido en cuenta mientras creábamos el sólido. Podemos ver una imagen de muestra aquí:

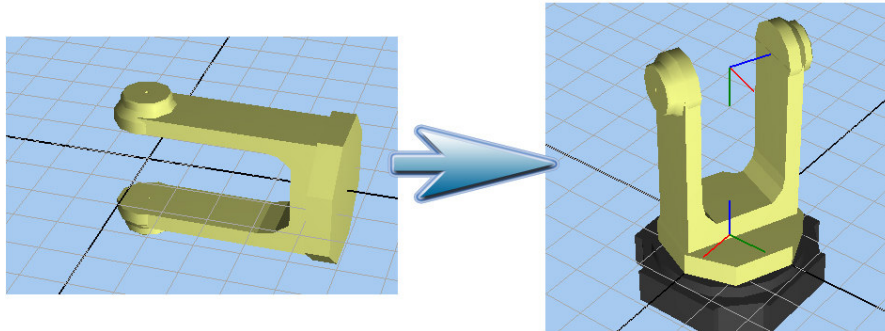


Figura 32. La referencia absoluta pasa a ser el sistema de referencia asociado al sólido

Lo mismo resultará aplicable en el siguiente módulo, en el cual cada objeto también tendrá un sistema de referencia asociado.

### Añadir o editar una articulación.

Independientemente de las articulaciones que tenga nuestro robot, podremos editar en todo momento la geometría asociada a la base fija del robot, que es el eslabón 0 o sólido 0 de la cadena cinemática mediante el panel lateral (Panel lateral > Panel DH > Botón Sólido 0) o mediante el menú principal (Menú principal > Robot > Base). Nos aparecerá un diálogo para escoger un archivo que previamente deberemos haber creado mediante el constructor de objetos.

Podremos añadir un sólido a nuestro robot bien mediante Panel lateral > Panel DH > Botón añadir sólido o bien mediante Menú principal > Robot > Añadir sólido. Aparecerá una ventana como la siguiente:



Figura 33. Ventana para añadir un sólido al robot

En Esta ventana podremos controlar los parámetros básicos de la articulación. Podremos elegir el tipo, Rotacional o prismática, el rango de la articulación (en grados si es rotacional y en milímetros si es prismática), la velocidad máxima (en grados/segundo si es rotacional y en milímetros/segundo si es prismática) y la geometría pulsando el botón Cargar Objeto. Al pulsar dicho botón abriremos un cuadro de diálogo en el que deberemos elegir un archivo que previamente deberemos haber creado mediante el constructor de objetos. A continuación se nos mostrará en la parte derecha una muestra del objeto. En caso de que optemos por



la opción de aceptar una nueva articulación se nos añadirá al robot, y la interfaz sufrirá varios cambios, se añadirá una fila a la matriz de parámetros de  $DH$  y se nos añadirá un control más a la botonera de guiado, elementos ambos que procederemos a explicar en los próximos epígrafes.

También se nos permitirá cambiar cualquiera de estos parámetros una vez creada la articulación, para ello tendremos bien que pulsar en el botón correspondiente a la articulación a la derecha de la matriz de parámetros  $DH$  (ver imagen) o bien haciendo doble clic con el ratón encima del sólido correspondiente en el visor tridimensional. Accederemos a la misma ventana, que nos mostrará los valores actuales y procederemos de igual manera a como habíamos hecho previamente.

### Parámetros de Denavit Hartenberg.

Una vez asignado un sistema de referencia a cada sólido de la cadena cinemática del robot según el protocolo de asignación de  $DH$ , la situación de un eslabón  $L_i$  respecto del anterior,  $L_{i-1}$  queda totalmente definida mediante los cuatro parámetros de Denavit Hartenberg,  $a_i$ ,  $d_i$ ,  $\alpha_i$  y  $\theta_i$ . Estos parámetros tienen el siguiente significado geométrico:

**$a_i$ :** Longitud del sólido  $L_i$ , medida como la perpendicular común entre los ejes de las articulaciones  $J_i$  y  $J_{i-1}$ .

**$d_i$ :** distancia entre los sólidos  $L_{i-1}$  y  $L_i$ . En caso de que la articulación sea prismática, mide su desplazamiento.

**$\alpha_i$ :** ángulo de torcimiento del eslabón  $L_i$ , medida como el ángulo entre los ejes de las articulaciones  $J_i$  y  $J_{i-1}$ .

**$\theta_i$ :** ángulo entre los sólidos  $L_{i-1}$  y  $L_i$ . En caso de que la articulación sea rotacional mide el ángulo de giro.

Estos parámetros serán accesibles en todo momento mediante la matriz situada en el panel lateral, subpanel  $DH$ , tal y como se puede ver en la siguiente imagen.

	$\theta$	$d$	$a$	$\alpha$	Tipo
1	q1	660	0	90	Más
2	q2	0	431	0	Más
3	q3	149	20	-90	Más
4	q4	433	0	90	Más

Figura 34. Matriz de parámetros  $DH$

Las casillas son accesibles en todo momento y el efecto del cambio será inmediato en el robot construido, pudiendo observar en todo momento el estado actual del robot en el visor tridimensional.





De los cuatro parámetros de Denavit Hartenberg uno de ellos se corresponde con el valor de la variable articular,  $\theta_i$  si la articulación es rotacional y  $d_i$  en el caso de que la articulación sea prismática, como bien acabamos de comentar. Estos parámetros no son accesibles desde la matriz como el resto, sus casillas pasan a tener un color gris y no son editables. Para poder modificar el valor de las coordenadas articulares se ha habilitado una ventana específica al más puro estilo de una botonera de guiado de un robot comercial, ventana que explicaremos más adelante con detalle.

A la hora de estar construyendo un robot puede sernos de gran utilidad el poder observar los sistemas de referencia de cada uno de los eslabones, así que tendremos la oportunidad de ver todos los sistemas de referencia, ver sólo el primero (la base, el sólido 0) y el último o bien no ver ninguno, opciones accesibles desde el menú principal (menú principal > ver > ejes del robot). El color de los ejes será el habitual en estos casos, rojo para el eje X, verde para el eje Y y azul para el eje Z. En caso de que tengamos algún eje visible, éste será visible en todo momento, aunque tenga algún objeto delante, para así tener una mejor imagen y lograr una mayor comprensión.

### Definición de la posición de reposo.

Podemos definirnos una posición de reposo, que será la configuración (conjunto de variables articulares) que adoptará el robot cuando lo abramos, o cuando lo incluyamos en un escenario virtual mediante el siguiente módulo. Siempre tendremos esta posición accesible, bien desde la ventana de guiado bien desde una aplicación robótica.

Podremos definir la posición de reposo mediante (Panel lateral > subpanel DH > Botón Definir config. reposo) o desde el menú principal (menú principal > Robot > Definir configuración reposo). Inmediatamente aparecerá la siguiente ventana:

Articulación	Valor
1ª	90,0
2ª	-90,0
3ª	180,0
4ª	0,0
5ª	0,0
6ª	0,0

Comprobar rangos

Aceptar Cancelar

Figura 35. Ventana para definir la posición de reposo

Deberemos introducir el valor de cada articulación en la posición de reposo. Inicialmente el botón de Aceptar está bloqueado. Para desbloquearlo deberemos pulsar el botón de comprobar Rangos. Si hay alguna articulación fuera de rango se nos avisará, y en caso de que estén todas entre sus correspondientes límites la posición de reposo será modificada.



### 4.3 IMPLEMENTACIÓN DE LA CINEMÁTICA INVERSA

Podemos entender el modelo geométrico inverso de un robot como la determinación del valor que debe tomar cada una de las articulaciones para situar el órgano terminal en una posición conocida.

#### Definición de la configuración

La solución al modelo inverso no es única, y según sea la morfología del robot se podrá llegar a la solución (en caso de que se pueda llegar) de una o varias maneras. A esto le llamaremos configuraciones, y podemos ver un ejemplo en la siguiente figura:

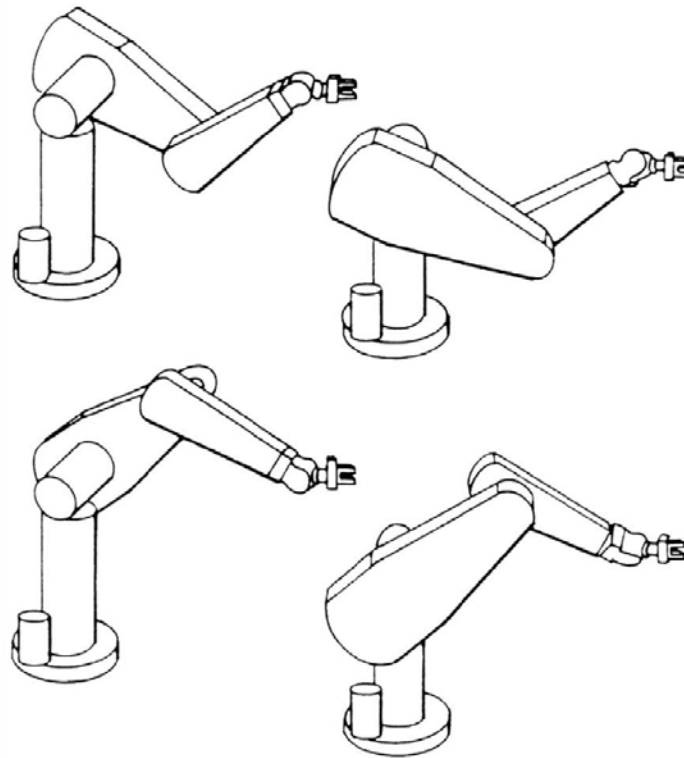


Figura 36. Diferentes configuraciones que puede adoptar el robot

Antes de proceder a la implementación programada del modelo inverso del robot deberemos definir cuales será las posibles configuraciones que podrá adoptar nuestro robot, y lo haremos mediante una ventana específica que será accesible desde el panel lateral (panel lateral > panel CI > Botón Definir configuración) o bien desde el menú principal (Menú principal > definir configuración) que tendrá la siguiente forma:

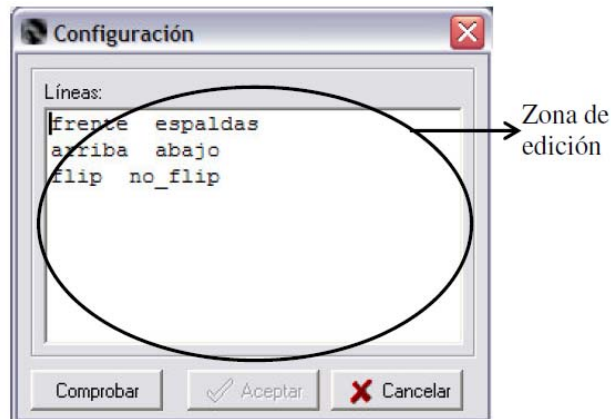


Figura 37. Ventana de configuración

El número de configuraciones que pueda adaptar un robot dependerá fuertemente de su morfología, así, por ejemplo, los robots con una primera articulación rotacional por regla general podrán llegar a su destino de frente o de espaldas, si el resto de las articulaciones son también rotacionales también tendremos más pares de configuraciones.

Par implementar esto en nuestro robot deberemos escribir en la zona activa, la zona de edición de texto, en cada línea cada par de configuraciones posibles del robot. Les daremos el nombre que queramos y los separaremos mediante uno o varios espacios. Cada par de configuraciones deberá ir en una línea y podrán contener cualquier carácter menos el espacio, que está destinado como hemos dicho a actuar de separador.

Así, por ejemplo, para el ejemplo del robot de la figura XX en la primera línea deberemos escribir *'frente espaldas'* (sin las comillas, claro está), o cualquier identificador que queramos, pero no podremos poner, por ejemplo identificadores como *de frente*, porque generaríamos un error. Podemos verlo mejor en la siguiente imagen:

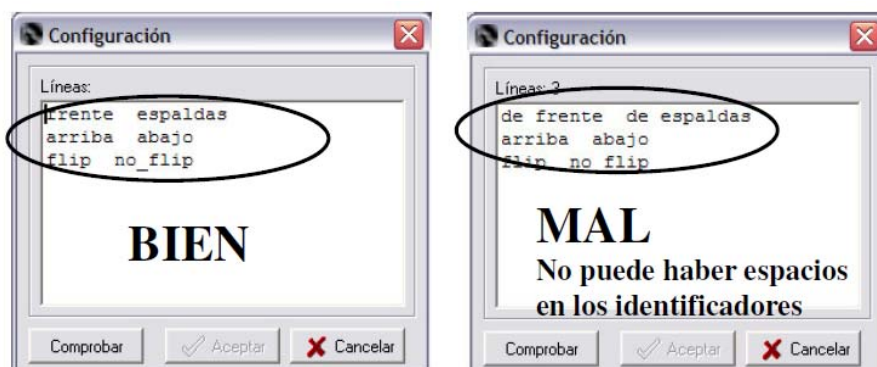


Figura 38. No puede haber espacios en los indicadores

Para que la implementación de las posibles configuraciones sea efectiva deberemos pulsar el botón de aceptar, botón que se encontrará bloqueado hasta que sean revisados los errores mediante el botón comprobar. Si hubiera algún error se nos avisaría mientras que si estuviéramos exentos de errores se aplicarían las configuraciones.



