



ESCUELA UNIVERSITARIA DE
INGENIERÍA TÉCNICA INDUSTRIAL

**SOFTWARE DE CREACIÓN Y
EXPLOTACIÓN DE ESCENARIOS
ROBÓTICOS**

MANUAL DE INSTRUCCIONES



Javier Fermín Hernández
Ingeniería Técnica Industrial en Electrónica Industrial
E.U.I.T.I. de la Universidad de Zaragoza

Indice.

1. Introducción.....	4
1.1 Introducción a RobotScene.....	4
1.2 Requisitos mínimos.....	4
1.3 Instalación	4
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 triedro 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
4.2 Descripción de la interfaz de usuario.....	36
4.3 Metodología de creación de un robot.....	38
4.4 Implementación de la cinemática inversa.	41
5. El constructor de escenarios.	54
5.1 Descripción de la interfaz de usuario.....	54
5.2 Creación del escenario.....	57
5.3 Explotación del escenario.....	62
6. Descripción del lenguaje de programación de robots.	69
6.1 Tipos de datos estándares.	69
Tipos reales:	69
6.2 Tipos de datos propios del lenguaje robótico:	71
6.3 Instrucciones del lenguaje.....	72
6.3.1 Primitivas para la importación de datos.	72
6.3.2 Primitivas para el trabajo con transformaciones homogéneas.	72
6.3.3 Primitivas de uso del robot.	74

1. Introducción.

1.1 Introducción a RobotScene

Mediante la RobotScene usted será capaz de crear de una forma visual, rápida y muy intuitiva escenarios tridimensionales virtuales con robots industriales para posteriormente controlar a dichos robots y simular aplicaciones reales, en las que los robots interactuarán con el entorno.

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

Si embargo sí que serán necesarios conocimientos generales de programación para poder aprovechar todas las características que se le ofrecen, de cara a poder implementar usted mismo 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.

Concluimos pues en que a diferencia de otras aplicaciones similares, en RobotScene usted 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.

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 Pentium a 500 MHz o superior.
- Sistema operativo Windows 2000, Windows 98, Windows Me o Windows XP.
- 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.

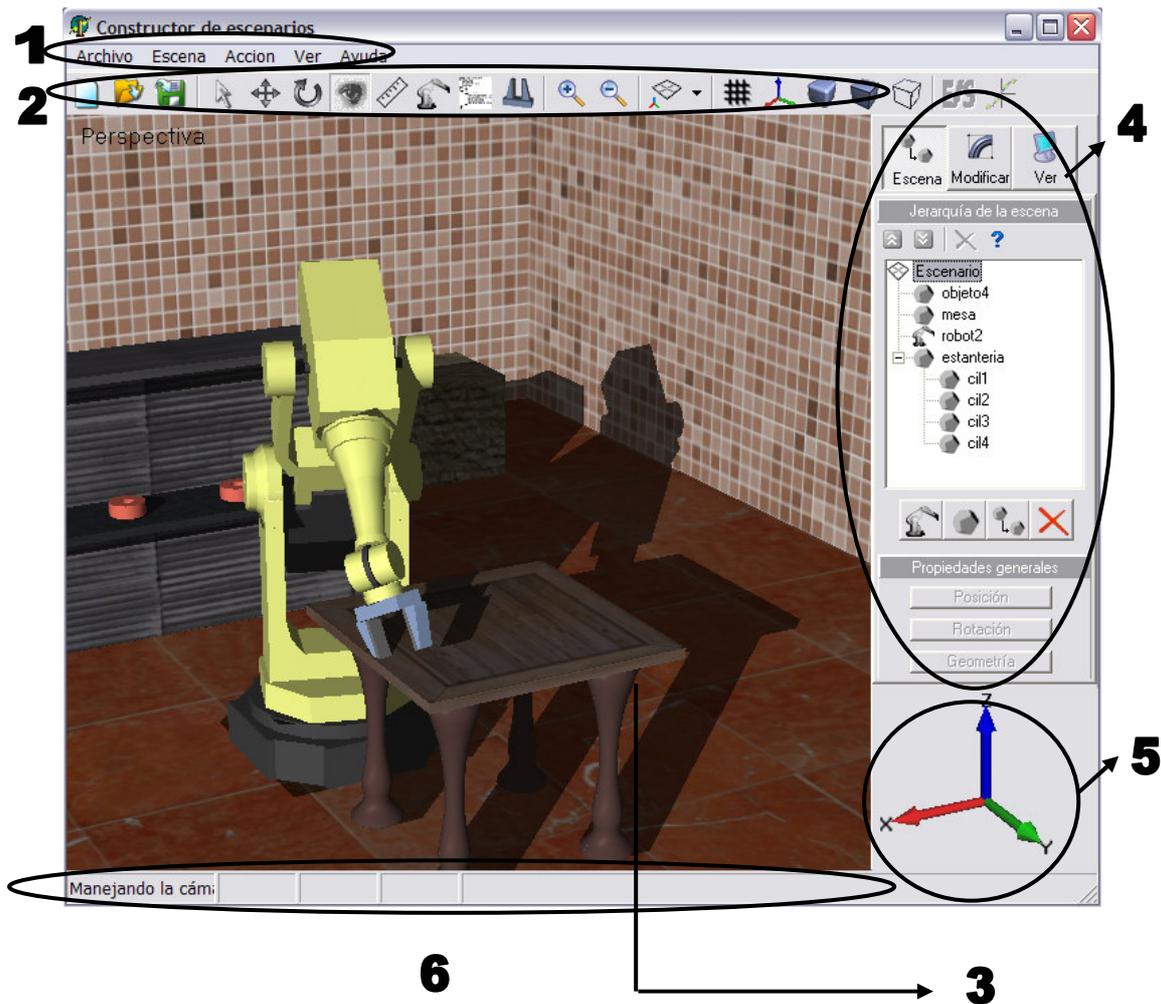
1.4 Desinstalación

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, que procederemos a explicar a continuación, 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:



Interfaz de usuario del constructor de objetos

- 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, que pasamos a explicar sus funcionalidades por encima:

En el primer módulo, el constructor de objetos, se realiza una construcción geométrica de un sólido, independientemente del uso que se le vaya a dar a dicho sólido, que podrá formar parte de un robot o bien ser un elemento del escenario, pero no es en este módulo donde debemos decidir su funcionalidad. Aquí llevamos a cabo el modelado del sólido de una forma visual, una de las partes más costosa de todo el proyecto y de la cual dependen el resto de módulos que componen el proyecto. 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 a la construcción de un robot virtual. Para dicha construcción 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 (θ , α , a , 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 máxima.

Otra de las características de este módulo es la resolución programada del modelo inverso del robot mediante scripts (código fuente que interpretaremos), por lo que se deberán obtener previamente las ecuaciones correspondientes para posteriormente proceder a su implementación programada, así que serán necesarios llegados a este punto unos ligeros conocimientos de programación por parte del usuario.

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

Por último, a través del constructor de escenarios procederemos a crear nuestro escenario virtual colocando nuestro (o nuestros) robot en un entorno virtual y procederemos a su explotación, bien mediante un control que emula una botonera de guiado o bien programando directamente nuestro robot mediante un lenguaje robótico creado para el caso.

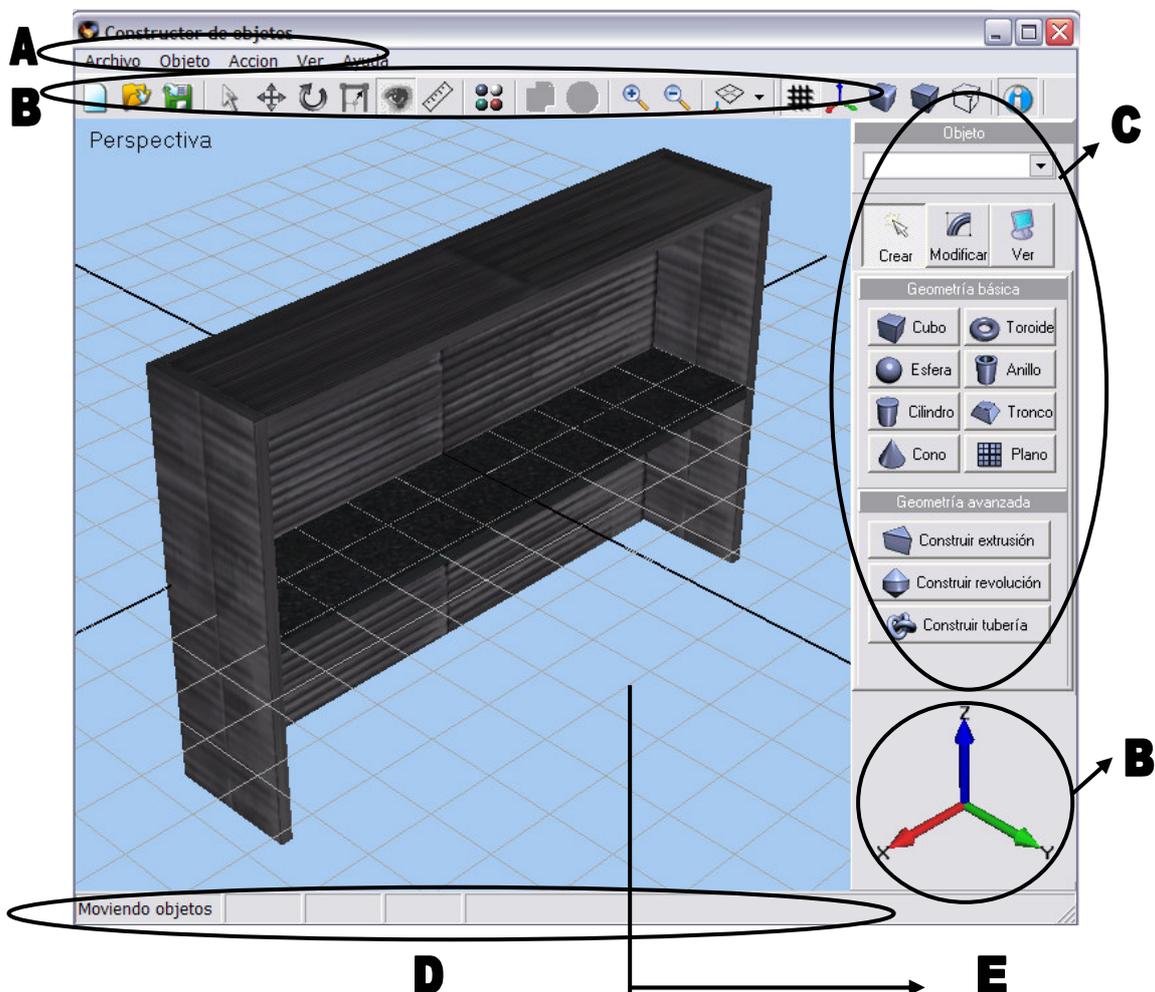
3. El Constructor de objetos.

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 propiciando 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.



Interfaz 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 elementos o bien controlar la representación visual.

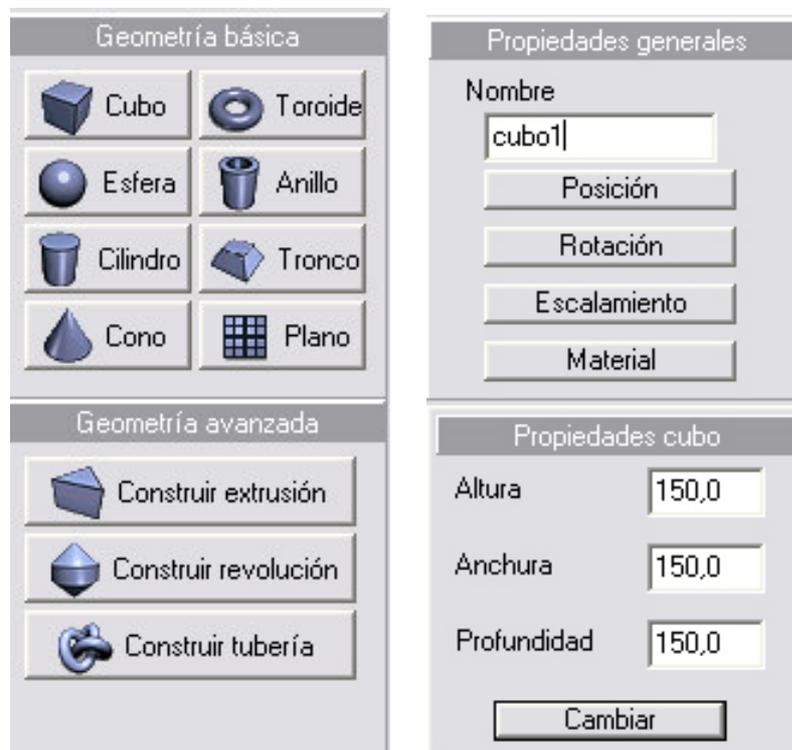
Tenemos tres zonas diferenciadas como podemos apreciar en la imagen:

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.