Las posibles configuraciones implementadas podrán ser accesibles deberás ser tenidas en cuenta a la hora de implementar el modelo inverso y serán accesibles bien desde la ventana de guiado o bien desde la aplicación robótica.

### El editor de cinemática inversa

Una vez obtenidas las ecuaciones del modelo inverso (las deberá obtener el usuario, existen variedad de métodos para ello que no explicaremos aquí) deberá implementarlas en la aplicación para poder usar ciertas características en su robot. Para ello se ha habilitado un lenguaje de script de sintaxis muy similar al pascal. Este Script no es más que código fuente que 'compilará' y ejecutará la aplicación principal cada vez que haya que resolverse el modelo inverso, por lo que no se necesitará ningún compilador externo, ni tener acceso al código fuente del programa ni realizar ninguna modificación sobre dicho código. En el script tendremos que programar que resuelvan el modelo inverso en función de la configuración adoptada por el robot. El editor de cinemática inversa tiene la siguiente interfaz:

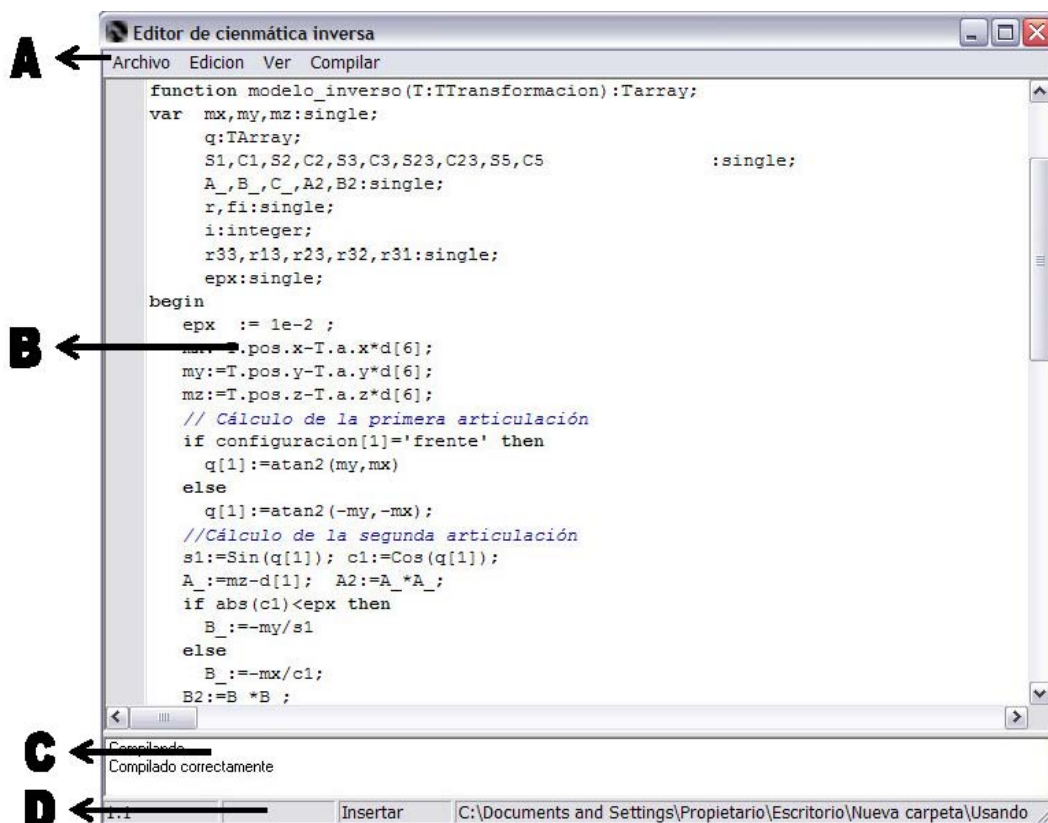


Figura 39. Interfaz de usuario del editor de cinemática inversa

Podemos distinguir los siguientes elementos:

#### A. Menú principal

Desde aquí controlaremos todas las acciones que podremos realizar con el editor.



Disponemos de los siguientes submenús:

Submenú archivo: Desde aquí gestionamos el trabajo con los archivos de código fuente, tenemos las siguientes opciones:

Nuevo: Genera un archivo nuevo, sólo con la cabecera de la función del modelo inverso.

Abrir: Abre un archivo ya existente. La extensión para el código fuente de la cinemática inversa es \*.ci por defecto, aunque si en el diálogo de abrir archivo accionáramos la pestaña de todos los archivos, podremos abrir cualquier fichero de texto.

Guardar y guardar como. Guardan el código fuente en un fichero. La diferencia es que guardar como permite elegir el nombre final del fichero mediante un diálogo estándar mientras que guardar sólo guarda los cambios en el fichero ya existente.

Submenú edición: Desde este menú se tiene acceso a las funciones más comunes en cualquier editor de texto como pueden ser deshacer, rehacer, copiar, cortar, pegar, eliminar o seleccionar todo.

Deshacer: Permite volver atrás en los cambios realizados en el área de edición de texto. Podemos retroceder un gran número de pasos (1024), más que suficiente para revisar y corregir cualquier error que hubiéramos cometido.

Rehacer: Es el paso contrario a deshacer, en caso de que no quisiéramos haber ejecutado el comando deshacer, deberemos ejecutar esta acción.

Copiar: Copiar el texto seleccionado y lo coloca en el portapapeles.

Cortar: Corta el texto seleccionado y lo coloca en el portapapeles.

Pegar: Inserta el contenido del portapapeles.

Eliminar: Borra el texto seleccionado.

Seleccionar todo: Selecciona todo el texto.

Submenú ver. Desde este submenú podemos cambiar la fuente, mediante un diálogo estándar de Windows o podremos hacer visibles los números de las líneas (ver > Ver números de línea). Activando esta posibilidad aparecerá a la izquierda de cada línea el correspondiente número, empezando desde 1.

Submenú compilar. Desde este submenú tenemos acceso a la compilación del modelo inverso.

## **B. Zona de edición de código fuente**

En esta zona deberemos escribir el código fuente que resuelve el modelo inverso del robot. Las palabras reservadas del lenguaje (*if, then, else, while, etc*) se



remarcarán en negrita, lo cual proporciona una mayor claridad a la hora de analizar el código puesto que se mejora la legibilidad del código. Los comentarios pasarán a tener un color azul y estar en cursiva, lo que también favorece la legibilidad del código.

### C. Zona de mensajes

En esta zona se nos mostrarán los mensajes generados (si los hubiera) por el compilador tras la compilación, que podrán ser de dos tipos, avisos y errores, y se explicarán a continuación.

### D. Barra de estado

En esta zona inferior de la pantalla se muestra información referente al estado del texto, en la primera casilla se nos muestran las coordenadas del cursor, en formato fila:columna. En la segunda se nos muestra si el archivo ha sido modificado desde la última vez que se guardó. En la tercera nos indica el modo de escritura, que podrá ser insertar o sobrescribir. Se puede conmutar entre ambos pulsando la tecla Insert. Finalmente, en el panel de más a la derecha se nos muestra el nombre del archivo actual.

### Sintaxis de la función del modelo inverso.

Como ya hemos comentado, el lenguaje utilizado en el script es lenguaje pascal estándar, con todo lo que ello implica. En cuanto a los tipos de datos, aparte de los habituales en el lenguaje pascal (enteros reales, cadenas...) tenemos nuestros tipos de datos específicos a los que podremos acceder desde el script y que harán referencia a localizaciones, variables articulares y configuraciones adoptadas. Veamos a continuación todos los tipos de datos accesibles desde el script:

#### Tipos de datos estándares.

#### Tipos enteros

Tipo	Rango	Formato
Integer	-2147483648..2147483647	Con signo 32 bits.
Cardinal	0..4294967295	Sin signo 32 bits.
Shortint	-128..127	Con signo 8 bits
Smallint	-32768..32767	Con signo 16 bits.
Longint	-2147483648..2147483647	Con signo 32 bits.
Byte	0..255	Sin signo 8 bits
Word	0..65535	Sin signo 16 bits
Longword	0..4294967295	Sin signo 32 bits.

Existen dos tipos base, de 32 bits que son integer y cardinal. Las operaciones aritméticas entre tipos enteros por regla general devolverán un valor de tipo integer. A efectos resultan equivalentes los tipos Integer y Longint, y Cardinal y



Longword. No hay problemas de asignación de diferentes tipos mientras el valor esté dentro del rango adecuado.

### Tipos reales:

Tipo	Rango	Cifras	Tamaño
Single	$1.5 \times 10^{-45} \dots 3.4 \times 10^{38}$	7-8	4 bytes
Double	$5.0 \times 10^{-324} \dots 1.7 \times 10^{308}$	15-16	8 bytes
Extended	$3.6 \times 10^{-4951} \dots 1.1 \times 10^{4932}$	19-20	10 bytes
Currency	-922337203685477.5808.. 922337203685477.5807	19-20	8 bytes

Todos estos datos tienen una notación de coma flotante, excepto el tipo de dato currency, que es representado en coma fija, es representado internamente como un entero de 64 bits con los últimos 4 dígitos, los menos significativos representando las cifras decimales.

Se recomienda encarecidamente el uso del tipo de dato single, de 4 bytes (32 bits), puesto que muchas rutinas algebraicas están optimizadas para este tipo de dato, con lo que la ejecución resultará mucho más fluida.

### Tipos carácter (char):

El tipo char es un tipo que puede contener un solo carácter. Cada uno de estos caracteres puede ser expresado gracias al código ASCII. Un carácter char se debe encerrar entre comillas simples, por ejemplo: 'A' 'b' '\*' '\ ' '5'.

Los tipos de datos carácter son ordinales, lo cual significa que cada carácter representa una posición en una serie ordenada (de 0 a 255). El orden de la letra A es 65, el de la letra B es 66, etc. Para saber el orden basta con mirar una tabla ASCII.

La función predefinida chr permite referenciar todos los caracteres, así, chr(65) equivale a la letra A, chr(66) equivale a la letra B, y así sucesivamente.

### Tipo cadena (string)

Un tipo string (cadena) es una secuencia de caracteres correspondientes al código ASCII, de longitud indefinida, escrito en una línea sobre el programa y encerrado entre comillas simples, por ejemplo: 'cadena', 'otra cadena'. Se permite la cadena de longitud nula y se representa como ''.

### Tipo booleano:

Una variable de tipo booleano puede tomar dos valores true o false (verdadero o falso), y surge como resultado de aplicar los siguientes operadores: =, >, <, >=, <= y <>.

Las operaciones que pueden realizarse con este tipo de datos son **not**, **and**, **or**, **xor**.



### Tipo array (vector)

Este tipo de dato complejo está compuesto por elementos del mismo tipo y sus elementos son accesibles mediante un índice. El tamaño está fijado y deberá especificarse en el momento de declarar una variable de tipo array. Por ejemplo:

```
Type nombre_array=array [1..8] of single;
```

Es un vector de 8 elementos, con índices de 1 a 8. Para acceder a uno de sus elementos nos referiremos al índice, así, el segundo elemento será `nombre_array[2]` y será un tipo de dato `single`.

También puede haber arrays de varias dimensiones o matrices, accesibles desde dos índices. Ejemplo `array[1..numero_filas,1..numero_columnas]` of `single`.

### Tipo registro (record)

Es un tipo de datos estructurado que consta de un conjunto de elementos que pueden ser del mismo tipo o de tipos diferentes..

Los componentes de un registro se denominan campos. Cada campo tiene un nombre, llamado identificador de campo, que es un identificador elegido por el programador cuando se declara el tipo de dato `record`. Ejemplo:

```
type  
  
Mi_registro = record  
    Campo1:single;  
    Campo2:integer;  
end;
```

Para acceder al campo primero deberemos escribir: `Mi_registro.campo1`

### Tipos de datos específicos accesibles desde el script:

#### Coordenadas de un punto:

Para referirnos a un vector tridimensional disponemos del dato `TCoordenadas`, cuya declaración es la siguiente:

```
type  
  
TCoordenadas = record  
    x,y,z:single;  
end;
```

Así, para referirnos a la componente  $x$  de un punto  $P$  ( $P$  es de tipo `TCoordenadas`) deberemos escribir `P.x`





### Transformación homogénea:

Se ha definido el tipo de dato *TTransformation* que representa una matriz de transformación homogénea, que nos permitirá decir la posición y la orientación:

```
type
TTransformation=record
    pos,n,o,a:TCoordenadas;
end;
```

Así, para una transformación  $T$ , de tipo *TTransformation*, nos referiremos a la componente  $x$  de la posición mediante  $T.pos.x$ , o para referirnos a la componente  $n_y$  de la submatriz de rotación  $noa$  mediante  $n.y$ .

### Variables articulares.

Para referirnos al valor de las articulaciones del robot se ha definido el tipo de dato *TArray* de la siguiente manera:

```
type
TArray = array [1..max_articulaciones] of single;
```

Donde *max\_articulaciones* es una constante, su valor es 10 y representa el número máximo de articulaciones que puede tener un robot. Es suficientemente alto como para poder definir cualquier tipo de robot, por muy redundante que éste sea.