Diferentes opciones del panel lateral

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 tres dimensiones.

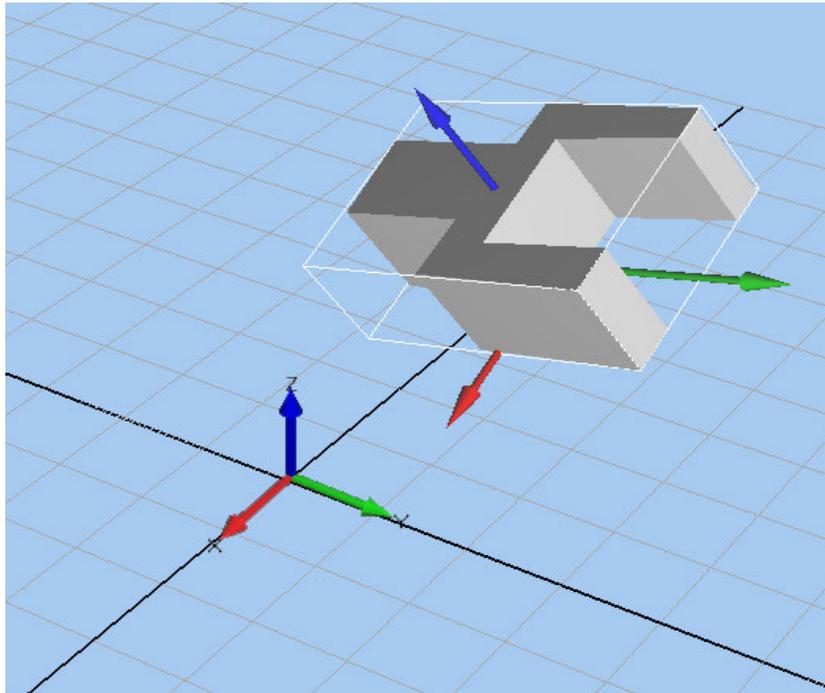
En esta parte de la pantalla se procede a la representación tridimensional de la escena que está siendo creada. Aparte de la escena propiamente dicha podemos observar la cuadrícula o rejilla y los ejes principales, representados por dos líneas más gruesas que se cruzan en el origen. La cuadrícula está 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.



Referencia absoluta y referencia relativa del 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 triedro universal.

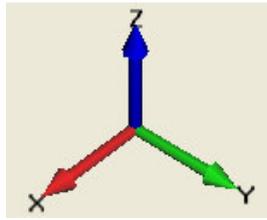
En este módulo tenemos seis modos de operación distintos, siendo un modo de operación el comportamiento que tiene la escena tridimensional frente a entradas provenientes del ratón. Se puede acceder a estos modos bien mediante el menú principal (Menú > Acción) o bien mediante la barra de herramientas.

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 [] 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.



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.

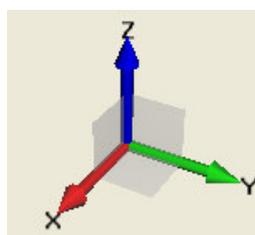
Al seleccionar un objeto, se habilitará el panel de propiedades generales, por lo que podremos acceder a ellas, y también aparecerá el panel de propiedades específicas correspondientes al tipo de objeto. Mientras exista un elemento seleccionado se podrá acceder a sus características mediante ambos paneles 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 > boton[] o menú principal > objeto > copiar objeto) lo que nos creará una copia del objeto seleccionado y eliminar objeto (barra de herramientas > boton [] 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 [] 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:

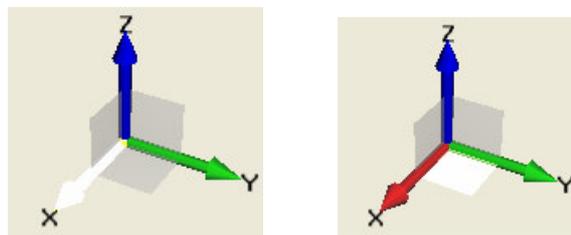


Forma que adopta el triedro en el modo de traslación de objetos

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.



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

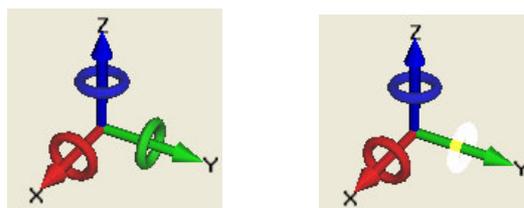
Cuando nos encontremos en un visor ortogonal hay direcciones de movimiento que pierden sentido, así que son deshabilitadas, así, por ejemplo, si nos situamos en el visor cenital (Visor XY), el movimiento a través del eje Z queda deshabilitado, puesto que en esta vista el eje Z es perpendicular al plano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



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  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.

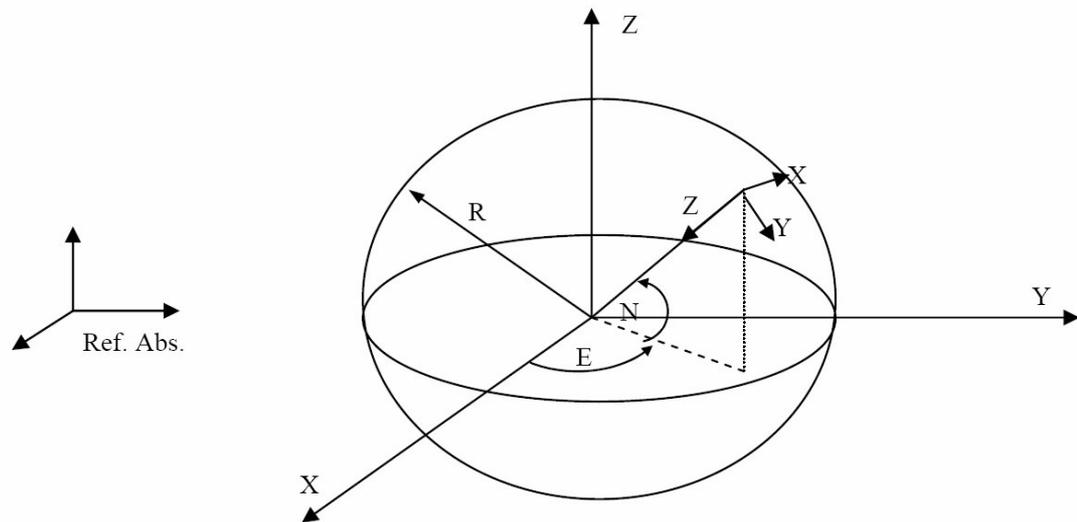
Cuando nos encontramos en este modo de operación seleccionaremos el objeto que queramos poniendo el puntero del ratón encima y pulsando el botón izquierdo del ratón, de la misma forma que hemos venido haciendo hasta ahora, con la diferencia de que mientras tengamos el botón del ratón apretado podremos 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 de 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 [] 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.



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 [] 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 [] 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.

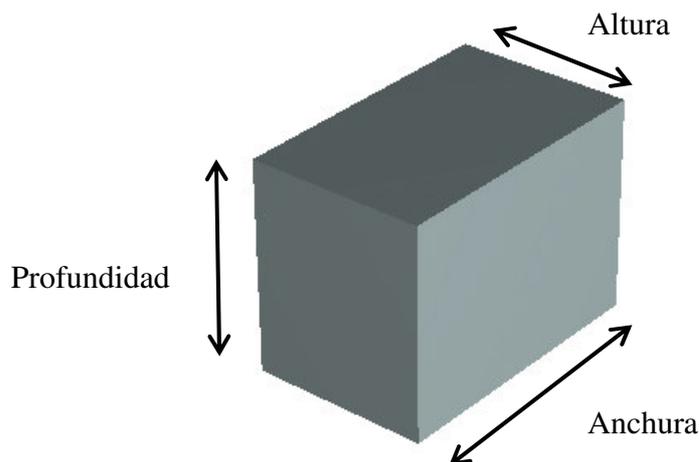
Como ya se ha explicado suficientemente, mediante este módulo construiremos un sólido a partir de formas geométricas predefinidas. Todos los objetos que componen el sólido tienen una serie de atributos comunes, como la posición, orientación, escala o material. Se puede acceder a estas propiedades a partir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

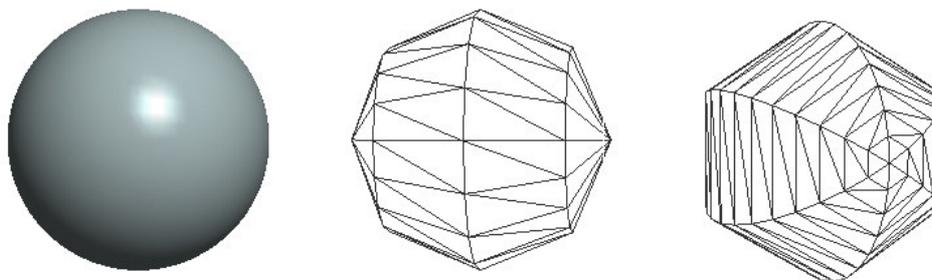
Es el objeto más sencillo y tiene tres características básicas: altura, anchura y profundidad, y podremos acceder a ellas desde el panel lateral (*Panel lateral > Panel Modificar > Propiedades caja*). Podemos apreciar el significado de dichas propiedades en la siguiente figura:



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:



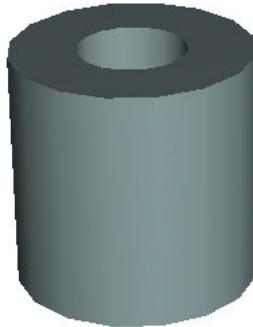
A la izquierda una esfera suavizada, en el centro una esfera con 4 divisiones y en la derecha una esfera con 6 rodajas

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

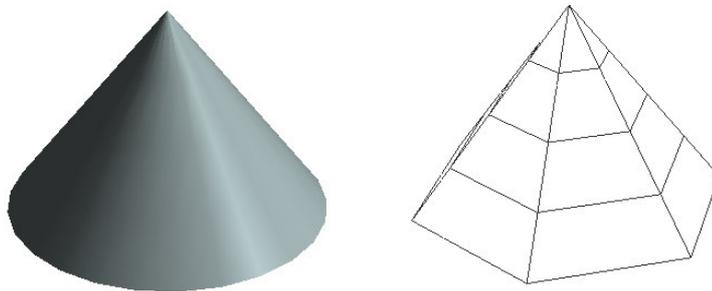
superior e inferior pueden ser distintos, por lo que se pueden llegar a construir troncos de cono. Por otra parte, la propiedad lados se refiere al número de lados de la base circular. Podemos apreciar estos parámetros en la siguiente figura:



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:

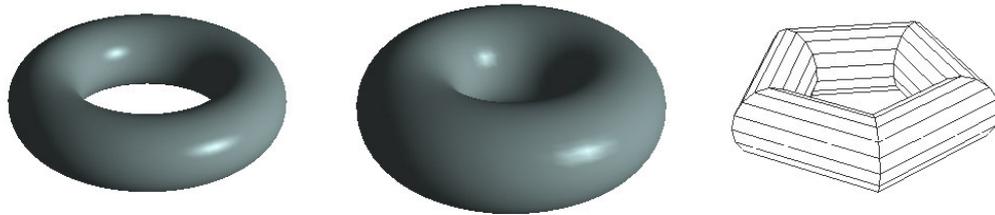


A la izquierda un cono suavizado y a la derecha un cono de 6 lados.

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 lados establece el número de lados de cada sección del toroide. Podemos apreciar estos parámetros en la siguiente figura:



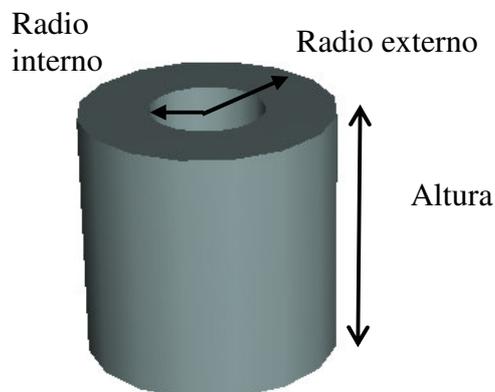
Distintos tipos de toroides.