De manera que para referirnos al primer elemento, valor de la primera articulación lo haremos mediante  $q[1]$ , siendo  $q$  una variable de tipo *Tarray*.

### Configuración adoptada por el robot.

El robot puede adoptar diferentes configuraciones, y para referirnos a la configuración actual se ha definido el tipo de dato *TUnaConfig* de la manera siguiente:

```
type
TUnaConfig=array[1..5]of string;
```



### Descripción de la función modelo\_inverso:

Cada vez que necesite resolverse el modelo inverso se realizará una llamada a esta función definida mediante el script, que tiene la siguiente cabecera y que no deberá modificarse bajo ningún concepto:

```
function modelo_inverso(T:TTransformation):TArray;
```

El parámetro  $T$ , de tipo *TTransformación* que se le pasa a la función se refiere a la localización deseada del elemento terminal, y deberemos devolver (para que una función devuelva un valor deberemos usar la variable interna *Result*) un tipo de dato *TArray* con las variables articulares calculadas. Si quisiéramos saber la posición  $x$  del punto final, escribiremos  $T.pos.x$ , mientras que si por ejemplo quisiéramos que la primera articulación tomara el valor de 13 deberíamos escribir  $Result[1] := 13$ ;

Por otra parte también tendremos acceso a los parámetros de Denavit Hartenberg mediante una serie de variables globales de tipo *TArray*, a los valores máximos y mínimos de la articulaciones, también mediante variables globales de tipo *Tarray*, y a la configuración con la que deberá resolverse el modelo inverso, de tipo *TUnaConfig*:

```
var  
a,d,alfa,tita:TArray;  
configuracion:TUnaConfig;  
qmax,qmin:Tarray;
```

De modo que si queremos conocer el valor del parámetro  $d$  de la cuarta articulación nos referiremos a él como  $d[4]$ , y si queremos saber el valor máximo que puede tomar la primera articulación lo haremos mediante  $qmax[1]$ .

Para la configuración, como explicamos en su momento en su ventana correspondiente escribíamos en cada línea cada par de posibles configuraciones que pudiera adoptar el robot. Mediante la ventana de guiado o mediante la aplicación robótica habremos adoptado una configuración, y ahora, en la programación del modelo inverso tendremos que tenerla en cuenta mediante la variable global *configuracion*. Si en la ventana correspondiente teníamos  $n$  líneas ello significa que tenemos  $n$  pares de configuraciones posibles, y tendremos acceso mediante los  $n$  elementos del vector *configuracion*. Así, por ejemplo si en primer par de configuraciones (la primera línea) definido en su ventana correspondiente era:

*Frente espaldas*

Si el robot ha adoptado la primera de las configuraciones el valor de *configuracion[1]* será la cadena 'Frente', y así para el resto de las posibles configuraciones. Deberemos tener cuidado en respetar mayúsculas y minúsculas.

Aparte de todo lo comentado también tendremos acceso desde el script a una serie de funciones adicionales, funciones algebraicas y trigonométricas que nos serán de gran ayuda:



```
function Sin(X: Single): Single;           // Seno
function Cos(X: Single): Single;         // Coseno
function Sqrt(X: Single): Single;        // Raíz
function Atan2(x,y:single):single;      // Arcotangente y cuadrante
function ACos(x:single):single;         // Arco Coseno
function degrees(rads:single):single;   // Convierte a grados
function rads(degrees:single):single;   // Convierte a radianes
```

Así pues, ya tenemos definido todo lo necesario para programar las ecuaciones del modelo inverso en el script.

Tenemos que remarcar también que aunque posteriormente podremos cambiar el sistema de referencia del mundo (comúnmente conocido como referencia BASE o WORLD) y el sistema de referencia asociado al elemento terminal, en el caso en que tengamos una herramienta (Sistema de referencia comúnmente conocido como TOOL), no se deberá tener en cuenta en la programación del modelo inverso del robot, las transformaciones necesarias se realizarán de manera automática en el momento en que cambiemos estas referencias, cosa que podremos hacer en el siguiente módulo.

Como último apunte, y no por ello menos importante, más bien al contrario, todas las funciones trigonométricas trabajan en radianes, mientras que la función necesita que devolvamos los ángulos en grados, así que será necesaria una conversión a grados una vez que tengamos calculados los valores articulares mediante las funciones que se ofrecen.

### **Compilación del modelo inverso.**

Bien desde el menú principal (Menú principal > Compilar > Compilar el modelo inverso) o desde la ventana principal de la aplicación (Panel lateral > Panel CI > Compilar modelo Inverso) podremos compilar el script del modelo inverso. La zona de mensajes inferior nos mostrará las advertencias necesarias o los errores cometidos. Las advertencias más habituales suelen referirse a variables declaradas que no han sido utilizadas. Se nos mostrará cada advertencia mediante la palabra [Hint] y a continuación la línea en la que ocurre la advertencia y el tipo de advertencia. Si se ha compilado satisfactoriamente aparecerá el mensaje *Compilado correctamente*, en caso contrario habremos cometido algún error se indicará el error en la zona de mensajes y el editor se situará automáticamente en la línea en la que se ha producido el error, que quedará remarcada en rojo.

Una vez que hayamos definido el modelo inverso ya podremos usar todas las funciones del robot que conllevan especificar su destino en coordenadas operacionales.

### **La ventana de Guiado del robot.**

Mediante esta ventana, a la cual ya hecho alguna referencia, se podrá efectuar el guiado del robot, que podrá ser un guiado articular o guiado cartesiano.

La ventana presenta la siguiente interfaz, que será accesible tanto desde este módulo como del posterior módulo de construcción y explotación de escenarios.



Figura 40. Ventana de guiado

Lo primero que cabe destacar es que disponemos de tres pestañas, la primera, con el nombre *Guiado* nos servirá para realizar el guiado propiamente dicho. La segunda, con nombre *Ir a*, nos permitirá especificar un destino al robot, tanto en coordenadas articulares como en el espacio de la tarea. La tercera pestaña, de nombre *config*, nos servirá para especificar la configuración del robot con la que se resolverá el modelo inverso.

#### Modo guiado:

En la parte superior tenemos un selector mediante el cual podremos seleccionar el tipo de guiado que queremos, guiado articular (opción *articular*), o guiado en coordenadas cartesianas, que a su vez podrá ser respecto al sistema de referencia absoluto del mundo (opción *mundo*) o bien relativo a la herramienta (opción *herramienta*). En caso de guiar en coordenadas articulares la interfaz pasará a mostrarnos una serie de barras (una por cada articulación) con las que podremos mover cada una de las articulaciones del robot. A la izquierda y a la derecha de cada barra nos aparecerá el valor máximo y el mínimo que puede tomar dicha articulación, mientras que el valor actual aparecerá en la barra de estado que se encuentra en la parte inferior de la interfaz principal.

Si seleccionamos un guiado cartesiano podremos modificar la posición del elemento terminal modificando los parámetros de posición y los de orientación mediante los ángulos de Roll Pitch Yaw. Se modificarán de manera incremental pudiendo elegir en todo momento el incremento. Para utilizar este tipo de guiado deberemos haber compilado antes el modelo inverso correctamente, en caso contrario nos saldrá un mensaje de advertencia indicándonos que debemos proceder a su implementación.

#### Modo Ir a:

En este modo también podremos seleccionar el modo de control que deseemos mediante un selector, que nos permitirá elegir entre articular o mundo, en ambos casos le indicaremos un destino al robot, que realizara un movimiento coordinado hacia dicho destino.

Si estamos en modo *articular* nos aparecerá una tabla en la que deberemos especificar el valor que queremos que alcance cada articulación mientras que si estamos en modo *mundo* el destino lo deberemos ofrecer en coordenadas de la tarea. En caso de que no se pueda llegar al destino especificado se nos avisará, indicando el valor al que se quería llegar y el valor límite que puede alcanzar dicha articulación. De igual manera que antes, para poder especificar las coordenadas de



destino en coordenadas cartesianas deberemos haber definido antes el modelo inverso. Por último también se ha habilitado un botón (botón *Reposo*) para poder llevar al robot a su posición de reposo.

### **Modo configuración**

Aparecerán en este modo una serie de controles para poder especificar la configuración con la que se resolverá el modelo inverso. Como ya se ha explicado suficientemente antes, podremos llegar al mismo destino de varias formas. Se habilitarán una serie de selectores, tantos como pares de configuraciones hallamos definido en su momento, y mediante cada selector podremos elegir la configuración que adoptará el robot.



## 5 EL CONSTRUCTOR DE ESCENARIOS

El constructor de escenarios es el más importante de los tres módulos que componen el simulador. Como su propio nombre indica, permite la construcción de escenarios robóticos añadiendo a la escena robots y objetos creados previamente con los otros dos módulos. El módulo permite, no sólo la construcción del escenario sino también su explotación posterior. De esta forma, podemos dotar a los robots de movimientos mediante guiado, registro de puntos para crear una secuencia o por medio de programación textual.

En este apartado se describirá el constructor de escenarios, haciendo hincapié en sus características y funcionalidades. Una vez acabado este apartado el lector tendrá una idea general sobre lo que es **RobotScene** (versión multi-robot), concretamente del último y más importante de sus módulos, y sobre las capacidades técnicas y de simulación que ofrece.

### 5.1 DESCRIPCIÓN DE LA INTERFAZ DE USUARIO

En esta descripción de la interfaz, se describirán las posibilidades del constructor de escenarios de manera que un usuario no familiarizado con el programa pueda entender a grandes rasgos para qué sirve y cómo funciona, así como las opciones que ofrecen los menús y barras de herramientas. La imagen siguiente muestra la interfaz general de usuario:

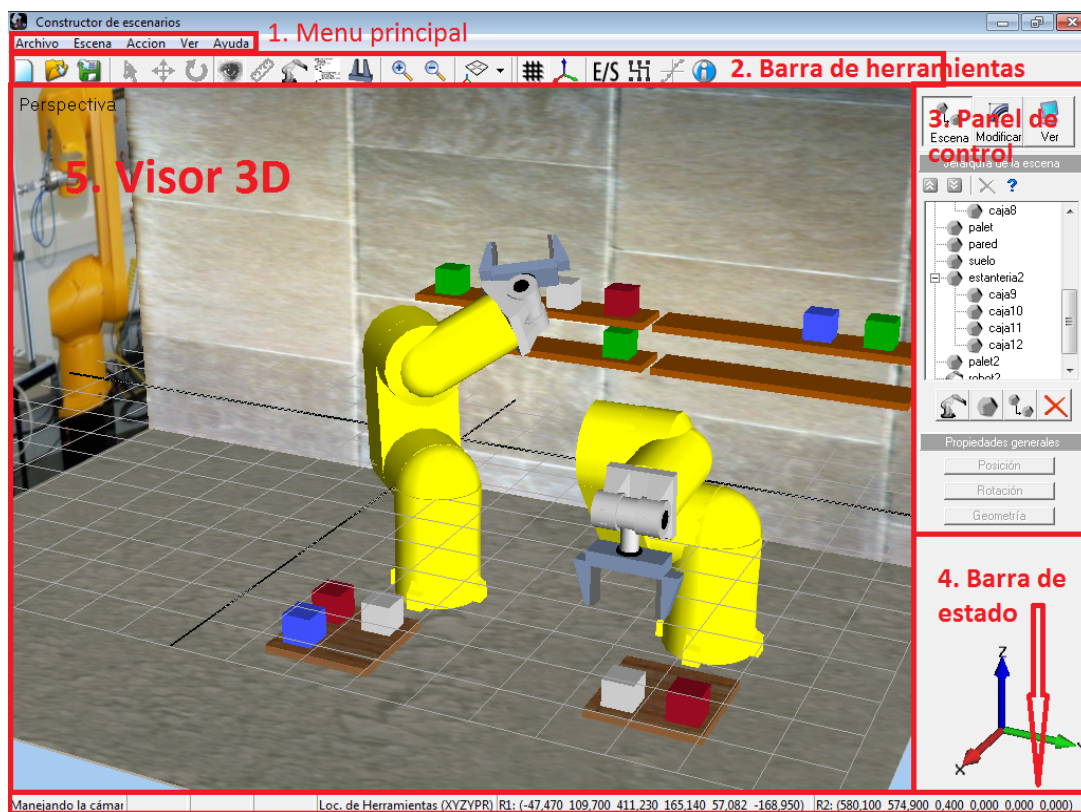


Figura 41. Captura de la interfaz de usuario



Tal como se puede observar, el módulo constructor de escenarios presenta las siguientes zonas:

## 1. Menú principal.

Desde el menú principal podemos acceder a prácticamente todas las funcionalidades del programa. Está compuesto por los siguientes elementos:

**Archivo:** En esta opción del menú el usuario puede cargar un archivo de escenario (.esc), guardar el escenario, crear un nuevo escenario o salir del programa.

**Escena:** Mediante esta parte del menú el usuario tiene la opción de añadir elementos al escenario, el usuario puede añadir un nuevo sólido, añadir un robot, o eliminar el objeto seleccionado.

**Acción:** Este menú permite al usuario seleccionar los diferentes modos de operación. Por un lado nos encontramos las acciones relativas a los objetos del escenario, teniendo la posibilidad de seleccionar objetos, trasladar objetos y rotar objetos. Por otro lado encontramos acciones relativas a gráficos como son manejar la cámara y medir distancias en el escenario. Finalmente, el usuario puede realizar acciones relativas a los robots. Dichas acciones son *guiar robot*, *programar robot* y *añadir una herramienta al robot*.

**Ver:** Desde este menú se acceden a diferentes opciones referentes a la visualización de la escena. Las dos primeras opciones *acercar* y *alejarse* permiten ajustar la distancia de la cámara con el escenario actuando como zoom. También se encuentran en este menú diferentes opciones gráficas relativas al modo de representación de los objetos: *suavizado*, *sin suavizar* y *alambres*. *Suavizado* permite ver los objetos con un buen acabado gráfico mientras que *alambres* deja en segundo plano las opciones gráficas para mejorar el rendimiento. Mediante el menú *ver* se posibilita la opción de cambiar de vista pudiendo elegir entre cuatro vistas diferentes: perspectiva, cenital (XY), lateral (YZ), frontal (XZ). Las últimas opciones de este menú permiten abrir las diferentes ventanas que componen el constructor de escenarios: ventana de registro, ventana de entradas y salidas (E/S) y la ventana de cableado.

**Ayuda:** Menú informativo que presenta las siguientes opciones: *acerca de* permite al usuario informarse sobre los créditos de los creadores del programa, *configuración* permite al usuario cambiar ciertas opciones gráficas como el antialiasing, el color de fondo o la profundidad de color.

## 2. Barra de herramientas





















Como suele ser habitual en los programas informáticos, el constructor de escenarios cuenta con una poderosa barra de herramientas, que a disposición del usuario, permite realizar todas las acciones del constructor de escenarios.



Figura 42. Barra de herramientas del constructor de escenarios



A continuación se detalla individualmente la acción que realiza cada uno de los botones.

-  Crea un nuevo escenario.
-  Carga un escenario previamente guardado.
-  Guarda el escenario actual.
-  Selecciona objetos del escenario.
-  Traslada objetos en el escenario.
-  Rotar objetos del escenario.
-  Permite al usuario manejar la cámara libremente en el escenario
-  Mide distancias en el escenario.
-  Guía un robot del escenario.
-  Abre la ventana de programación al pinchar sobre el robot.
-  Permite añadir o eliminar una herramienta a un robot.
-  Acerca la cámara (zoom in).
-  Aleja la cámara (zoom out).
-  Selecciona las diferentes vistas de la cámara: perspectiva, cenital (XY), lateral (YZ), frontal (XZ).
-  Activa u oculta la rejilla en el escenario.
-  Permite observar los ejes de posición y orientación de los objetos del escenario.
-  Permite observar la ventana de entradas y salidas de los robots.
-  Activa la ventana de cableado.
-  Activa la ventana de registro por puntos cuando se esta guiando un robot.
-  Permite observar información relativa al driver OpenGL y tarjeta gráfica.

### **3. Panel de control**

El constructor de escenarios de **RobotScene** cuenta con un panel de control situado en la parte derecha de la ventana. Este panel de control sirve principalmente para controlar la creación de escenario.





El panel de control se compone realmente de tres subpaneles. Para cambiar entre subpaneles se han habilitado tres botones que se muestran en la siguiente imagen:



Figura 43. Botones *Escena*, *Modificar* y *Ver*, creados para alternar entre subpaneles.

A continuación se detalla la forma y contenido de cada uno de los subpaneles que conforman el panel de control.

### Panel Escena

Es el más importante de los tres, por sus funcionalidades; es también el más utilizado. Su función principal es la de creación del escenario. Está dividido en dos partes.

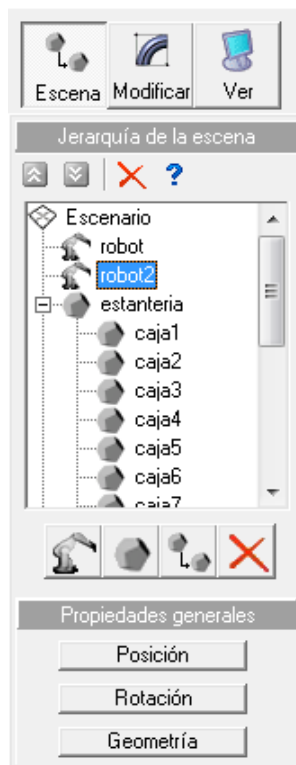








Figura 44. Imagen del panel *Escena*.

La primera de estas dos partes está creada bajo el título de *Jerarquía de la escena*, como indica su nombre, permite establecer el contenido del escenario de forma jerarquizada, estableciendo una preferencia de los elementos que componen el escenario de forma arbórea.

Más adelante se hablará con más detalle acerca de la jerarquía de la escena. Por el momento, es suficiente conocer para qué sirve cada uno de los botones del panel.

- Permite subir de posición un elemento del árbol dentro de su jerarquía.
- Permite bajar de posición un elemento del árbol dentro de su jerarquía.



-  Elimina un elemento del árbol de jerarquía, y por tanto, del escenario.
-  Botón ayuda.
-  Añade un robot a la escena.
-  Añade un objeto a la escena.
-  Añade un hijo a un objeto de la escena.
-  Borra completamente el escenario.

La segunda de las partes se denomina *Propiedades generales* y permite ajustar numéricamente la posición y la orientación de los objetos del escenario, así como cambiar su geometría.

### Panel Modificar

El panel *Modificar* nos permite activar efectos gráficos como sombras o cielo y permite activar o desactivar la detección de colisiones.

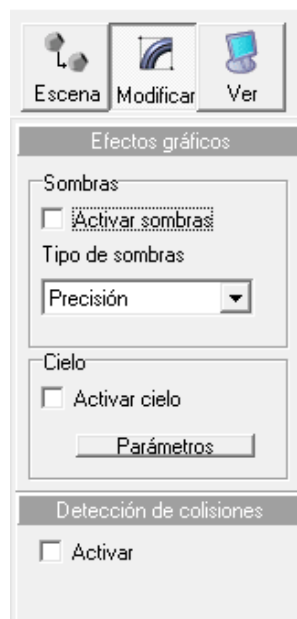


Figura 45. Panel *Modificar*.



## Panel Ver

El panel *Ver* nos permite activar efectos gráficos relativos a la cámara. La luz puede activarse y desactivarse, así como, cambiar su posición y su color. Otra de las opciones de este panel es cambiar la vista de la cámara.

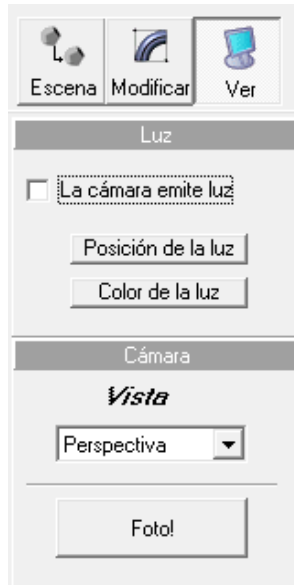


Figura 46. Imagen del panel *Ver*.

Adicionalmente el botón que se muestra a continuación permite capturar la imagen del visor de perspectiva en tres dimensiones para posteriormente guardarlas en el disco duro:



Figura 47. Botón que permite capturar la imagen del visor 3D.

## 4. Barra de estado

La barra de estado del constructor de escenarios muestra en su parte lateral izquierda información relativa al modo de operación actual del escenario. En la zona central muestra la información relativa al objeto seleccionado, es decir, su orientación y posición. Por último en la parte lateral derecha hay un gran espacio reservado para mostrar la orientación y posición de la herramienta del robot.

Moviendo objetos	X: -172,9	Y: 338,2	Z: 151,8	Loc. de Herramientas (XYZVPR)	R1: (236,200 31,600 615,000 180,000 0,000 0,000)	R2: (217,500 570,400 615,000 180,000 0,000)
------------------	-----------	----------	----------	-------------------------------	--	---

Figura 48. Barra de estado del constructor de escenarios.

## 5. Visor 3D

El visor 3D ocupa gran parte de la ventana principal del constructor de escenarios. Dicho visor es la pieza clave del constructor de escenarios porque nos muestra la representación visual en 3 dimensiones del escenario, en otras palabras, es donde se observará el movimiento de los robots de la aplicación. El usuario puede interactuar con dicho visor para realizar ciertas operaciones con los objetos del escenario, como podría ser moverlos o rotarlos. Aparte del escenario propiamente dicho se pueden ver otros elementos como ejes o rejillas que facilitan la comprensión de la escena.

El visor 3D es utilizado por tanto, como elemento para la visualización y modificación del escenario, y a partir de él, es posible crear y explotar el escenario.




## 5.2 CREACIÓN DEL ESCENARIO

Como se ha comentado con anterioridad el constructor de escenarios sirve fundamentalmente para crear escenarios que posteriormente serán utilizados en una tarea robótica. En este apartado se profundiza en las diferentes opciones para crear el escenario, que deberá hacerse inevitablemente utilizando los objetos y robots creados con los otros dos módulos de **RobotScene**, el constructor de objetos, y el constructor de robots.

### Añadir objetos

Una parte fundamental en la creación del escenario es poder añadir objetos que posteriormente nuestro robot tendrá la capacidad de manipular. Se pueden añadir objetos al escenario de dos diferentes maneras:

→ *Menú principal > Escena > Nuevo sólido*

→ *Panel Escena > Jerarquía de la escena > Botón añadir objeto* .

Utilizando cualquiera de estas dos formas el programa nos abrirá la siguiente ventana:

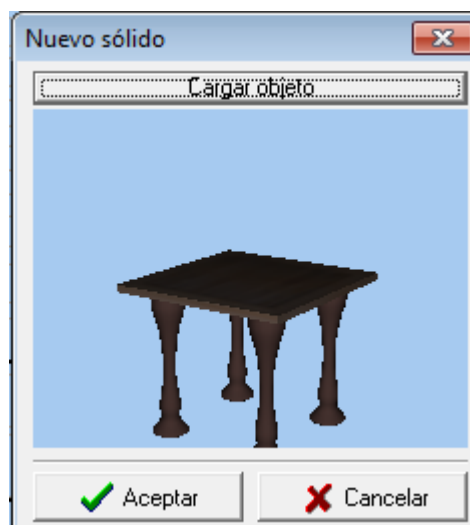


Figura 49. Constructor de escenarios. Ventana para cargar sólidos.

Al pinchar con el ratón en la opción *Cargar objeto* se abrirá un explorador que permite seleccionar objetos almacenados en el disco duro. Los objetos que se carguen deben haber sido creados ineludiblemente mediante el correspondiente constructor de objetos, por tanto, el formato de estos archivos debe ser *.3d*. Una vez observemos en la ventana la geometría del objeto cargado debemos aceptar y el objeto formará ya parte del escenario y se podrá observar en el visor 3D. Cada sólido añadido al escenario tiene asociado un sistema de referencia que debemos tener en cuenta a la hora de crear el escenario.




Aunque no es recomendable, es posible añadir objetos al escenario sin geometría. De esta manera se consiguen puntos en el escenario que pueden ser útiles a la hora de crear una trayectoria por puntos de registro.

### Añadir robots

Para que un robot pueda manipular los sólidos que formen parte del escenario debemos incluir al menos un robot en el escenario. Añadir un robot al escenario se realiza de forma similar a añadir un objeto. Podemos añadir un robot al escenario de las siguientes formas:

→ *Menú principal > Escena > Nuevo robot*

→ *Panel Escena > Jerarquía de la escena > Botón añadir robot* 

Utilizando cualquiera de estos dos procedimientos el programa nos abrirá la siguiente ventana:

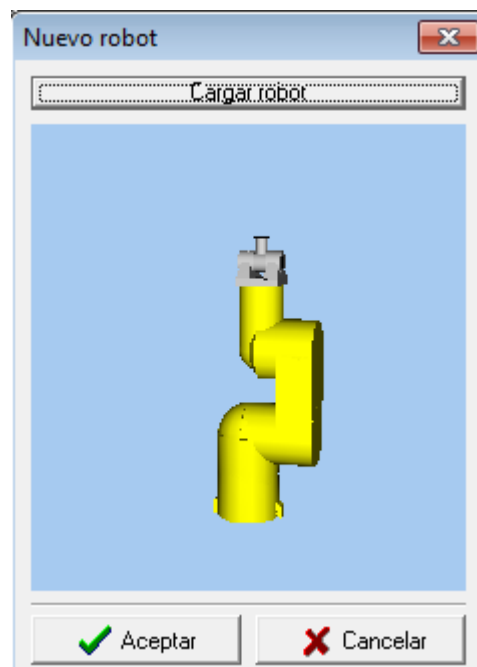


Figura 50. Constructor de escenarios. Ventana para cargar robots.

Si pinchamos con el ratón en la opción *Cargar robot* se abrirá un explorador que permite seleccionar robots del disco duro. Los robots que carguemos han de haber sido creados ineludiblemente mediante el correspondiente constructor de robots, por tanto, el formato de estos archivos debe ser *.rob*. Cuando se observe en la ventana la geometría del robot cargado, se debe y el robot formará ya parte del escenario y aparecerá representado en el visor 3D.