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 [I]) 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:

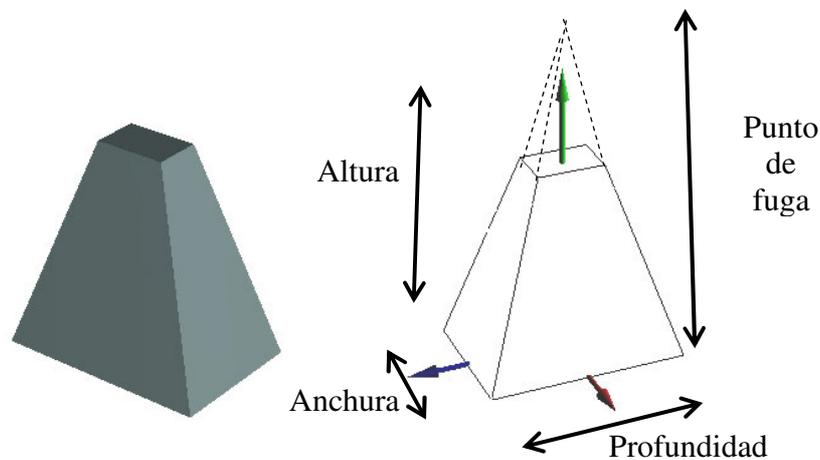


Parámetros de un anillo

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:



Parámetros del tronco de pirámide, visualizando también los ejes del objeto.

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).

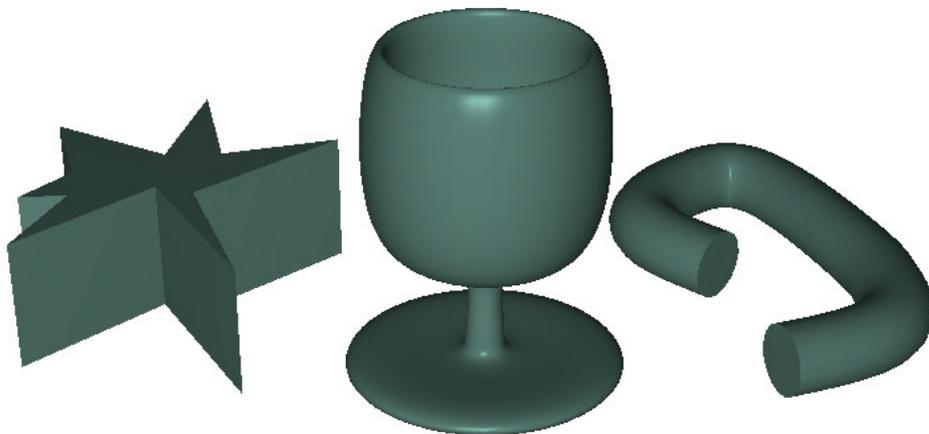
El plano será creado en el plano XY y tiene las siguientes propiedades específicas: Anchura, profundidad, repeticiones en X y repeticiones en Y. La anchura y la profundidad se refieren al las dimensiones del plano medidas en los ejes X e Y respectivamente, mientras que los parámetros de las repeticiones 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.



Planos con textura y con diferentes repeticiones, el de la derecha tiene 3 en Y y 3 en X.

3.5 Formas más complejas

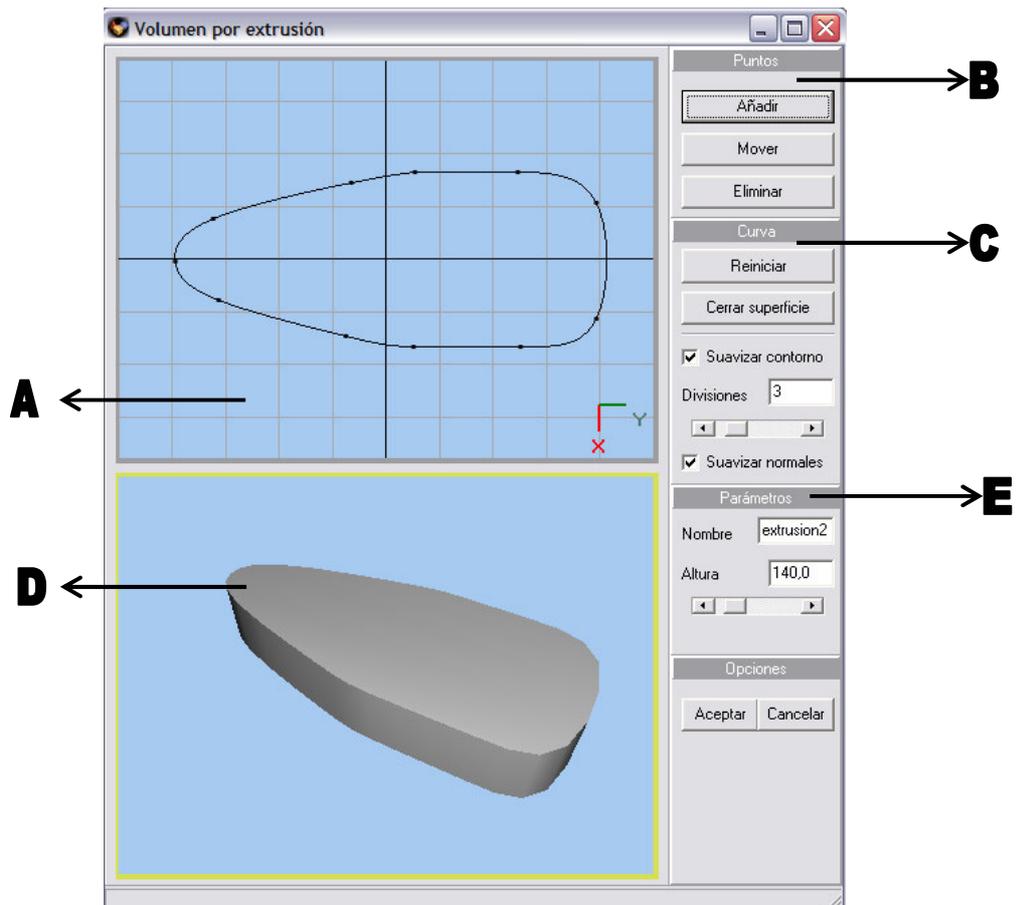
Aparte de las formas básicas explicadas, para la construcción de objetos de formas más complejas se ha habilitado la creación de volúmenes mediante ‘desplazamiento’ de superficies. De esta manera se deberá definir una superficie plana, bien mediante segmentos o mediante curvas ‘spline’, y una vez definida 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:



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 [I]) 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.



Interfaz para la construcción de sólidos por extrusión

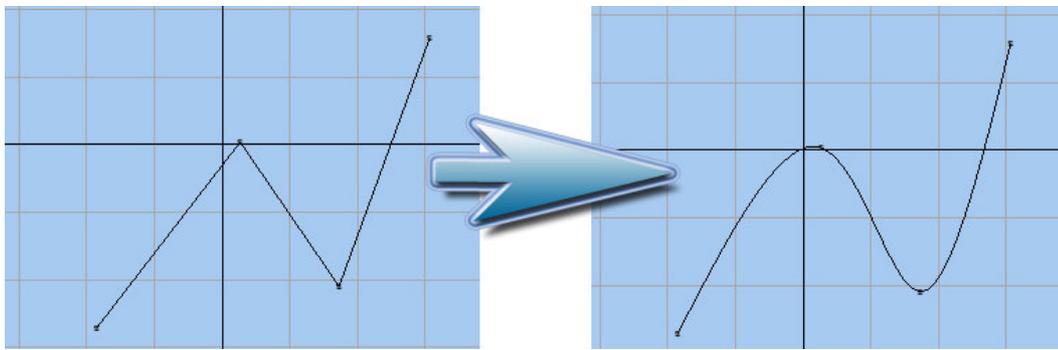
Podemos distinguir las siguientes zonas en la interfaz:

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 me

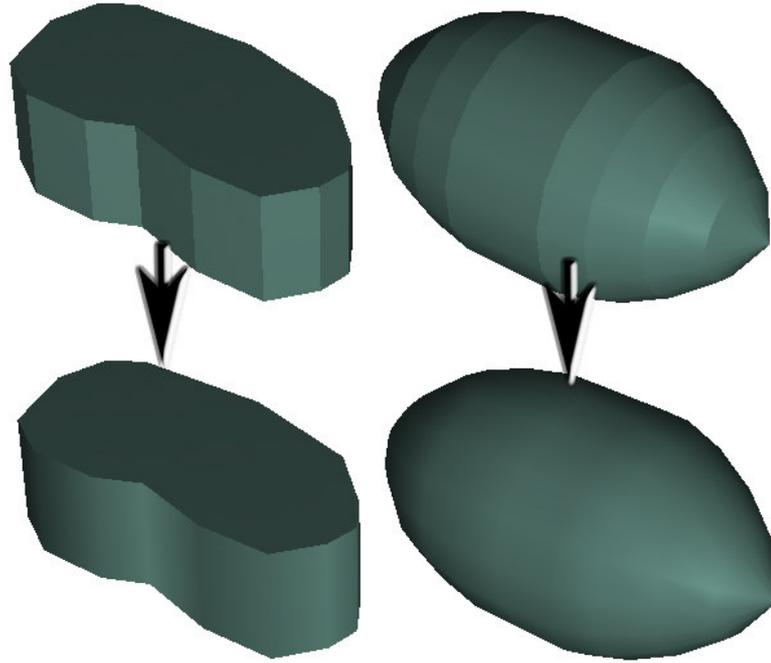
manera similar a como lo hemos hecho en el modo anterior, con la diferencia de que el punto que seleccionemos con el ratón será eliminado.

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:



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:



Efecto del suavizado de normales en una extrusión (izquierda) y en 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.

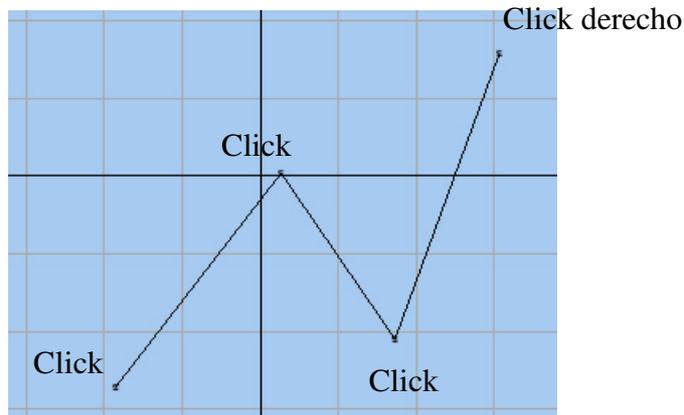
E) Zona de definición del volumen: Una vez creada la superficie, el volumen generado queda definido mediante una serie de parámetros accesibles desde esta zona. Todas las partes de la interfaz comentadas hasta ahora son comunes para todo tipo de sólidos mientras que esta zona será específica del tipo 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.



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

Sólido por extrusión.

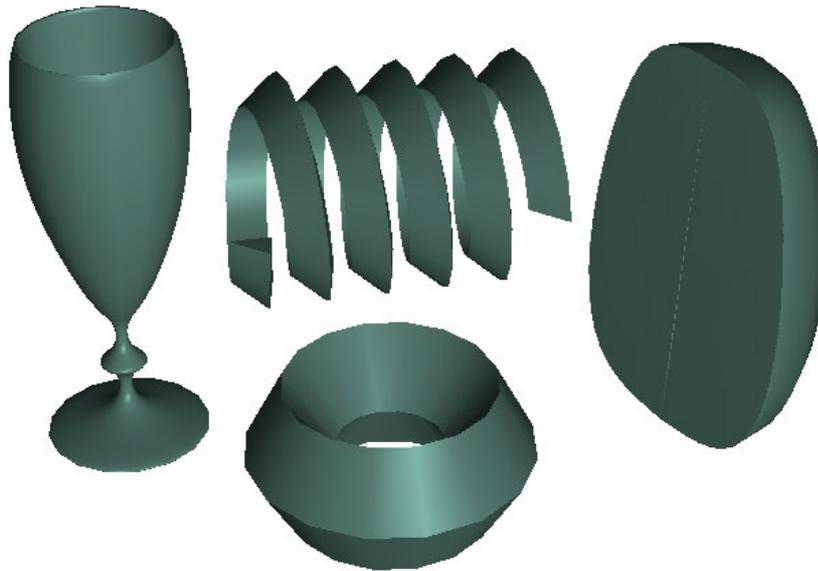
La construcción de este tipo de sólido se produce mediante la traslación a lo largo del eje Z de una superficie cerrada definida en el plano XY. Para este tipo de sólidos la interfaz tomará el aspecto de la figura XX (la de muestra).