Como ya se ha comentado para los objetos, cada robot añadido al escenario tiene asociado un sistema de referencia que debemos tener en cuenta a la hora de crear el escenario. El sistema de referencia del robot es coincidente con el sistema de referencia absoluto, conocido en robótica como sistema de coordenadas mundo. El constructor de escenarios nos da la posibilidad de trasladar y rotar el robot como pasaba con los objetos, cuando esto ocurra el



programa realizará automáticamente el cambio del sistema de referencia mundo. Lo habitual en robótica es cambiar la referencia mundo en el programa robótico mediante la instrucción WORLD.

### Añadir una herramienta al robot

Generalmente nuestro robot necesitará de una herramienta para realizar la tarea robótica asignada. Para incluir una herramienta en un robot en el constructor de escenarios, es necesario que se haya cargado en la escena al menos un robot. Se añade una herramienta a un robot accediendo al modo de operación *Añadir herramienta*. Es posible hacerlo de las siguientes formas:

- *Menú principal > Acción > Añadir una herramienta al robot*
- *Barra de herramientas > Botón añadir una herramienta al robot.* 

Una vez que se haya accedido a este modo de operación, se selecciona el robot al cual se desea añadir una herramienta. Para seleccionarlo se pincha sobre el robot en el visor 3D o sobre el nombre del robot en la jerarquía de la escena en el panel de la zona derecha del constructor de escenarios. El programa nos mostrará la siguiente ventana:



Figura 51. Constructor de escenarios. Ventana para añadir herramienta.

Al pinchar con el ratón en la opción *Cargar herramienta* se abrirá un explorador para seleccionar las herramientas del disco duro. Sólo es posible cargar aquellas que han sido creadas previamente mediante el correspondiente constructor de robots; por tanto, el formato de estos archivos debe ser *.herr*. Cuando la ventana muestre la geometría de la herramienta cargada, se acepta y la herramienta formará ya parte del robot y se podrá visualizar en el visor 3D.

Si el robot ya tenía colocada una herramienta al seleccionar el robot en el modo de operación de añadir herramientas, la ventana cargará



automáticamente la geometría de la herramienta. Si se desea eliminar la herramienta tan sólo es necesario pulsar el botón *Eliminar herramienta*.

### Jerarquía de la escena

Como se ha comentado en la descripción de la interfaz, el constructor de escenarios posee en la zona lateral derecha de la ventana un panel mediante el que se puede modificar el escenario y la jerarquía de la escena. Para ello se utiliza el correspondiente *Panel Escena*:

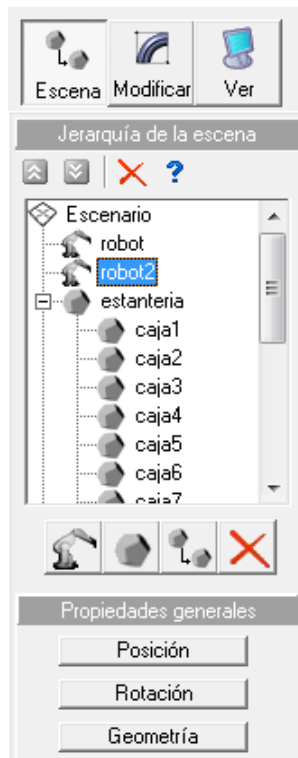




Figura 52. Constructor de escenarios. Panel escena.

Este panel muestra la jerarquía de la escena de forma arbórea. El primer elemento del panel es el escenario que representa la referencia absoluta. A partir de él colgarán todos los demás elementos del escenario. Mediante el panel pueden añadirse robots, objetos e hijos. Los robots únicamente pueden colgar del escenario y no puede haber ningún objeto que cuelgue de los robots. La jerarquía de la escena puede modificarse arrastrando los objetos. Como es lógico no puede arrastrarse un objeto a otro de sus hijos. Este panel permite cambiar el nombre de los objetos haciendo doble click en el nombre del elemento que queramos renombrar.

La estructura jerarquizada de los elementos del escenario acarrea ciertas consecuencias. A la hora de trasladar los objetos, si se hace mediante el visor 3D se modificará su posición absoluta y la de sus hijos. Si se hace numéricamente mediante el correspondiente botón se modificará la posición con respecto al padre.

Para organizar los diferentes elementos se utilizan los botones:  . Su uso permite cambiar la posición de los elementos pero no altera la jerarquía. Aunque no tenga efectos prácticos directos, es posible de esta manera ordenar los elementos al gusto del programador. Adicionalmente, el programa dibujará en pantalla los elementos en orden vertical y descendente de manera que en algunas ocasiones pueden provocar efectos visuales no deseados; por tanto, puede ser necesaria su recolocación en el árbol.



### 5.3 EXPLOTACIÓN DEL ESCENARIO

Un escenario robótico se compone al menos de un robot y de varios objetos simples que el robot tendrá capacidad para manipular. Cuando se habla de explotación del escenario se alude a la capacidad de entablar comunicación con el robot para que realice una tarea concreta; de este modo, el robot se moverá en el escenario robótico y estará en condiciones de manipular los objetos del escenario, según las instrucciones que se le hayan dado.

La explotación del escenario en **RobotScene** permite, como en una situación real, guiar un robot, realizar un registro de trayectorias o programar una tarea robótica mediante lenguaje textual. Otra de las acciones que se puede llevar a cabo en la parte de explotación del escenario es comunicarse con los robots mediante sus entradas y salidas binarias que pueden ser monitorizadas en todo momento, constituyendo la herramienta perfecta para posibilitar la comunicación entre robots.

#### *Guiado de robots*

Una de las formas de explotar el escenario y de dotar de movimiento a los robots es mediante la herramienta de guiado. En **RobotScene** esta herramienta intenta emular las funciones básicas que se pueden realizar con la botonera de guiado de un robot industrial. En el módulo constructor de robots ya era posible realizar el guiado del robot, mediante la correspondiente ventana de guiado.

En el constructor de escenarios se puede acceder al modo de operación guiar robots de dos formas:

→ *Menú principal > Acción > Guiar robot*

→ *Barra de herramientas > Botón guiar robot* 

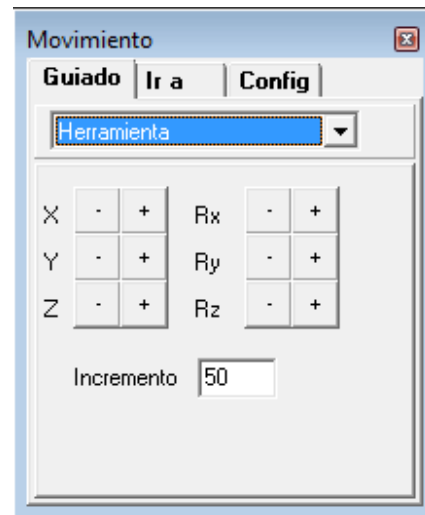
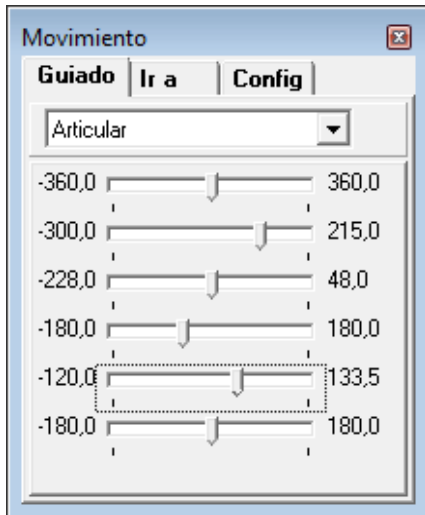
Una vez dentro de este modo de operación basta con pulsar con el botón izquierdo del ratón en el robot que deseamos guiar, o pinchar en su nombre en la jerarquía de la escena. De esta manera se abrirá la ventana de guiado de robots. Una vez abierta la ventana de guiado puede ser utilizada para varios robots, seleccionando el nuevo robot a guiar al pulsar el botón izquierdo del ratón sobre el robot o en la jerarquía de la escena.

La ventana de guiado es una simulación de una botonera de guiado de un robot industrial, por tanto ofrecerá distintas formas de guiado de los robots.

La primera pestaña de la ventana de guiado muestra la función de guiado que puede ser articular o en coordenadas del mundo real.

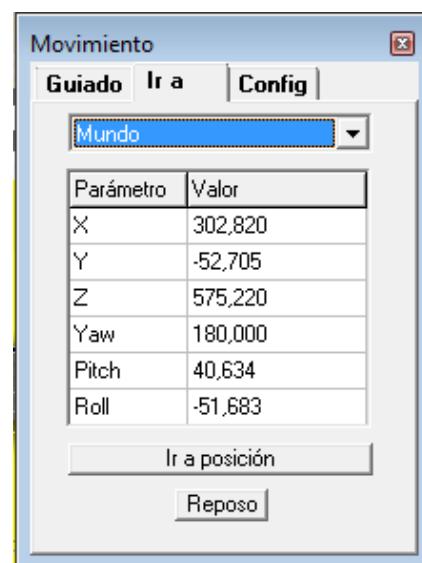
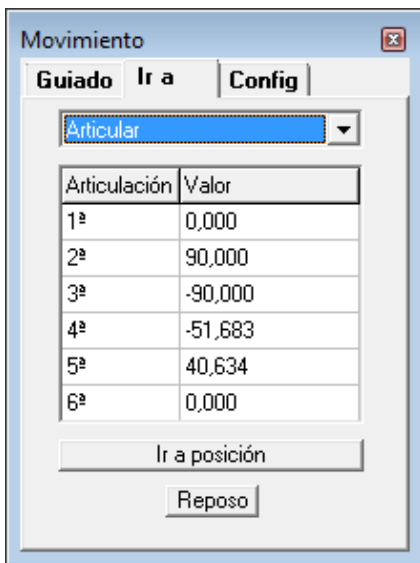
Con el guiado articular es posible guiar, usando el ratón, a cada articulación del robot hacia un valor concreto. En el guiado mediante coordenadas del mundo real es factible guiar el robot - tomando como referencia la absoluta o la de la propia herramienta del robot - aplicando pequeños incrementos en las posiciones y orientaciones. A continuación se muestran las imágenes de la ventana de guiado correspondientes a estos tipos de guiado:





Figuras 53 y 54.- Ventana de guiado: Guiado articular (izquierda) y en coordenadas mundo (derecha)

La segunda de las pestañas se denomina *Ir a*. Su uso permite guiar al robot a una posición y orientación concreta, bien mediante coordenadas articulares o en coordenadas mundo. En esta pestaña también está incluida la opción de ir a las coordenadas de reposo que fueron previamente establecidas en la construcción del robot.



Figuras 55 y 56.- Ir a. Coordenadas articulares (izquierda) e Ir a. Coordenadas mundo (derecha)

Por último la tercera pestaña permite elegir la configuración del robot:

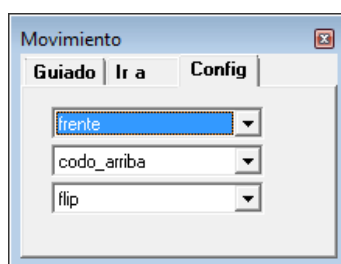


Figura 53. Ventana de Guiado. Pestaña de configuración.




### Registro de trayectorias

Una de las formas más básicas de programar una tarea robótica es la del registro de trayectorias. Esta forma de programar consiste en guiar al robot mediante la botonera - ventana de guiado en nuestro caso - hacia puntos concretos que serán almacenados para posteriormente cubrir una trayectoria que el robot será capaz de reproducir. Tiene sus orígenes en las aplicaciones de teleoperación (industria submarina, industria nuclear, etc). Actualmente es utilizado para aplicaciones como la descarga de máquinas o de soldadura por puntos.

Es un método para programar robots fácil y sencillo, para el que no se necesitan conocimientos de informática ni de programación. Sólo basta con habituarse al control del robot mediante la botonera correspondiente e ir registrando una serie de puntos.

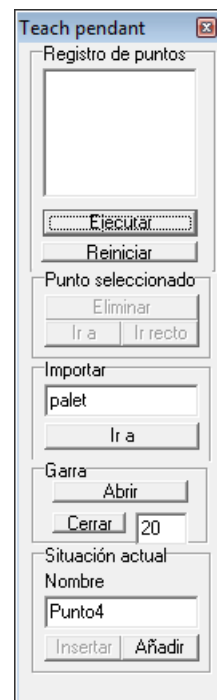
Para la implementación del registro de trayectorias en **RobotScene** se decidió utilizar la situación de la herramienta, es decir, su posición y orientación en vez de las posiciones articulares como se suele hacer en la industria robótica. La ventaja de recoger la situación del robot en vez de sus coordenadas articulares es que permite la portabilidad de dicha trayectoria. Si el registro de la trayectoria fuera tomado mediante coordenadas articulares, no se podría utilizar dicha trayectoria en otro robot diferente al que fue creado. En cambio la desventaja que presenta hacerlo de esta manera en un entorno industrial es que el robot tendría poca o nula adaptación a cambios en el escenario.

Para acceder a la ventana que nos permite realizar el registro de trayectorias en **RobotScene**, se debe entrar en el modo de operación de guiado y haber seleccionado previamente un robot. Una vez hecho esto, se activarán las opciones existentes en la aplicación para poder acceder a la ventana de programación. Para poder visualizar la ventana que nos permite realizar la trayectoria por puntos, existen dos maneras:

- *Menú principal > Ver > Ventana de registro*
-  *Barra de herramientas > Botón ventana de registro*

Aparecerá entonces en pantalla la ventana de registro de puntos que se muestra en la siguiente imagen

Figura 54. Ventana de registro de puntos.



Gracias a la ventana de registro de trayectorias es posible programar de forma sencilla una aplicación. Para registrar un punto hay que ir, primero a la posición dicho punto. Esto lo podemos realizar mediante la ventana de guiado o bien importando la posición del punto que queremos registrar. Para este último



menester la ventana cuenta con una zona denominada *importar* que permite mover el robot a la posición de un objeto perteneciente al escenario. Tan sólo es necesario escribir el nombre del objeto y pulsar en el botón *Ir a* para que el robot se mueva hacia esa posición.

Situados en el punto que se desee registrar, se debe atender a la zona inferior de la ventana denominada *Situación actual*. Se puede registrar el punto de dos maneras diferentes. Mediante *Insertar* se registrará el punto en la posición anterior al punto seleccionado. Mediante *Añadir* se registrará el punto en la última posición de los puntos registrados.

Los puntos registrados se van actualizando en la zona superior de la ventana denominada *Registro de puntos*. En esta zona se puede *Ejecutar* la trayectoria o eliminar todos los puntos pulsando el botón *Reiniciar*.

La zona denominada *Punto seleccionado* permite al robot moverse a la posición del punto que se encuentre seleccionado y que ha debido ser previamente registrado. Desde esta zona es posible, también, eliminar cualquier punto si se selecciona y se pulsa el botón *Eliminar*.

### ***Programación textual del robot***

Como anteriormente se ha comentado, es posible realizar la programación de los robots incluidos en el escenario mediante el registro de trayectorias de forma simple. No obstante, la forma más usual de especificar la operación de un robot es mediante la codificación textual de una serie de instrucciones que conformarán una aplicación robótica que, posteriormente, el robot será capaz de ejecutar. Esta forma de programar el robot es más compleja que la del registro de trayectorias pero, en cambio, es mucho más potente que la anterior.

Para poder programar cada robot mediante líneas de código o texto se ha implementado en **RobotScene** una ventana de programación por cada robot presente en el escenario.

Para acceder a la ventana de programación es necesario, como es lógico, que el escenario sobre el que se trabaja cuente al menos con un robot. Si se cumple esta condición se activa el modo de operación *programar robots* de dos formas posibles:

→ *Menú principal > Acción > Programar robot*

→ *Barra de herramientas > Botón programar robot*



Una vez dentro dicho modo de operación, se visualiza la ventana de programación de cada robot, tras pulsar sobre el robot deseado en el visor 3D o en el nombre del robot en el árbol de jerarquía. Aparecerá la siguiente ventana que corresponde con el editor de programas:

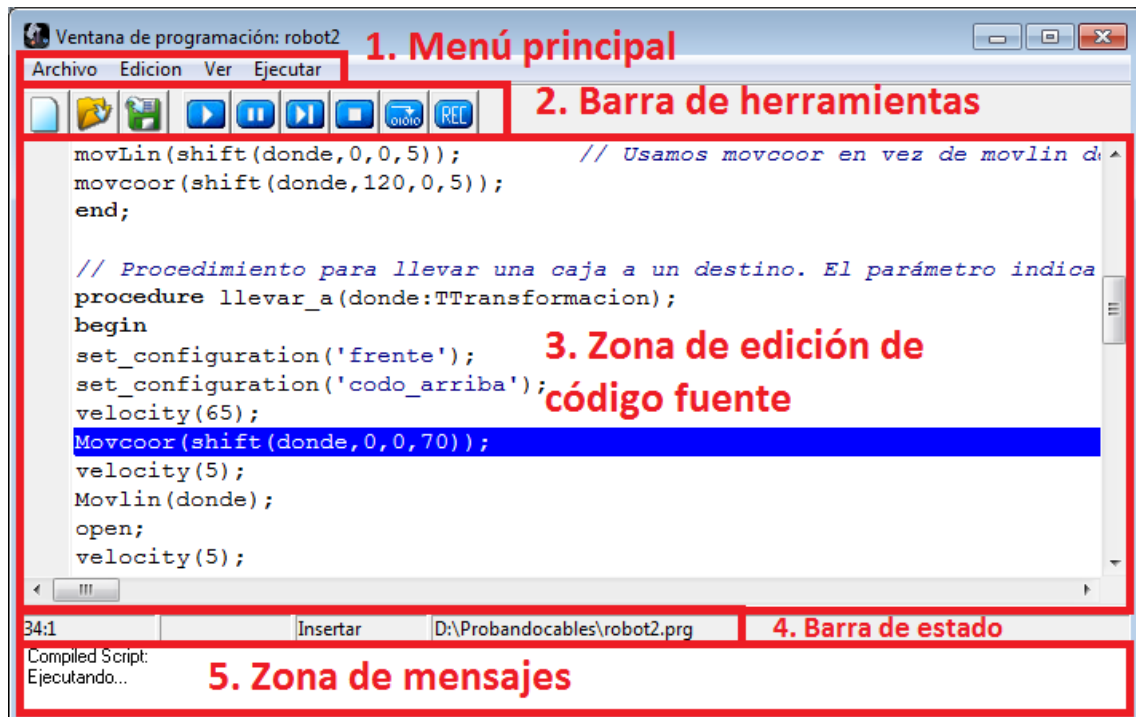


Figura 55. Ventana de programación de robots.

Tal como se observa en la imagen, la ventana de programación consta de las siguientes partes:

### 1. Menú principal.

Desde el menú principal se puede acceder a prácticamente todas las funcionalidades de la ventana de programación. Está compuesto por los siguientes elementos:

**Archivo:** En esta opción del menú el usuario tiene a su disposición los elementos relativos al manejo de ficheros. *Nuevo* permite al usuario generar un archivo *.prg* en blanco. *Abrir* permite visualizar al usuario un archivo *.prg* previamente guardado. *Guardar como* permite guardar en un fichero el programa actual, la aplicación permitirá al usuario elegir el nombre y destino del fichero. *Guardar* permite guardar automáticamente el archivo *.prg* con el nombre actual del fichero.

**Edición:** Desde este submenú el usuario puede realizar las acciones relativas a la edición de texto. *Deshacer* permite volver atrás en los cambios realizados en el editor de código fuente. *Rehacer* elimina el último deshacer realizado. A continuación aparecen una serie de comandos que permiten realizar acciones sobre el texto que son comunes para todos los editores de texto como son: *Copiar*, *Cortar*, *Pegar*, *Eliminar* y *Seleccionar todo*.

**Ver:** Este submenú permite al usuario acceder a las opciones gráficas de la ventana de programación. *Fuente* permite elegir el tipo de letra, tamaño y color del editor de código. *Ver números de línea* muestra en el editor a la izquierda de cada línea, el número de línea correspondiente. *Ir a línea* abre un cuadro de texto que nos permite escribir el número de línea al que queremos ir de forma automática.



**Ejecutar:** Desde este submenú se puede acceder a todas las opciones relativas a los modos de ejecución del programa robótico: *Compilar aplicación, Ejecutar instrucción, Ejecutar aplicación, Detener ejecución, Pausa y Grabar en video*. Estos modos de ejecución del programa se explicarán con más detalle más adelante.










## 2. Barra de herramientas

La ventana de programación del constructor de escenarios cuenta con una barra de herramientas que permite realizar todas las acciones de la ventana.



Figura 56. Barra de herramientas de la ventana de programación.

Las acciones que realizan cada uno de los botones son

-  Crea un nuevo archivo de programa.
-  Carga un archivo de programa previamente guardado.
-  Guarda el archivo de programa actual.
-  Compila y ejecuta el programa robótico actual.
-  Pausa la ejecución del programa.
-  Ejecuta una instrucción del programa.
-  Detiene la ejecución del programa.
-  Compila el programa.
-  Ejecuta el programa robótico guardando la imagen del visor 3D en un archivo de video.

## 3. Zona de edición de código fuente

Esta es la zona reservada para que el usuario pueda escribir el código fuente de la tarea robótica. Posteriormente dicho código será compilado y ejecutado por un robot. La sintaxis del programa robótico es similar a Pascal y será estudiada con mayor detenimiento en siguientes capítulos.

El editor de texto resaltará las palabras reservadas del lenguaje de programación (if, then, else, while, etc), facilitando la lectura del código y su comprensión. Los comentarios en el código fuente, aparecerán en azul y en cursiva para mejorar su visualización.

Cuando la tarea robótica se encuentre en ejecución el editor de código remarcará con fondo azul la línea que se está ejecutando en ese momento.



#### 4. Barra de estado

La barra de estado muestra información relativa al editor de texto. Está dividida en cuatro casillas que muestran diferente información. La primera casilla nos indica la posición del cursor en el editor en formato fila:columna. La segunda de las casillas muestra si el archivo ha sido modificado desde su apertura. La tercera casilla nos muestra la opción actual de introducción de texto en el editor podrá ser insertar o sobrescribir. Por último la cuarta casilla se rellena con la ruta del archivo abierto.

#### 5. Zona de mensajes

En esta zona del editor se muestran los mensajes de notificación del compilador. Es la vía que utiliza el compilador para avisarnos de lo que ha sucedido. En este lugar aparecerán mensajes avisando de errores en la compilación o del estado de ejecución del programa. Por ejemplo nos avisará si se ha compilado, ejecutado o pausado un programa robótico.

#### *Tipos de ejecución del programa robótico*

Como se ha podido observar en la definición de los controles de la ventana de programación, al ejecutar una tarea robótica **RobotScene** cuenta con diferentes tipos de ejecución y controles que capacitan al usuario para el completo control de ejecución del programa robótico.

**Compilar programa:** Mediante esta acción se genera un código interno que será utilizado posteriormente por el motor de scripts para la ejecución de la tarea robótica. No es una compilación en el sentido estricto de la palabra puesto que no se genera ningún archivo ejecutable en código máquina. Existen tres maneras diferentes para realizar la compilación del código robótico:

→ *Menú principal > Ejecutar > Compilar aplicación.*

→ *Barra de herramientas > Botón compilar la aplicación*



→ *Combinación de teclas: Ctrl + F9.*

Al realizar esta acción, en la zona de mensajes de la ventana de programación aparecerá la información relativa a la compilación que avisará si el programa ha sido compilado con éxito o si el código contenía algún error sintáctico, remarcando la línea errónea de color rojo.

**Ejecutar una instrucción:** Permite la ejecución de una sola línea de programa siguiendo el flujo de programa actual. Tras ejecutar la instrucción el script se mantendrá pausado (modo pausa) a la espera de nuevas acciones de control del programa. Si en el momento de realizar la acción no está en ejecución el programa robótico se realizará una compilación previa. Se puede ejecutar una instrucción mediante tres caminos:

→ *Menú principal > Ejecutar > Ejecutar instrucción.*

→ *Barra de herramientas > Botón ejecutar instrucción*




→ *Tecla F8.*



También conocido como ejecución paso a paso, este modo de control de programa es muy útil para mantener el control del flujo de ejecución del programa y para labores de depuración. Este modo de control facilita la detección y resolución de errores en el código del programa. Para facilitar la visualización de la línea actual en ejecución, se remarca el fondo de dicha línea de color azul.

**Ejecutar la aplicación:** Este modo de ejecución es el más utilizado. En este modo, el script ejecutará instrucciones de programa sucesivamente hasta finalizar todas las instrucciones. Este modo realiza una compilación previa para detectar errores en el código de la tarea. El editor de código remarca la línea actual en ejecución con color azul. Para realizar la ejecución completa de la aplicación se pueden utilizar tres opciones diferentes:


→ *Menú principal > Ejecutar > Ejecutar aplicación.*

→ *Barra de herramientas > Botón ejecutar aplicación* 

→ *Tecla F9.*

**Detener la ejecución:** Detiene por completo la ejecución del script. La próxima vez que se ejecute la aplicación o una instrucción, lo hará desde la primera línea de código y previa compilación del código. Para acceder a esta opción son válidas cualquiera de las siguientes alternativas:

→ *Menú principal > Ejecutar > Detener ejecución.*

→ *Barra de herramientas > Botón detener ejecución* 

→ *Combinación de teclas Ctrl + F2.*

**Pausa:** Este modo de ejecución pausa la ejecución del script de forma que pueda recuperarse el flujo de ejecución del programa posteriormente. La principal diferencia con el modo *detener ejecución* es que permite recuperar el flujo de programa cuando la aplicación vuelva a reanudarse, ya sea mediante la ejecución de la aplicación o mediante una ejecución paso a paso. Para pausar el script podemos realizar las siguientes acciones:


→ *Menú principal > Ejecutar > Pausa.*

→ *Barra de herramientas > Botón pausa* 

**Grabar en video:** Mediante este modo se puede ejecutar la aplicación mientras el escenario robótico está siendo grabado en video. La única diferencia entre este modo de ejecución y el de ejecutar la aplicación es que éste genera, durante la ejecución de la tarea, un archivo *.avi* en el que se almacena el video de la escena. Para establecer este modo de ejecución se nos presentan dos opciones:



→ *Menú principal > Ejecutar > Grabar en video.*

→ *Barra de herramientas > Botón grabar en video* .