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.

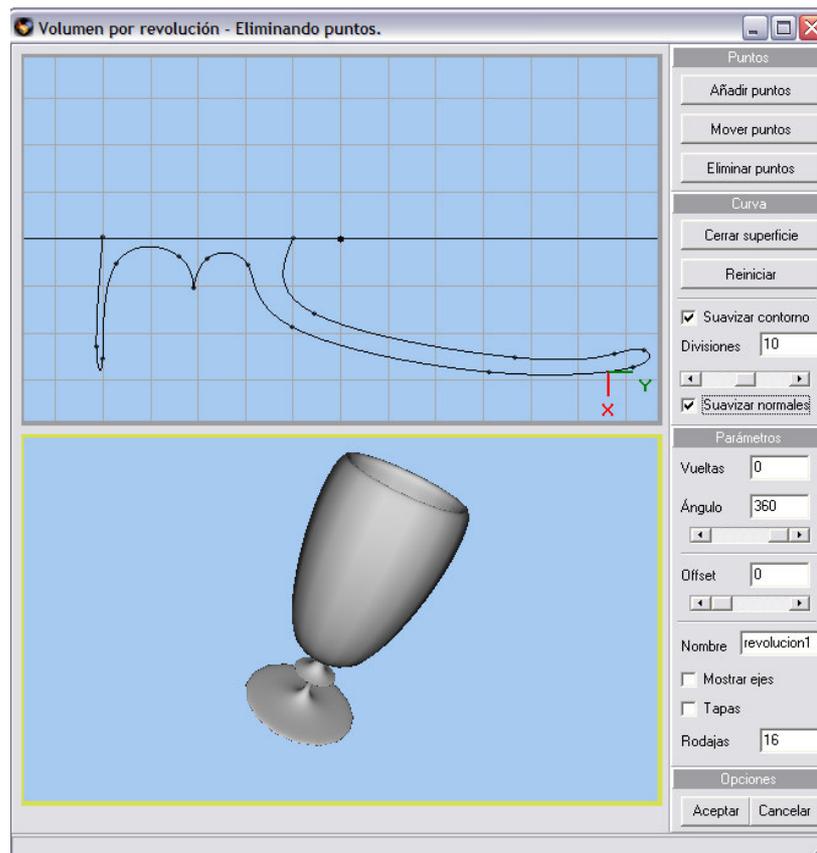
Sólido por revolución

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.



Diferentes sólidos que se pueden construir mediante revolución

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

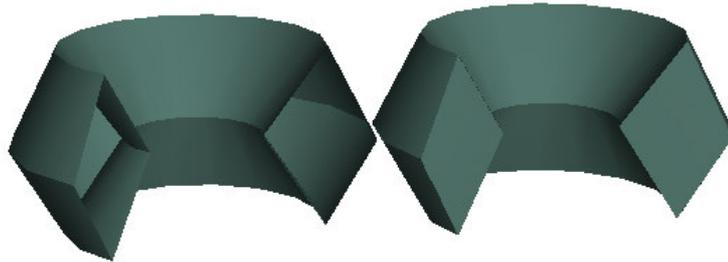


Interfaz para la construcción de sólidos de revolución.

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

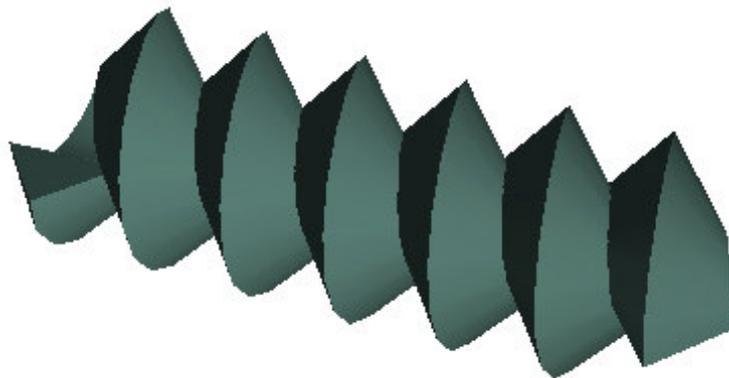
más complejo y más versátil, un gran tipo de sólidos pueden describirse de este modo. El parámetro vueltas se refiere al número de veces que la superficie gira una vuelta entera en torno al eje Y. A menos que no cambiemos el offset este parámetro debería ser 0. El parámetro ángulo se refiere al ángulo 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.



Efecto de las tapas para un ángulo menor de 360

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:

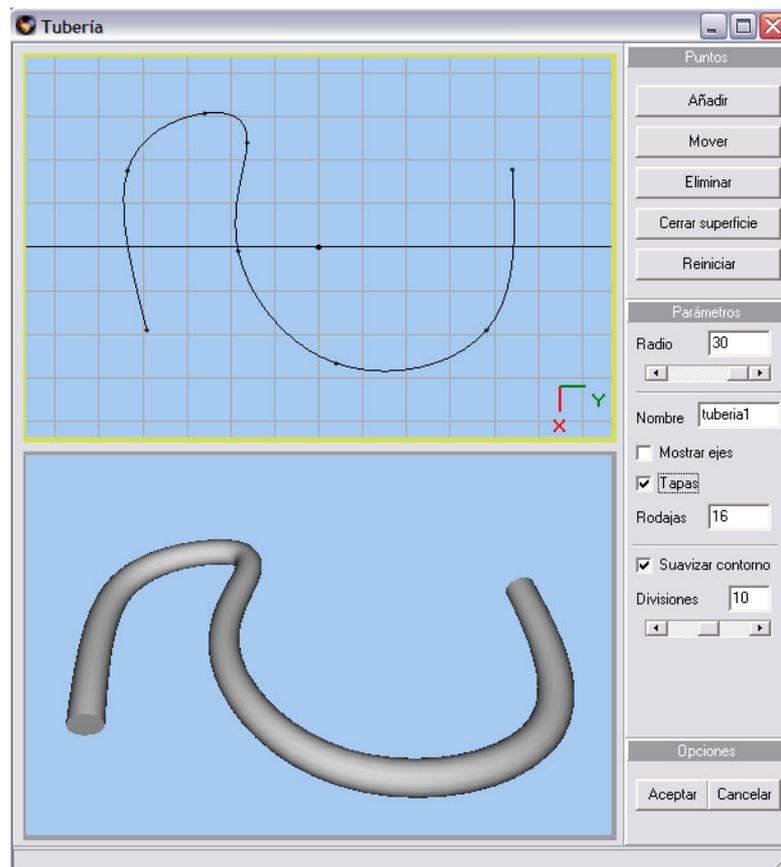


Sólido helicoidal, con offset no nulo.

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.



Interfaz para la construcción 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.

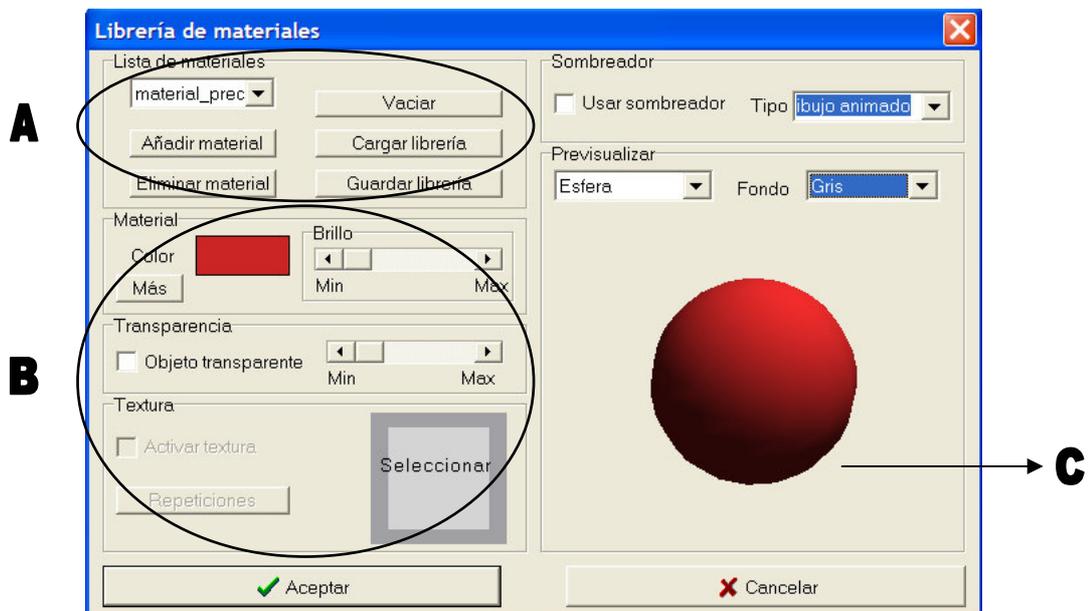
Librería de materiales.

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 [] de la barra de herramientas o mediante la propiedad material del objeto (Panel lateral > Panel Modificar > Propiedades Generales > Botón Material).



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:

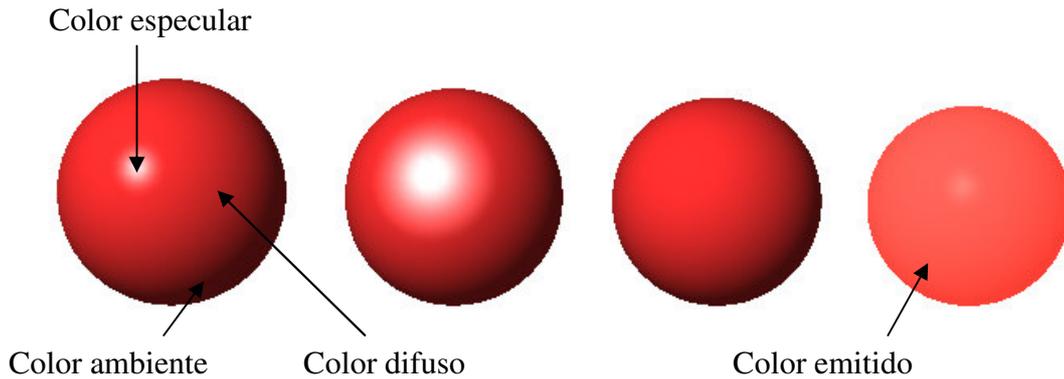
Color ambiental: es el color de la sombra del objeto, es la parte iluminada por la luz que rebota en el entorno, por la luz ambiente, a diferencia de luz que incide directamente en el objeto.

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 presenta 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:



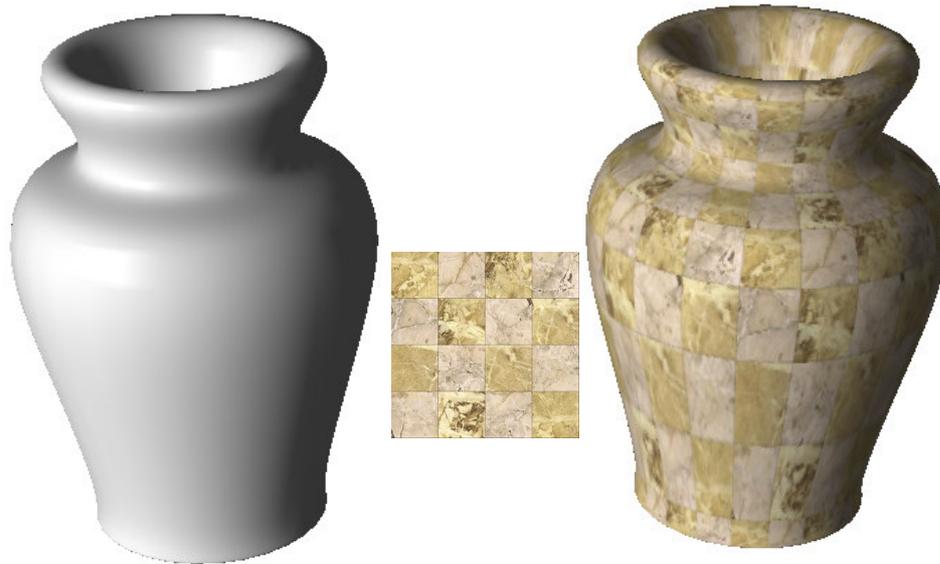
Diferentes combinaciones de colores para un mismo objeto.