Como ocurría al ejecutar la aplicación, se realizará una compilación previa. Si esta compilación es exitosa se abrirá un explorador para guardar el video con el nombre deseado. A continuación se abrirá otra ventana que permitirá elegir el codec (vid, Mpeg4, DivX, etc) de compresión y otras opciones como la calidad de compresión o la velocidad de datos. El video se grabará a unos 25 FPS.

### **Modelado del entorno**

Cabe recordar en este apartado que, tanto para la construcción del escenario como para su explotación, se utiliza la misma interfaz. Sin embargo el modelado del entorno es diferente para cada modo de utilización del escenario.

En la parte de creación del escenario es posible definir relaciones jerárquicas entre los diferentes objetos que componen el escenario. Por su parte, los objetos cuentan con unas propiedades tales como posición, orientación, geometría y relaciones jerárquicas. En cambio, cuando se explota el escenario, los objetos sólo cuentan con posición y orientación dentro de sus propiedades.

### **Comunicación entre robots**

Uno de los requisitos principales de la versión multi-robot de **RobotScene** es que los robots tuvieran capacidad para comunicarse entre sí. Para lograr este requisito se ha implementado en **RobotScene** una ventana que pretende simular el cableado existente en instalaciones industriales donde intervienen varios robots de forma simultánea. Por lo tanto, la ventana de cableado posibilita la creación de un cableado entre robots de una manera sencilla, intuitiva y visual, que será guardado en un archivo con formato *.cbl*.

Para la realización del cableado se ha simulado una botonera de 10 entradas y 10 salidas para cada robot.

Para poder acceder a la ventana de cableado es imprescindible que el escenario disponga de al menos dos robots. Si han sido añadidos a nuestra escena al menos dos robots se activarán automáticamente las opciones para acceder a dicha ventana. Desde el constructor de escenarios se puede acceder a la ventana de cableado de dos formas diferentes:

→ *Menú principal > Ver > Ventana de cableado.*

→ *Barra de herramientas > Botón ventana de cableado* .

Accediendo de cualquiera de las maneras se mostrará en pantalla la ventana de cableado que se presenta del siguiente modo:



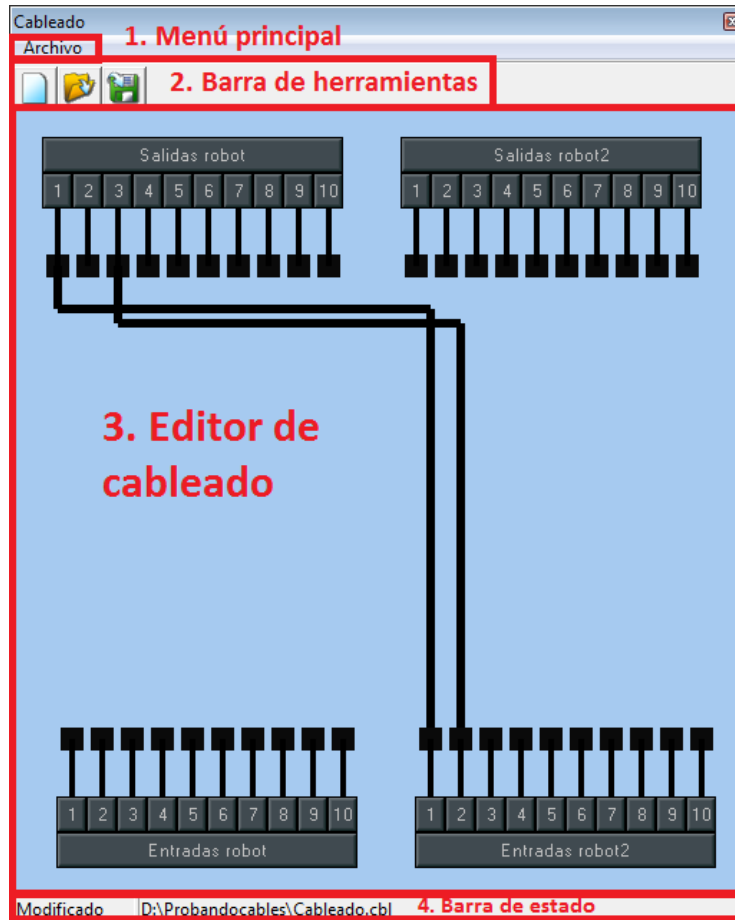


Figura 57. Constructor de escenarios. Ventana de cableado.

Tal como podemos observar en la imagen la ventana de cableado presenta las siguientes partes:

### 1. Menú principal.

Desde el menú principal podemos acceder a a las opciones relativas al guardado y carga del escenario.

**Archivo:** Esta es la única opción del menú, en ella el usuario tiene a su disposición los elementos relativos al manejo de ficheros.

*Nuevo* permite al usuario generar un archivo *.cbl* en blanco. Es posible acceder a esta opción mediante la combinación de teclas Ctrl + N.

*Abrir* permite visualizar al usuario un archivo *.cbl* previamente guardado. Es posible abrir un archivo mediante la combinación de teclas Ctrl + A.

*Guardar como* permite guardar en un fichero el cableado actual, la aplicación permitirá al usuario elegir el nombre y destino del fichero.

*Guardar* permite guardar automáticamente el archivo *.cbl* con el nombre actual del fichero. Es posible guardar el archivo mediante la combinación de teclas Ctrl + G.

*Cerrar ventana*, cierra la ventana de cableado.



## 2. Barra de herramientas

La barra de herramientas permite realizar las acciones relativas al manejo de ficheros de cableado.



Crea un nuevo archivo de cableado.

Carga un archivo de cableado previamente guardado.

Guarda el archivo de cableado actual.

## 3. Editor de cableado

Es la parte más importante de la ventana de cableado. Permite al usuario editar un archivo de cableado que simula un cableado real. Esta ventana ajusta su tamaño al número de robots en el escenario. Como se puede observar en la imagen de la ventana, en la zona superior se encuentran las botoneras que reúnen las salidas binarias de los robots industriales. Por otro lado, en la parte inferior se encuentran las botoneras correspondientes a las entradas digitales de los robots.

Este editor permite cablear las salidas de un robot con las entradas de los demás robots del escenario. Para realizar dicho cableado se pone a disposición del usuario esta interfaz en la que se pueden simular cableados de forma gráfica e interactiva. A continuación se describe el modo de uso del editor.

### Creación de un cable

La creación del cable debe hacerse de forma interactiva entre el ratón y el editor. El usuario puede mover el ratón por el editor. Si el usuario coloca el cursor encima de cualquiera de los pads de salida, observará que dicho pad cambia a color verde, lo que representa que es posible comenzar la creación del cable.

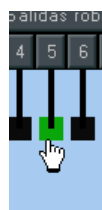


Figura 58. Cursor sobre un pad de salida en la ventana de cableado.

Para crear el cable solamente es necesario pulsar sobre un pad de salida y arrastrar el ratón manteniendo pulsado el botón izquierdo del ratón. De esta manera se ha creado un cable que puede ser arrastrado por la pantalla.

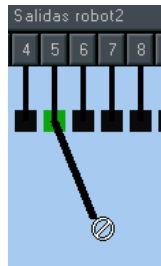


Figura 59. Ventana de cableado. Creación del cable.

Como se observa en la imagen anterior, el cursor cambia su imagen a una señal de prohibido que indica que, para terminar con la creación del cable, no es posible soltar el ratón en ese punto. También se advierte que los pads de entrada en los que puede terminar el cable se colorean en verde.

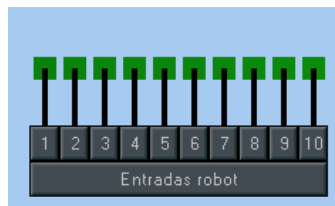


Figura 60. Ventana de cableado. Los pads de entrada válidos para crear un cable se colorean en verde.

Finalmente, para terminar con la creación del cable solo es necesario llevar el cursor del ratón a un pad de entrada válido y, a continuación, soltar el botón izquierdo del ratón. Para asegurarnos de que el pad de entrada es válido, debemos observar que el pad es de color verde y que cambia la imagen del cursor.

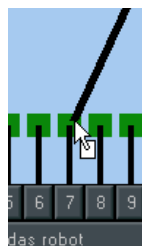


Figura 61. La imagen del cursor nos informa de que es un pad de entrada válido.

De esta manera se habrá creado un cable entre el pad de salida y el pad de entrada deseados. El programa implementará internamente dicho cable de modo que será posible la comunicación entre robots utilizando instrucciones propias del lenguaje robótico como *wait*, o *signal*.

### Eliminación de un cable


El editor de cableado, permite a su vez que se puedan eliminar cables. Eliminar un cable en el editor es tan sencillo como situar el cursor del ratón encima del cable que se desea eliminar y pulsar el botón derecho del ratón.

### Monitorización de entradas y salidas



**RobotScene** cuenta con una ventana que permite monitorizar de manera simple las entradas y salidas binarias de los robots del escenario. La ventana de monitorización se modifica automáticamente de acuerdo con el número de robots del escenario. Para poder visualizar la ventana de monitorización es imprescindible que el escenario cuente con al menos un robot. Si el escenario cuenta con uno o más robots se activarán las opciones que permiten al usuario abrir la ventana. Dichas opciones son dos y se puede acceder a ellas mediante

→ *Menú principal > Ver > Ventana E/S.*

→ *Barra de herramientas > Botón ventana de E/S* .

Con cualquiera de las dos opciones se mostrará en pantalla la ventana de entradas y salidas que se corresponde con la siguiente imagen:

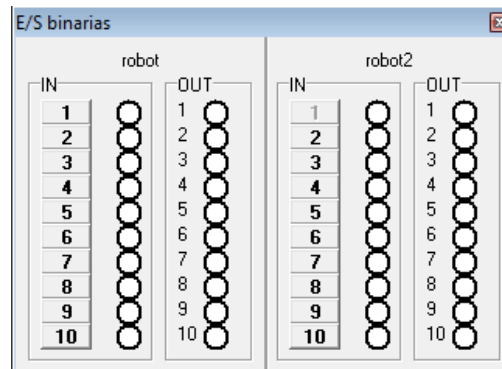


Figura 62. Constructor de escenarios. Ventana E/S.

La función de la ventana es la monitorización de las entradas y salidas; sin embargo, simulando una botonera de entradas en un robot real, las entradas son interruptores de tres estados.

Los dos primeros estados se corresponden con el 1 y el 0 binarios. El tercero de los estados se corresponde con entrada inhabilitada, que en la actual aplicación equivale a que dicha entrada se encuentra cableada. La inhabilitación de las entradas se realiza de forma automática cuando se cablea una entrada. En la botonera de entradas se ha implementado un botón por cada entrada que permite poner a 1 o 0 el valor de la entrada. En las siguientes imágenes se pueden observar el estado de las entradas:

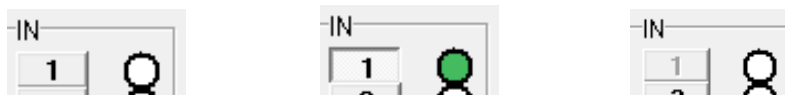
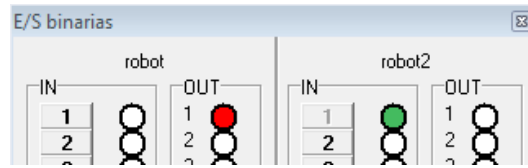


Figura 63. Ventana E/S. Entrada con valor 0, con valor 1 y deshabilitada.

Adicionalmente, cabe destacar que es posible abrir la ventana de monitorización durante la ejecución de un programa robótico por lo que constituye una buena herramienta para conocer el estado de entradas y salidas. Cuando una salida cambia su estado al 1 binario se observará de color rojo,



como puede apreciarse en la siguiente imagen en la que se activa, la salida 1 del robot 1 y la entrada 1 del robot 2 que previamente han sido cableadas:



**Figura 68.** Una salida y una entrada cableadas se señalan de diferente color para mejorar su visualización.



## 6 DESCRIPCIÓN DEL LENGUAJE DE PROGRAMACIÓN DE ROBOTS

Tal como se anunció en el apartado “Análisis de soluciones” el desarrollo de la versión multi-robot de **RobotScene** supone la elaboración de procedimientos que permitan el desarrollo de tres tareas que la versión mono-robot no permitía llevar a cabo: creación y gestión de ventanas de programación para cada robot, ejecución simultánea de instrucciones de programa y establecimiento de comunicación efectiva entre los robots.

En lo que sigue, se explica qué instrucciones conforman el lenguaje robótico. Dichas instrucciones robóticas son las mismas para una versión mono-robot que para una versión multi-robot, pues estas instrucciones se ejecutan sobre un único robot, y son generalmente instrucciones de movimiento. Además, la versión mono-robot de **RobotScene** ya contaba con instrucciones para manejar las entradas y salidas binarias. Dichas instrucciones han sido modificadas en el código fuente del programa para implementar una solución multi-robótica; sin embargo, para el usuario, las instrucciones tienen el mismo nombre y funcionan de la misma manera. La única modificación realizada en estas instrucciones respecto a la versión mono-robot ha sido el cambio de nombre de la antigua instrucción *stop* por *pause*.

Como consecuencia de lo comentado en el párrafo anterior, el lenguaje robótico utilizado se ajusta a lo que se recoge a continuación que es, en realidad, copia del lenguaje empleado en la versión mono-robot.