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:



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:



En el material del 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.

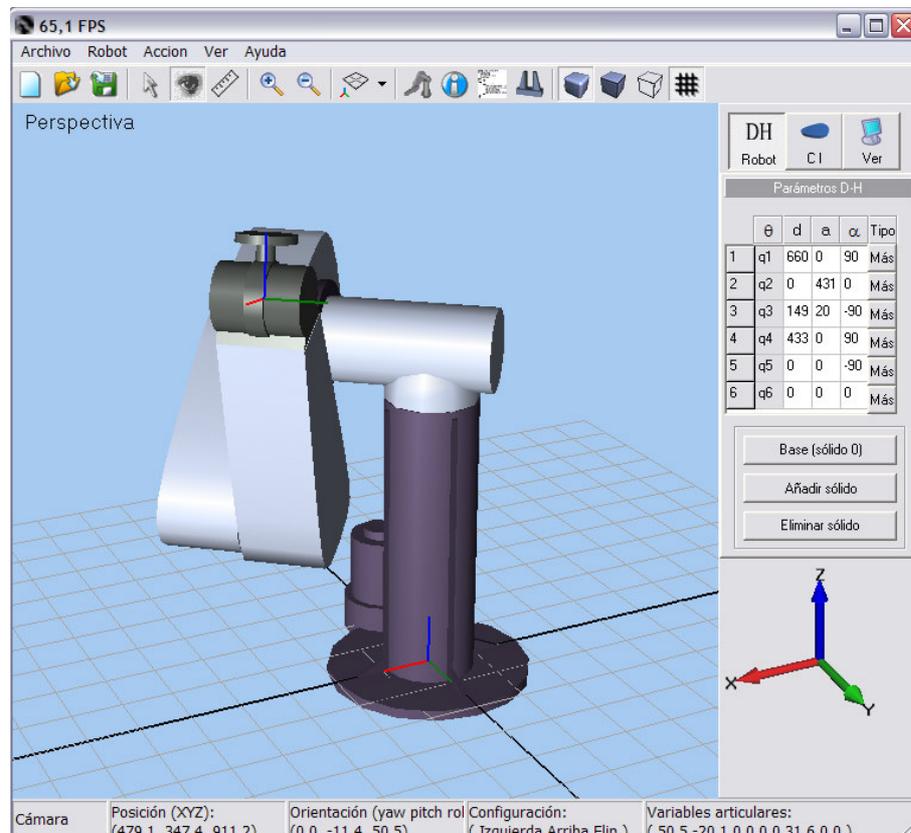
4. El constructor de Robots

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.

4.2 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.



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 [], 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 [] 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.

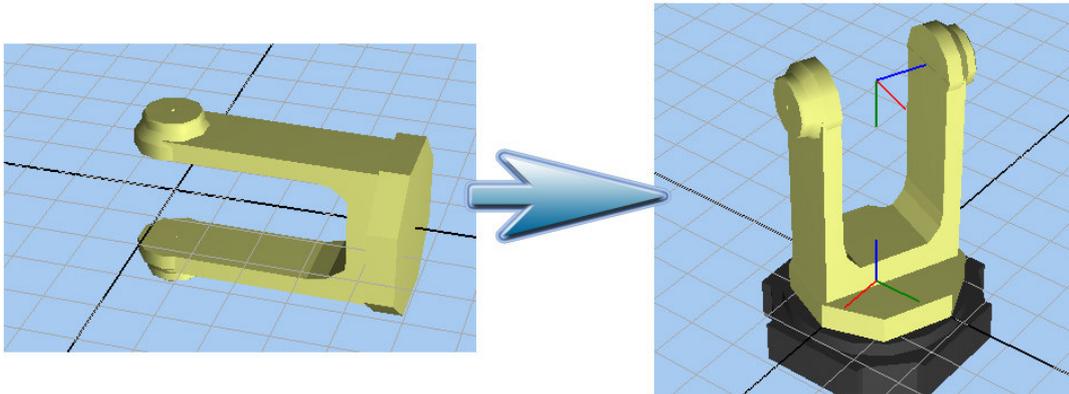
En esta parte de la pantalla se procede a la representación tridimensional del robot que está siendo creado. Aparte del robot podemos observar la cuadrícula o rejilla y los ejes principales, representados por dos líneas más gruesas que se cruzan en el origen, de manera idéntica a como se hace en el constructor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

4.3 Metodología 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:



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:



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 , α_i y θ_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.

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

θ_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.

	θ	d	a	α	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

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, θ_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 > panel Ci > Botón Definir reposo) o desde el menú principal (menú principal > Robot > Definir posición 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

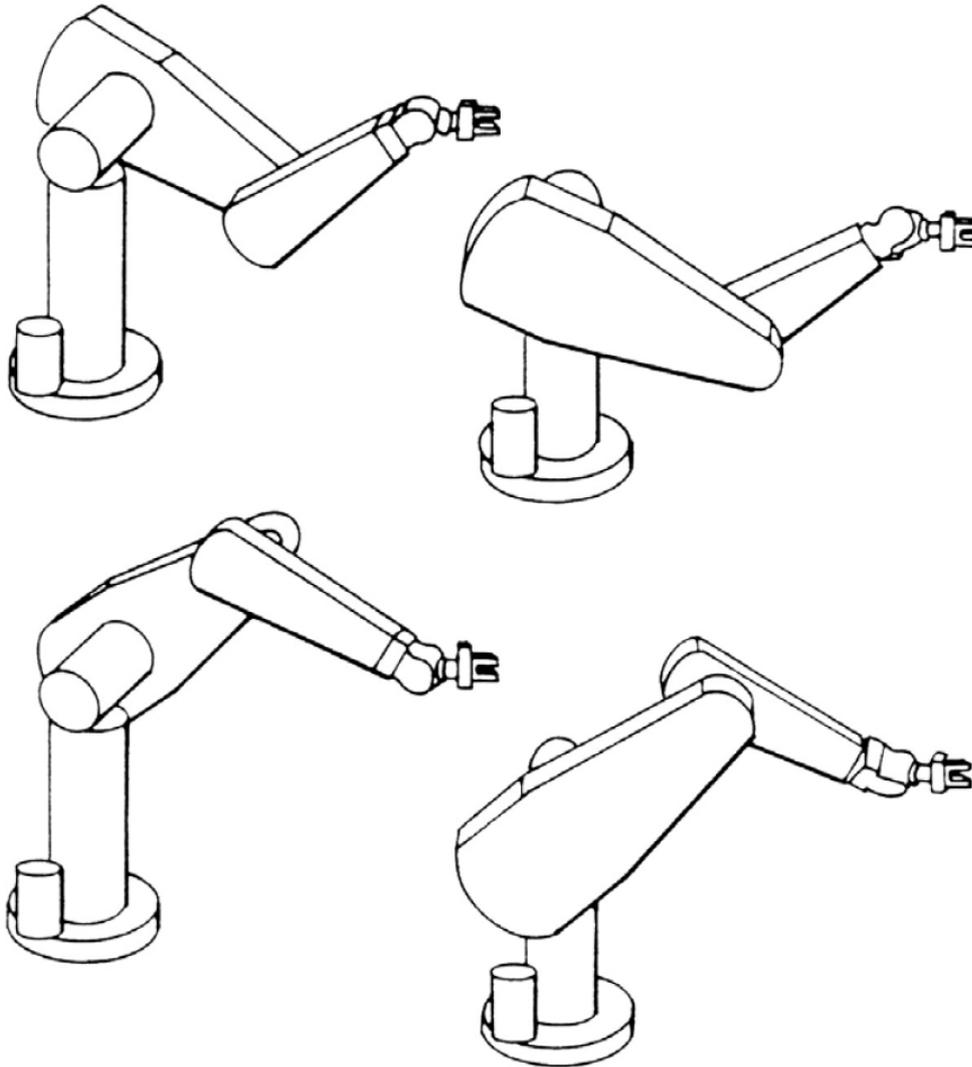
Deberemos introducir el valor de cada articulación en la posición de reposo. Inicialmente el botón de Aceptar está bloqueado. Para desbloquearlo deberemos pulsar el botón de comprobar Rangos. Si hay alguna articulación fuera de rango se nos avisará, y en caso de que estén todas entre sus correspondientes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

4.4 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.

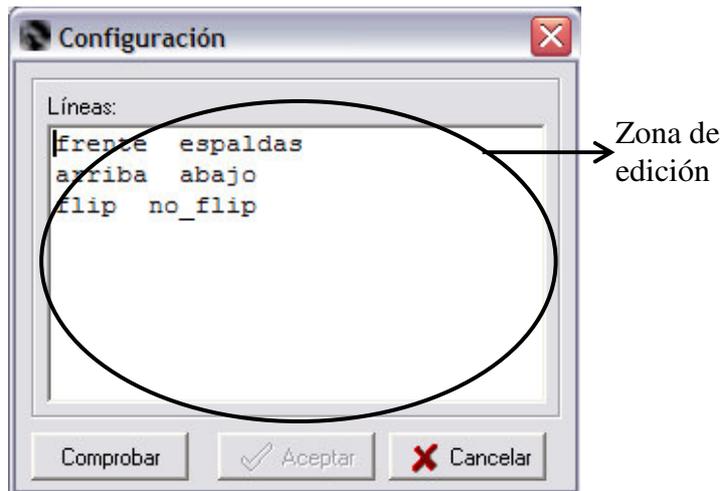
Definición de la configuración

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:



Diferentes configuraciones que puede adoptar el robot para llegar al mismo destino.

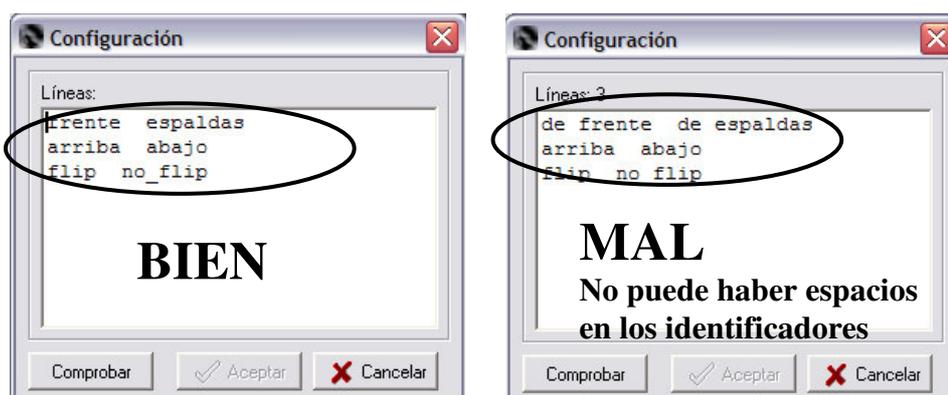
Antes de proceder a la implementación programada del modelo inverso del robot deberemos definir cuales será las posibles configuraciones que podrá adoptar nuestro robot, y lo haremos mediante una ventana específica que será accesible desde el panel lateral (panel lateral > panel CI > Botón Definir configuración)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



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:

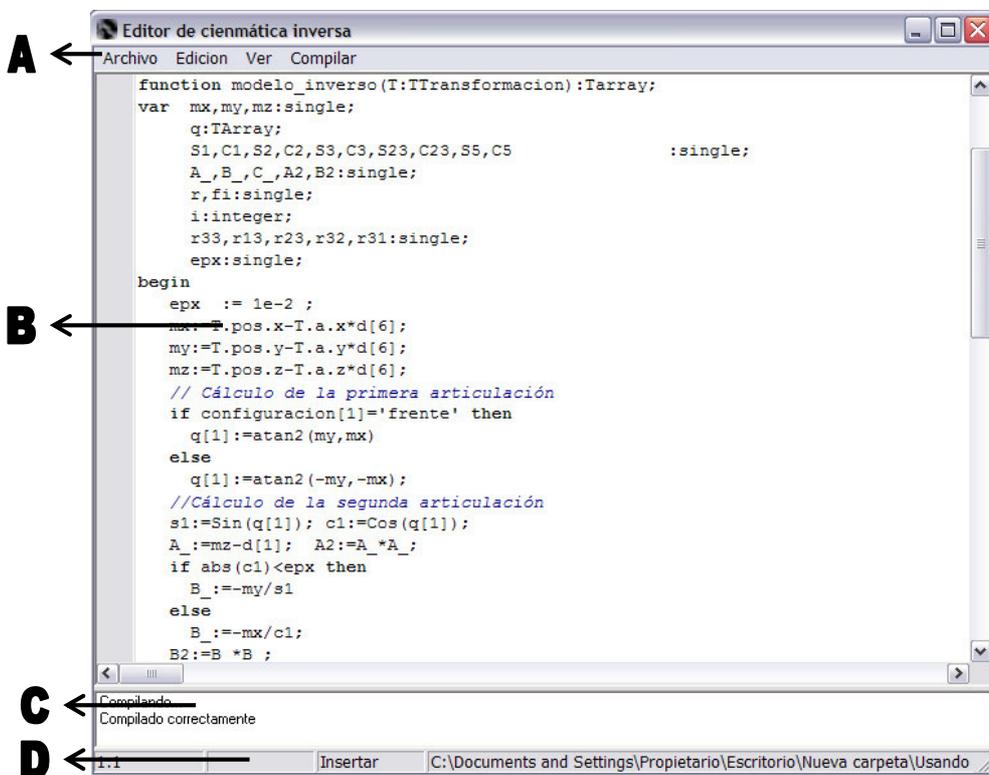


Para que la implementación de las posibles configuraciones sea efectiva deberemos pulsar el botón de aceptar, botón que se encontrará bloqueado hasta que sean revisados los errores mediante el botón comprobar. Si hubiera algún error se nos avisaría mientras que si estuviéramos exentos de errores se aplicarían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



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	$1.5 \times 10^{-45} .. 3.4 \times 10^{38}$	7-8	4 bytes
Double	$5.0 \times 10^{-324} .. 1.7 \times 10^{308}$	15-16	8 bytes
Extended	$3.6 \times 10^{-4951} .. 1.1 \times 10^{4932}$	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 *TTransformación* que representa una matriz de transformación homogénea, que nos permitirá decir la posición y la orientación:

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

Así, para una transformación *T*, de tipo *TTransformación*, 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:TTransformacion):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 Grados(radianes:single):single; // Convierte a grados
function Radianes(grados: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.

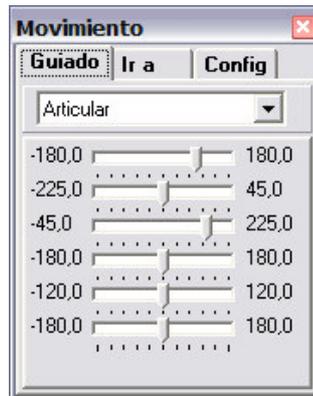
Bien desde el menú principal (Menú principal > Compilar > Compilar el modelo inverso) o desde la ventana principal de la aplicación (Panel lateral > Panel CI > Compilar modelo Inverso) podremos compilar el script del modelo inverso. La zona de mensajes inferior nos mostrará las advertencias necesarias 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 he 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.



Guiado del robot.

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

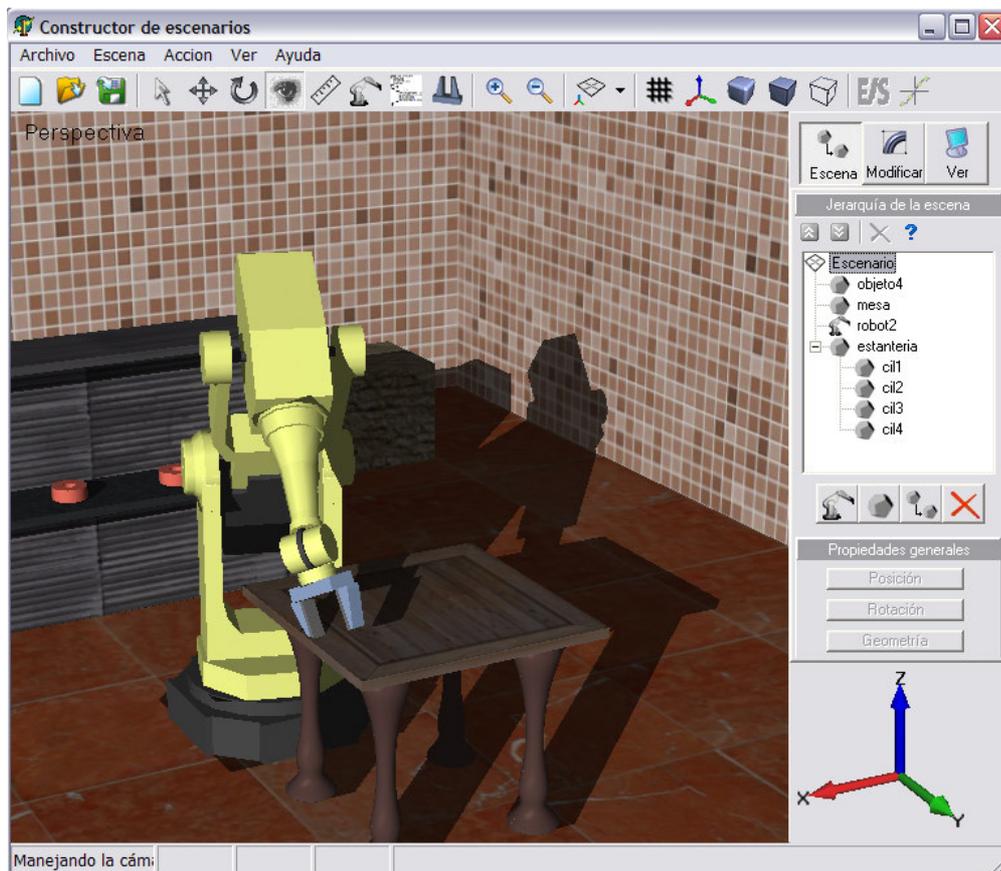
Aparecerán en este modo una serie de controles para poder especificar la configuración con la que se resolverá el modelo inverso. Como ya se ha explicado suficientemente antes, podremos llegar al mismo destino de varias formas. Se habilitarán una serie de selectores, tantos como pares de configuraciones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

5. El constructor de escenarios.

Una vez creados los sólidos y los robots, toma parte el último y más importante módulo, en el que podremos construir un escenario virtual robotizado y proceder a su explotación. Tal y como su nombre indica, este módulo tiene dos partes diferenciadas, la parte referente a la construcción del escenario y la parte referente a la explotación del mismo por parte de los robots industriales. La construcción se realizará de forma análoga a como realizábamos la construcción de los objetos, con la diferencia que trabajaremos a partir de sólidos previamente creados con el constructor de objetos, pudiendo definir una relación jerárquica entre ellos. Una vez construido el escenario virtual en el que se habrán de desenvolver los robots procederemos a la explotación del mismo, bien guiando los robots, registrando puntos y creando secuencias o bien de forma mucho más versátil, programando los robots mediante un lenguaje robótico habilitado mediante scripts.

5.1 Descripción de la interfaz de usuario

De cualquier forma, veamos la interfaz de usuario que presentará este módulo:



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 escenario, abrir uno existente, empezar uno nuevo o bien abandonar la aplicación (estos conceptos serán explicados con más profundidad posteriormente)

Escena: Mediante este menú se podrá editar la escena, añadiendo integrantes, que podrán ser de tipo robot (Acción *Nuevo robot*) o de tipo sólido (Acción *Nuevo sólido*), o bien eliminando integrantes (Acción *Eliminar seleccionado*). De todas formas, para obtener un mayor control sobre la jerarquía de la escena se ha habilitado un control específico en el panel lateral, que comentaremos con profundidad posteriormente.

Acción: Desde este menú se pueden seleccionar todos los modos de operación posibles, que podrán ser de dos tipos, los referentes a la creación de la escena, equivalentes a los ya existentes en el módulo constructor de objetos, como trasladar objetos, rotar objetos, manejar la cámara o realizar mediciones, y modos de operación específicos de la parte de la explotación de la escena, como Guiar robot, programar robot o añadir herramienta al robot, que serán explicados más adelante.

Ver: Desde este menú se tiene acceso a todas las opciones referentes a la visualización de la escena en tres dimensiones, de manera idéntica a como se realiza en el resto de módulos analizados, con dos opciones añadidas, la opción de acceder a la ventana de registro de trayectorias (Opción *Ventana registro*) y a la ventana de Entradas u salidas binarias (Opción *Ventana E/S*).

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 ocho 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 que se vean los ejes (el sistema de referencia) del escenario.

Los siguientes seleccionan el tipo de representación de los objetos.

C) Panel de control lateral

Desde este panel tendremos un control total sobre la escena que estemos creando, siendo una parte primordial en lo referente a la construcción del escenario robótico.

De la misma manera que ocurría en los anteriores módulos este panel se compone de tres subpaneles, accesibles desde los tres botones de control situados en su parte superior.

El primer subpanel es el más importante y se encarga de la creación y modificación del escenario virtual. Este panel está subdividido a su vez en dos partes, la parte superior, de nombre Jerarquía de la escena, donde se define la relación entre los diversos elementos que componen la escena, y la parte inferior, a través de la cual se puede acceder a las propiedades de cada objeto como Posición, orientación y geometría.

Podremos cambiar la geometría asociada al elemento seleccionado pulsando el botón *geometría*.

Mediante los botones de *orientación* y de *posición* podremos especificar numéricamente tanto la posición como la orientación del elemento seleccionado, mediante las correspondientes ventanas de diálogo que ya utilizáramos en el constructor de objetos para estos menesteres.

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 o su orientación cuando estemos construyendo la escena.

E) Zona de visualización en tres dimensiones.

En esta parte de la pantalla se procede a la representación tridimensional del escenario virtual que está siendo creado y una vez creado podremos observar el resultado del movimiento de los robots. Aparte del escenario propiamente dicho 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 hemos venido haciendo en los anteriores módulos. Cuando estemos creando el escenario estarán accesibles el resto de las opciones, mientras que cuando estemos ejecutando alguna aplicación robótica, bien mediante programación o bien mediante la ejecución de una serie de trayectorias aprendidas mediante guiado se invalidarán el resto de controles (menos el de la cámara) de forma que no podamos modificar la escena mientras haya elementos en movimiento.

Modos de Operación

Este es el módulo que más modos de operación tiene, puesto que es el módulo en el que más acciones podremos realizar, tanto referentes a la construcción como referentes a la explotación.

Tendremos cinco modos que nos serán familiares de otros módulos como serán el modo de selección de objetos, el de traslación de objetos, el de rotación de objetos, el de manejo de la cámara y el de medida de distancias, accesibles todos ellos bien desde el menú principal (Menú principal > Acción) o desde la barra de herramientas. El trabajo

con estos modos de operación se realiza de manera análoga a como hemos venido haciendo tanto en el constructor de objetos como en el constructor de herramientas y que ya ha sido suficientemente explicada.

Los siguientes tres modos de operación son el modo de guiado de robots, el modo de programación de robots y el modo de añadir herramientas, que serán explicados cuando expliquemos la parte referente a la explotación de la escena.

5.2 Creación del escenario

Tal y como hemos comentado suficientemente, esta es una de las dos partes fundamentales de este módulo, en la que nos encargaremos de construir nuestro escenario virtual bien a base de sólidos simples, creados mediante el constructor de objetos, o bien a base de robots, creados mediante el constructor de robots.

Añadir objetos.

Se puede añadir un sólido a nuestra escena desde el menú principal (*Menú principal > escena > Nuevo objeto*) o desde el panel lateral (*Panel lateral > panel escena > panel jerarquía de la escena > botón [J]*). Nos aparecerá la siguiente ventana:



Ventana para cargar un sólido

Podemos elegir 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 una muestra del objeto. Si aceptamos, habremos insertado un objeto en la escena con la correspondiente geometría. En caso de que no cargáramos ninguna geometría, el objeto no tendría asociada ninguna geometría y podría servir, por ejemplo para la explotación del escenario para definir puntos de paso o de aproximación a objetos, si bien se recomienda definir este tipo de puntos desde la aplicación robótica, pero que puede ser de utilidad en caso de estar en modo de registro de trayectorias para ejecutarlas posteriormente.

Cada sólido que añadamos al escenario tiene asociado un sistema de referencia, 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, de igual manera a como hemos explicado en la parte correspondiente del constructor de robots.

Añadir robots.

Se puede añadir un robot a nuestra escena desde el menú principal (*Menú principal > escena > Nuevo robot*) o desde el panel lateral (*Panel lateral > panel escena > panel jerarquía de la escena > botón []*). Nos aparecerá la siguiente ventana:



Ventana para cargar un robot.