### 1. Tipos de datos estándares.

El lenguaje de programación habilitado para la programación de las aplicaciones robóticas es un lenguaje basado en Pascal y tiene su misma sintaxis. Por esta razón, se podrán utilizar gran parte de los datos definidos para el lenguaje Pascal estándar.

#### *Tipos enteros*

Tipo	Rango	Formato
Integer	-2147483648 a 2147483647	Con signo 32 bits.
Cardinal	0 a 4294967295	Sin signo 32 bits.
Shortint	-128 a 127	Con signo 8 bits
Smallint	-32768 a 32767	Con signo 16 bits.
Longint	-2147483648 a 2147483647	Con signo 32 bits.
Byte	0 a 255	Sin signo 8 bits
Word	0 a 65535	Sin signo 16 bits
Longword	0 a 4294967295	Sin signo 32 bits.

Existen dos tipos base de 32 bits que son *integer* y *cardinal*. Las operaciones aritméticas entre tipos enteros, por regla general, devolverán un valor de tipo *integer*. A todos los efectos resultan equivalentes los tipos *Integer* y *Longint*, así



como *Cardinal* y *Longword*. No hay problemas de asignación de diferentes tipos mientras el valor esté dentro del rango adecuado.

**Tipos reales:**

Tipo	Rango	Cifras	Tamaño
Single	$1.5 \times 10^{-45}$ a $3.4 \times 10^{38}$	7-8	4 bytes
Double	$5.0 \times 10^{-324}$ a $1.7 \times 10^{308}$	15-16	8 bytes
Extended	$3.6 \times 10^{-4951}$ a $1.1 \times 10^{4932}$	19-20	10 bytes
Currency	-922337203685477.5808 a 922337203685477.5807	19-20	8 bytes

Todos estos datos tienen una notación de coma flotante, excepto el tipo de dato *currency*, que se representa con coma fija. La representación de éste último se realiza internamente como un entero de 64 bits con los últimos 4 dígitos, los menos significativos, representando las cifras decimales.

Se recomienda insistentemente el uso del tipo de dato *single*, de 4 bytes (32 bits), puesto que muchas rutinas algebraicas están optimizadas para este tipo de dato, con lo que la ejecución resultará mucho más fluida.

**Tipos carácter (char):**

El tipo *char* puede contener un solo carácter. Cada uno de estos caracteres puede ser expresado gracias al código ASCII. Un carácter *char* se representa entre comillas simples, por ejemplo: 'A' 'b' '\*' '\ ' '5'.

Los tipos de datos carácter son ordinales, lo cual significa que cada carácter representa una posición en una serie ordenada (de 0 a 255). El orden de la letra A es 65, el de la letra B es 66, etc. Para saber el orden basta con mirar una tabla ASCII.

La función predefinida *chr* permite referenciar todos los caracteres, así, *chr(65)* equivale a la letra A, *chr(66)* equivale a la letra B, y así sucesivamente.

**Tipo cadena (string)**

Un tipo *string* (cadena) es una secuencia de caracteres correspondientes al código ASCII, de longitud indefinida, escrito en una línea sobre el programa y representado entre comillas simples, por ejemplo: 'cadena', 'otra cadena'. Se permite la cadena de longitud nula y se representa como ''.

**Tipo booleano**

Una variable de tipo *booleano* puede tomar dos valores *true* o *false* (verdadero o falso), y surge como resultado de aplicar los operadores de comparación: =, >, <, >=, <= y <>.

Las operaciones que pueden realizarse con este tipo de datos son **not**, **and**, **or**, **xor**.



### **Tipo array (vector)**

Este tipo de dato complejo almacena un conjunto de elementos del mismo tipo. Sus elementos son accesibles de forma independiente mediante un índice. El tamaño está fijado y deberá especificarse en el momento de declarar una variable de tipo array.

A continuación se muestra un ejemplo de la declaración de un tipo de variable vector:

```
Type nombre_array=array [1..8] of single;
```

Es un vector de 8 elementos, con índices de 1 a 8. Para acceder a uno de sus elementos nos referiremos al índice, así, el segundo elemento será `nombre_array[2]` y será un tipo de dato `single`.

En el lenguaje robótico esta permitido la declaración y uso de *arrays* de varias dimensiones o matrices, accesibles desde dos índices. Ejemplo `array[1..numero_filas,1..numero_columnas] of single`.

### **Tipo registro (record)**

Es un tipo de datos estructurado que almacena un conjunto de elementos que pueden ser del mismo tipo o de tipos diferentes.

Los componentes de un registro se denominan campos. Cada campo tiene un nombre, llamado identificador de campo, que es un identificador elegido por el programador cuando se declara el tipo de dato `record`. Un ejemplo de `record` se puede observar a continuación:

```
type  
  
Mi_registro = record  
    Campo1:single;  
    Campo2:integer;  
end;
```

Para acceder al primero de los campos se escribe: `Mi_registro.campo1`

## **2. Tipos de datos propios del lenguaje robótico:**

### **Coordenadas de un punto:**

Para referirnos a un vector tridimensional disponemos del dato *TCoordenadas*, cuya declaración es la siguiente:





```
type  
  
TCoordenadas = record  
    x,y,z:single;  
end;
```

Se puede observar en la declaración que el tipo *TCoordenadas* es de tipo *record*, y por esta razón se acceden a sus campos de la misma manera que en un *record*. Para referirnos a la componente x de un punto P (P es de tipo *TCoordenadas*) deberemos escribir *P.x*

### **Transformación homogénea:**

Se ha definido el tipo de dato *TTransformation* que representa una matriz de transformación homogénea, que permitirá almacenar la posición y la orientación de un sistema de referencia:

```
type  
  
TTransformation=record  
    pos,n,o,a:TCoordenadas;  
end;
```

Para una transformación *T*, de tipo *TTransformation*, nos referiremos a la componente x de la posición mediante *T.pos.x*, o para referirnos a la componente *n<sub>y</sub>* de la submatriz de rotación *n, o, a* mediante *n.y*.

### **3. Funciones algebraicas y trigonométricas:**

El lenguaje robótico permite la utilización de una serie funciones algebraicas y trigonométricas para facilitar la programación de la tarea robótica:

```
function Sin(X: Single): Single;           // Seno  
function Cos(X: Single): Single;         // Coseno  
function Sqrt(X: Single): Single;        // Raíz  
function Atan2(x,y:single):single;       // Arcotangente y cuadrante  
function ACos(x:single):single;          // Arco Coseno  
function degrees(rads:single):single     // Convierte a grados  
function rads(degrees:single):single;    // Convierte a radianes
```

### **4. Instrucciones del lenguaje.**

Existen tres tipos de primitivas dentro del lenguaje robótico. Las referentes a la importación de datos de la interfaz, las primitivas algebraicas para el trabajo matricial con transformaciones homogéneas y las primitivas de movimiento del robot. Las primitivas de movimiento del robot incluyen tanto las primitivas de



uso de la herramienta como las primitivas para la gestión de las entradas y salidas binarias del robot.

### ***Primitivas para la importación de datos de la jerarquía de la escena.***

Mediante una serie de funciones habilitadas, se puede incorporar a la tarea robótica transformaciones que habrán sido previamente definidas en la construcción del escenario. Se dispone de las dos funciones siguientes:

```
function Import_Relative_Transform(s:string):TTransformation;  
function Import_Absolute_Transform(s:string):TTransformation;
```

Ambas funciones devolverán un valor del tipo TTransformación y toman como parámetro el nombre que tiene en el árbol el objeto que se quiere importar. La diferencia radica en que mediante *Import\_Absolute\_Transform* se importa la transformación absoluta del objeto, mientras que mediante la función *Import\_Relative\_Transform* se importa la transformación del objeto pero relativa a su antecesor en el árbol. Si se utiliza esta última técnica, para especificar la transformación total del objeto se deben realizar una serie de composiciones de transformaciones, recorriendo el árbol, mientras que si se importa de forma absoluta, toda esta serie de operaciones se realizará de forma interna y automática.

### ***Primitivas para la modificación de la jerarquía de la escena.***

Es posible modificar desde programa la jerarquía de la escena, de forma similar a las modificaciones que pueden realizarse de forma interactiva. Esta reasignación permite, por ejemplo, manipular un conjunto de objetos por medio de la manipulación del objeto padre (un palet y las piezas que haya sobre él).

```
procedure Affix (child,new_father:string);
```

El procedimiento toma como parámetro los nombres con los que figuran en el árbol los objetos involucrados. Esta operación no modifica la localización absoluta del objeto modificado, lo que conlleva necesariamente el correspondiente cambio en su localización relativa.



**Primitivas para el trabajo con transformaciones homogéneas.**

Estas primitivas permiten realizar operaciones básicas con el tipo de dato *TTransformation*. Todas las rotaciones irán expresadas en grados y las distancias en milímetros.

<b>function</b> RotYPR(yaw,pitch,roll:single):TTransformation;	Define una transformación a partir de los ángulos de roll, pitch y yaw.
<b>function</b> RotEuler(alfa,beta,gamma:single):TTransformation;	Define una transformación a partir de los ángulos de Euler.
<b>function</b> RotX(angle:single):TTransformation; <b>function</b> RotY(angle:single):TTransformation; <b>function</b> RotZ(angle:single):TTransformation;	Rotaciones en torno a los ejes principales.
<b>function</b> RotVector(axis:Tcoordenadas;angle:single):TTransformation	Rotación en torno a un eje arbitrario. La especificación del eje de giro se hará mediante el parámetro de tipo Tcoordenadas.
<b>function</b> NullTr:TTransformation;	Definición de la transformación nula, o lo que es lo mismo, la matriz identidad.
<b>function</b> Comp(T1,T2:TTransformation):TTransformation;	Composición (multiplicación) de dos transformaciones.
<b>function</b> Invert(T:TTransformation):TTransformation	Devuelve la inversa de una matriz.
<b>function</b> TransfYPR(x,y,z,yaw,pitch,roll:single):TTransformation;	Define una transformación mediante x, y, z y los ángulos roll, pitch y yaw.
<b>function</b> TransfEuler(x,y,z,alfa,beta,gamma:single):TTransformation;	Define una transformación mediante x, y, z y los ángulos alfa,beta,gamma.
<b>function</b> Translation(x,y,z:single):TTransformation;	Define una traslación.
<b>function</b> Shift(T:TTransformation; x,y,z:single):TTransformation;	Traslada una transformación respecto de la referencia absoluta.
<b>function</b> ShiftRel(T:TTransformation; x,y,z:single):TTransformation;	Traslada una transformación respecto de la propia referencia.



**Primitivas de uso del robot.**

<b>procedure</b> procedure speed(percent:single);	Define la velocidad de movimiento del robot en tanto por ciento de la máxima permisible.
<b>procedure</b> acceleration(percent:single);	Define la aceleración de movimiento del robot en tanto por ciento de la máxima permisible.
<b>function</b> where_tool:TTransformation;	Nos devuelve la localización de la herramienta del robot, si la tuviera, en caso contrario nos devolverá la localización del último de los sólidos del robot.
<b>function</b> joints:TArray;	Devuelve el valor de las articulaciones del robot.
<b>procedure</b> set_configuration(conf:string);	Procedimiento para cambiar la configuración del robot.
<b>procedure</b> pause(ms:integer);	Detiene la ejecución de la aplicación durante un tiempo expresado en milisegundos.
<b>procedure</b> MovJoint(destination:Tarray);	Movimiento coordinado articular cuyo destino es un vector de coordenadas articulares que es tomado como parámetro.
<b>procedure</b> MovCoor(destination:TTransformation);	Movimiento coordinado articular cuyo destino es una transformación.
<b>procedure</b> MovLin(destination:TTransformation);	Movimiento rectilíneo cuyo destino es una transformación.
<b>procedure</b> MovRelCoor(destination:TTransformation);	Movimiento coordinado relativo a la herramienta.
<b>procedure</b> MovRelLin(destination:TTransformation);	Movimiento rectilíneo relativo a la herramienta.
<b>procedure</b> Drive(number:integer;amount:single);	Mueve la articulación dada por el primer parámetro un ángulo/distancia dada por el segundo parámetro.
<b>procedure</b> Home;	Vuelve a la posición de reposo del robot.

Las siguientes instrucciones son relativas al uso de la herramienta asociada al robot. Si el robot no tuviera ninguna herramienta asociada y se



utilizara alguna de las siguientes expresiones, dichas expresiones no tendrían ningún efecto.

<b>procedure</b> Tool(T:Transformation);	Cambia la referencia asociada a la herramienta, y por tanto, la referencia asociada al extremo del robot.
<b>procedure</b> Open;	Abre la herramienta hasta la apertura máxima.
<b>procedure</b> close(amount:single)	Cierra la herramienta hasta la apertura especificada por el parámetro (en mm).

Las siguientes instrucciones son relativas al uso de las entradas y salidas binarias asociadas al robot.

<b>procedure</b> wait(i:integer);	Detiene la ejecución del programa hasta que se active la entrada referenciada.
<b>procedure</b> signal(i:integer);	Activa o desactiva la salida referenciada dependiendo de su signo. Si el signo es positivo se activará la salida, mientras que si es negativo se desactivará.
<b>function</b> sig(i:integer):Boolean;	Lee el estado de la entrada tomada como parámetro.