Podemos elegir el robot que queremos insertar pulsando el botón *Cargar robot*. 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 robots. A continuación se nos mostrará una muestra del robot que queremos cargar y si aceptamos ya habremos insertado un robot en nuestra escena.

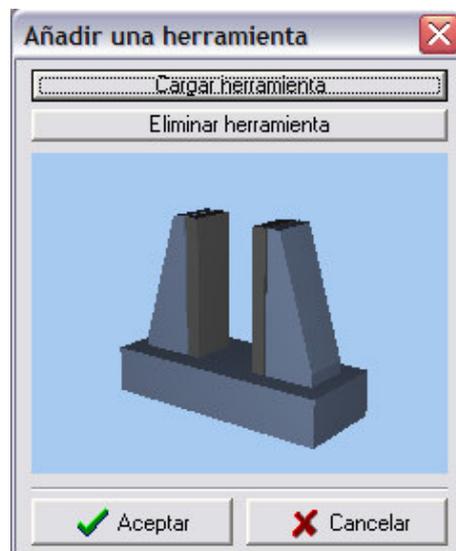
De igual manera que hemos comentado para los objetos, cada robot tiene asociado un sistema de referencia, que en este caso es el sistema de referencia asociado a la base del robot. En principio, este sistema de referencia es coincidente con el sistema de referencia absoluto, conocido comúnmente en robótica como sistema de coordenadas mundo. De igual manera que podemos trasladar y rotar los sólidos, también podremos realizar las mismas operaciones con los robots, y lo que estaremos haciendo internamente será cambiar el sistema de referencia mundo para cada robot, no hará falta especificarlo en la aplicación robótica como suele ser habitual en estos casos. La especificación de los destinos para cada robot se realizará con respecto al sistema de

referencia absoluto, el sistema de referencia mundo, y el cambio necesario para adaptar el sistema de referencia mundo con el asociado a cada robot (normalmente mediante una instrucción tipo WORLD o similares) se realizará automáticamente cuando cambiemos la posición u orientación de un robot mientras editamos la escena.

Asociar herramientas a los robots

Una vez que hayamos creado un robot y lo hayamos insertado en la escena podremos proceder a asociarle una herramienta, que también deberemos tener creada. Para ello tendremos que acceder al modo de operación de Añadir herramientas, accesible desde el menú principal (*Menú principal > Acción > Añadir una herramienta al robot*) o desde la barra de herramientas mediante el botón []

Una vez que estemos en este modo de operación deberemos seleccionar con el ratón, bien en el visor tridimensional, bien en el control de la jerarquía de la escena el robot al que le queramos asociar una herramienta y obtendremos una ventana como la siguiente.



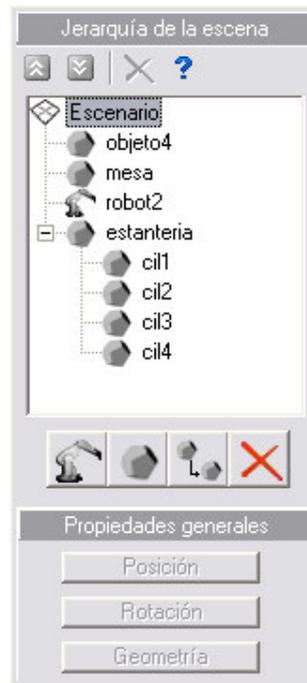
Ventana para añadir una herramienta al robot seleccionado.

En esta ventana podemos elegir la herramienta que queremos pulsando el botón Cargar Herramienta. 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 herramientas integrado en el constructor de robots. A continuación se nos mostrará una muestra de la herramienta escogida. Si el robot ya tuviera una herramienta asociada nada más activar esta ventana ya aparecería una muestra de la herramienta actual del robot. Si aceptamos, habremos asociado una herramienta en el robot seleccionado.

Jerarquía de la escena

Los diferentes elementos que conforman nuestro escenario virtual presentan una estructura arborescente.

Para poder controlar esta estructura, se ha creado un control de árbol en el panel lateral (*Panel lateral > Panel escena > Panel Jerarquía de la escena*), a través del cual se podrán editar y modificar todas las relaciones jerárquicas entre los objetos.



Panel de control de la jerarquía de la escena y ejemplo de estructuración del entorno.

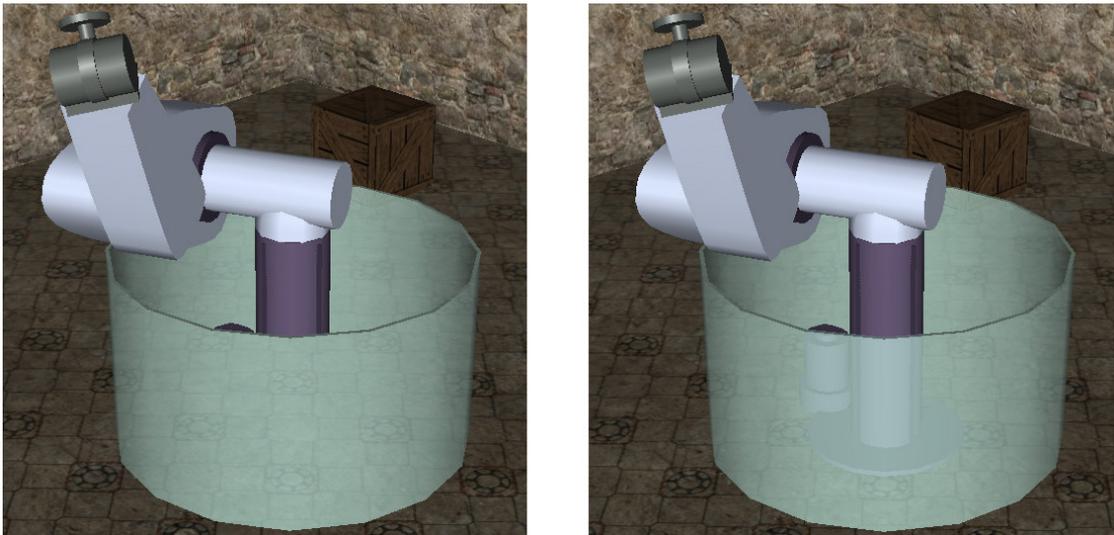
Tendremos siempre visible en este control el elemento Escenario, que representa la referencia absoluta, y el resto de objetos se definirán con respecto a él. Los sólidos se podrán arrastrar dentro del árbol, para así crear las estructuras complejas de jerarquía que podamos necesitar. En cuanto a los robots surgirán varias limitaciones, se tendrán que ‘colgar’ directamente del escenario y no podrán tener ningún hijo, esto se realizará internamente cuando manipulemos objetos, pero no desde este módulo. Evidentemente tampoco se podrá arrastrar un objeto a uno de sus hijos, porque se crearía un bucle infinito. Los robots tienen el icono [] mientras que los objetos tienen el icono []. Podremos cambiar el nombre de cada elemento si pulsamos con el ratón encima del nombre.

Cuando arrastremos un elemento, también arrastraremos con él a sus hijos, esto es, a todos los elementos que cuelgan de él. Del mismo modo cuando cambiemos la posición o la orientación del elemento, también cambiaremos la de sus hijos, mientras que si cambiamos la de sus hijos la del padre permanecerá intacta, ahí la utilidad de la habilitación de una estructura arborescente para la definición de la escena.

A la hora de trasladar u orientar los objetos nos encontraremos diferencias según lo hagamos interactivamente mediante los modos de operación correspondientes o numéricamente. Si lo hacemos interactivamente estaremos modificando la posición y/o orientación absolutas del objeto seleccionado (y de sus hijos si los hubiere) mientras que si especificamos la situación del elemento numéricamente lo haremos con respecto a su predecesor.

Todos estos conceptos tratados de estructura arborescente pudieran dar pie a propiciar un modelo del universo en la programación del robot, como ocurre en lenguajes como AL, donde existen instrucciones de atadura y desatadura como AFFIX/UNFIX, pero sólo podremos crear una estructura del universo mientras estemos construyendo el escenario, no así cuando estemos en modo de explotación del escenario. Una vez construido, procederemos a la programación del robot de una manera tradicional, sin instrucciones de atadura, aunque, como explicaremos más adelante podremos importar información del árbol mediante sencillas instrucciones habilitadas para tal efecto.

También podremos mover ascendentemente o descendentemente los objetos mediante los botones [] y [] respectivamente, ambos situados en la parte superior del control de árbol. Este efecto no tendrá en principio ningún efecto práctico, pero sí pudiera tener algún tipo de efecto visual. Internamente, los objetos se pintan en la pantalla según el orden del árbol, verticalmente y hacia abajo. Esto tendrá relevancia cuando tengamos algún objeto transparente, puesto que un objeto transparente podría causar efectos no deseados si se pintara al principio de la escena, tal y como podemos ver en la siguiente imagen. Para solucionar esto, deberemos colocar los elementos que tengan transparencias de algún tipo al final de la jerarquía para poder visualizar correctamente la escena.



En la imagen de la izquierda hay incorrecciones en el uso de las transparencias.

Así, tal y como podemos ver en la imagen del ejemplo, tenemos tres objetos colgados directamente de la raíz de la escena, al mismo nivel: un robot, el fondo y un objeto transparente. En la imagen de la izquierda el orden de los objetos de la escena es fondo > objeto transparente > robot. Como se puede ver, el objeto transparente lo es para el fondo (va después de él en la jerarquía), pero es opaco para el robot, con lo que el efecto óptico conseguido es erróneo, o cuanto menos, extraño. En la imagen de la derecha el objeto transparente está situado al final de la jerarquía y el efecto conseguido es el correcto. En cuanto al orden de la colocación entre el robot y el fondo resulta irrelevante.

5.3 Explotación del escenario

Así pues, Con todo lo explicado hasta ahora, podremos construir un escenario virtual robotizado, lo que constituye la primera parte, al menos conceptual del proyecto. Ahora nos toca acometer la segunda parte, la parte referente a la explotación del escenario por parte de los robots industriales. En esta parte podremos guiar los robots, realizar un registro de trayectorias o programarlos para que realicen una tarea. También podremos comunicarnos con ellos mediante una serie de entradas y salidas binarias.

Aunque conceptualmente sea un concepto distinto, se comparte interfaz con la parte de construcción del escenario y podremos combinar ambos conceptos. Podemos, por ejemplo, ejecutar una tarea robótica, y cuando el robot termine la tarea podremos cambiar la posición de los objetos, o del robot mismo sin salir del programa y volver a ejecutar la aplicación robótica, realizando todos los cambios que queramos. La única restricción que tenemos es que mientras un robot esté realizando una tarea (definida mediante programación o mediante registro de trayectorias) no podremos realizar operaciones relacionadas con la construcción/manipulación del escenario, lo cual, por otra parte resulta lógico.

Guiado de robots

Ya podíamos guiar robots, aunque con ciertas limitaciones, en el módulo de construcción de robots, mediante una ventana de guiado que poseía las funciones básicas que se pueden encontrar, por ejemplo en una botonera de guiado de un robot industrial.

En este módulo tendremos acceso a la misma ventana para guiar robots, pero accederemos a ella de otro modo, puesto que podremos guiar varios robots. Tal y como comentábamos antes, tenemos varios modos de operación, que nos definen el comportamiento interactivo del ratón con el visor en tres dimensiones, permitiéndonos modificar el escenario de forma interactiva, pero también tendremos modos de operación en la parte referente a la explotación de escenarios, como por ejemplo para el guiado, tendremos el modo de operación guiado, accesible desde el menú principal (*Menú principal > Acción > Guiar robot*) o desde la barra de herramientas mediante el icono []. Así pues, una vez que estemos en modo guiado, para proceder al guiado de un robot de la escena deberemos poner el puntero del ratón encima y pulsar el botón izquierdo, tras lo cual aparecerá la conocida ventana de guiado, mediante la cual podremos controlar el robot que hayamos seleccionado. Si quisiéramos manejar otro robot estando en este modo de operación sólo tendremos que realizar la misma operación y la ventana de guiado se adecuará al nuevo robot seleccionado.

Registro de trayectorias

La enseñanza por guiado es uno de los dos métodos básicos de programación que existen, en el cual guiaremos el robot para que alcance las situaciones deseadas y procederemos a su registro. Posteriormente el robot podrá reproducir todas las situaciones almacenadas.

Desde un punto de vista conceptual es el modo de programación más simple. Tiene sus antecedentes históricos en las aplicaciones de teleoperación (industria nuclear, submarina, etc.).

Este registro resulta útil en aplicaciones como la carga y descarga de máquinas o la soldadura por puntos. La enseñanza de las situaciones a memorizar se realiza mediante la ventana de guiado.

La principal ventaja de este método de programación es la simplicidad conceptual, lo que permite que el usuario del programa pueda desenvolverse aunque sus conocimientos de informática y programación sean mínimos.

A diferencia de lo que ocurre en una gran parte de los robots industriales, en los que mediante este modo de programación lo que se almacena son las variables articulares del robot, nosotros almacenamos la situación (posición + orientación) del elemento terminal del robot, con lo que conseguimos portabilidad para otros robots, algo imposible si el registro fuera articular.

Ahora bien, este sistema de programación presenta varios inconvenientes y limitaciones, como por ejemplo la poca adaptación del robot a cambios en el entorno de trabajo, no podemos comunicarnos con el exterior y sobre todo la mínima capacidad expresiva como lenguaje de programación, puesto que la estructura de los programa no puede ser más que lineal (no existen discriminaciones condicionales, ni iteraciones, etc).

Para proceder al registro de trayectorias deberemos activar el modo de guiado y seleccionar el robot que deseemos programar. Una vez seleccionado el robot tendremos disponible la opción de registro de trayectorias, bien desde el menú principal (*Menú principal > Ver > Ventana de registro*) o bien desde el icono [] de la barra de herramientas. Nos aparecerá una ventana como la que se muestra en la siguiente imagen:



Lo primero es posicionar al robot en la situación que deseemos registrar, para ello podremos guiarlo mediante la ventana de guiado o podremos utilizar la función habilitada en la propia ventana de registro de puntos *Importar*. Escribiremos el nombre del elemento del árbol al que deseemos ir y pulsaremos el botón de *Ir a* situado justo en la parte inferior de la casilla de edición de texto.

Una vez que tengamos al robot donde queramos, procederemos al registro del punto mediante el panel *Situación actual*, situado en la parte inferior de la ventana de registro. Dentro de este panel podremos elegir tanto el nombre del punto como la forma de insertarlo en la lista de puntos. Si elegimos la opción *Añadir*, lo añadiremos al final de la lista, mientras que si escogemos la opción *Insertar* lo insertaremos justo antes del punto que esté seleccionado en ese momento.

Los puntos registrados aparecerán dentro del panel *Registro de puntos*, situado en la parte superior de la ventana, y en este panel podremos seleccionar el punto activo, pulsando el botón izquierdo del ratón cuando estemos encima, podremos ejecutar la secuencia completa mediante el botón *Ejecutar* o bien podremos borrar la secuencia mediante el botón *Reiniciar*.

Cuando tengamos seleccionado algún punto tendremos acceso a él mediante el panel *Punto seleccionado*, y las opciones de que disponemos son de ir a dicho punto desde la posición actual del robot (botón *Ir a*), ir a dicho punto en línea recta (botón *Ir recto*) o bien eliminar dicho punto de la lista de puntos registrados mediante el botón *Eliminar*.

Programación textual del robot.

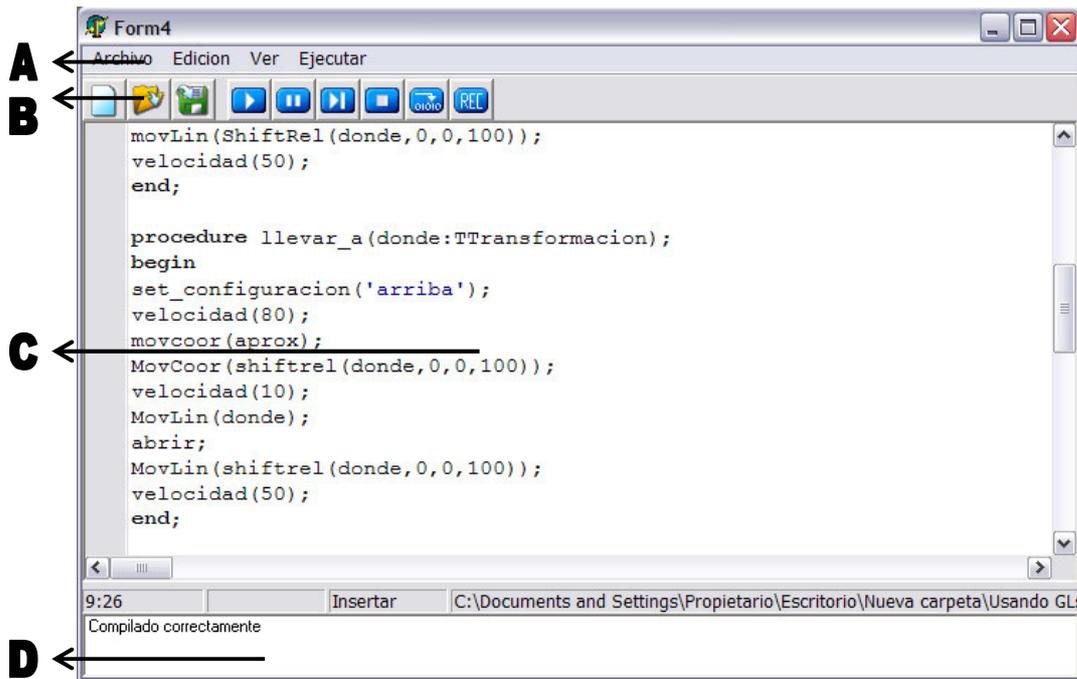
Como alternativa a la programación por guiado, el método de programación textual permite indicar la tarea que debe realizar el robot mediante el uso de un lenguaje de programación específico. Un programa robótico se corresponde con un texto que deberemos editar compuesto por una serie de órdenes que posteriormente ejecutará el robot.

Entorno de programación

De la misma manera que para proceder al guiado de un robot necesitamos activar el modo de operación necesario, para programar un robot necesitaremos activar el modo de operación de programación de robots, accesible bien desde el menú principal (menú principal > Acción > Programar robot) o bien desde el botón de la barra de herramientas.

Una vez que hayamos activado este modo de operación deberemos seleccionar el robot que queremos programar mediante el ratón, tanto en el visor tridimensional como a través del control de árbol situado en el panel lateral.

Una vez seleccionado el robot que deseemos programar aparecerá la interfaz de usuario del editor de programa robótico:



Interfaz para la definición del programa robótico.

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 Ejecutar. Desde este submenú tenemos acceso las diferentes opciones de ejecución del programa, que serán comentadas más adelante y que son las siguientes: Compilar aplicación, ejecutar instrucción, ejecutar aplicación, detener ejecución, pausa y grabar en video.

B) Barra de herramientas

Desde esta zona tendremos acceso a las funciones más habituales. Los tres primeros iconos de la izquierda permiten, respectivamente, generar un archivo nuevo, abrir uno existente o bien guardar en un fichero el actual programa robótico.

Los siguientes hacen referencia a las diferentes opciones de ejecución del programa, tal y como las acabamos de comentar y que explicaremos con profundidad más adelante.

C) Zona de edición de código fuente:

En esta zona procederemos a escribir el código fuente del programa robótico. 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.

D) 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.

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.

Tipo de ejecución del programa

Compilar aplicación. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Compilar aplicación*) o mediante el botón [] de la barra de herramientas o mediante la combinación de teclas Ctrl+F9. No es una compilación en el sentido estricto de la palabra puesto que no generamos ningún fichero ejecutable, no generamos código máquina sino código interno que entiende el motor de scripts utilizado. Al ejecutar esta acción, si hubiera algún fallo de sintaxis en nuestro programa robótico seríamos avisados, marcándose en rojo oscuro la línea en la cual se encuentra el error y apareciendo en la zona de mensajes una descripción del tipo de error cometido.

Ejecutar Instrucción. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Ejecutar instrucción*) o mediante el botón [] de la barra de herramientas o mediante la tecla F8. Internamente se realiza en primer momento una compilación, así que si hubiera algún error se procedería igual que en el caso anterior y no se ejecutaría nada. Como su nombre indica, de este modo ejecutaremos las instrucciones que componen nuestro programa robótico de una en una, lo que se conoce como ejecución paso a paso, ideal para tener un control mayor y supervisión sobre el flujo de ejecución del programa y para descubrir y resolver errores que pudiéramos haber cometido en la programación. La línea en la cual se halla la instrucción que ha de ejecutarse se remarcará en color azul de forma que podamos seguir mejor la ejecución del programa.

Ejecutar Aplicación. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Ejecutar aplicación*) o mediante el botón [] de la barra de herramientas o bien mediante la tecla F9. Mediante esta opción procederemos a la ejecución del programa robótica. Como en la anterior opción, internamente e realiza una compilación, así que si hubiera algún error se procedería como ya hemos explicado.

Detener ejecución. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Detener ejecución*) o mediante el botón [] de la barra de herramientas o mediante la combinación de teclas Ctrl+F2. Podremos detener la ejecución del programa cuando éste se esté ejecutando o cuando estemos en modo paso a paso.

Pausa. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Compilar aplicación*) o mediante el botón [] de la barra de herramientas. Podremos pausar la ejecución del programa cuando éste se esté ejecutando. La diferencia entre la pausa y la detención del programa es que al pausar el programa podremos seguir ejecutando desde el punto en el que nos quedamos, o podremos

ejecutar paso a paso, pero si queremos detener definitivamente la ejecución deberemos utilizar la acción de detención del programa. Cada vez que ejecutamos una instrucción en modo paso a paso, una vez completada la ejecución de la instrucción se pasa a modo pausa.

Grabar en video. Pausa. Podemos realizar esta acción mediante el menú principal (*Menú principal > Ejecutar > Grabar en video*) o mediante el botón [] de la barra de herramientas. Mediante esta acción se ejecutará la aplicación robótica de forma similar a cuando ejecutamos el programa, con la diferencia de que toda la acción que suceda en el visor tridimensional quedará registrada en un fichero de video de formato *.avi. Si la compilación es correcta, aparecerá un cuadro de diálogo que nos permitirá elegir el nombre del archivo de video en el cual queremos grabar la ejecución. Una vez introducido el nombre nos aparecerá otro cuadro de diálogo mediante el cual podremos elegir la compresión del fichero y/o el codec a utilizar (por ejemplo DivX, XviD, Mpeg4, o los que tengamos instalados en nuestro ordenador). Algunos codecs permiten una parametrización, sobre aspectos estándar como la calidad de compresión, la velocidad de datos o la posibilidad de elegir los cuadros claves. En algunos también podemos realizar una configuración, que dependerá del propio codec, la ventana a la que accederemos mediante esta opción será la que proporciona el fabricante. De cualquier forma, independientemente de la compresión y del codec elegidos el video se grabará a 25 imágenes por segundo.

Modelado del entorno.

El modelo del entorno es la representación que tiene el robot de los objetos con los que interacciona. Aunque ya lo hemos citado antes, conviene diferenciar el modelo del entorno que tenemos a la hora de construir el escenario, donde nos podemos definir relaciones jerárquicas entre los diferentes objetos y el modelo del entorno que tiene el robot a través de la programación textual. A la hora de definir el entorno mientras lo estamos construyendo los objetos constan de propiedades como la posición, la orientación, la geometría y su situación en el árbol (padre y posibles hijos) mientras que para la programación del robot los objetos tendrán solo las propiedades de posición y orientación mediante matrices de transformación homogéneas.

Ahora bien, desde el programa robótico podremos importar información proveniente del árbol, para así evitar tener que definir doblemente la posición de los objetos cuando creamos el escenario y cuando escribimos el programa. Dichas instrucciones de importación forman parte del lenguaje de programación robótica y serán explicadas en su debido momento.

6. Descripción del lenguaje de programación de robots.

Tal y como ocurre en la programación del modelo inverso del robot, el lenguaje de programación habilitado para la programación de una aplicación robótica es un lenguaje basado en pascal, con su misma sintaxis y en el que se podrán usar gran parte de los datos definidos para el lenguaje pascal estándar. Se deberá utilizar programación estructurada, no ha sido habilitada la programación orientada a objetos.

6.1 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*

6.2 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;
```

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 *TTransformación* que representa una matriz de transformación homogénea, que nos permitirá decir la posición y la orientación de un sistema de referencia:

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

Así, para una transformación *T*, de tipo *TTransformación*, 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*.

Aparte de todo lo comentado también tendremos acceso desde el lenguaje 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 Grados(radianes:single):single; // Convierte a grados
function Radianes(grados:single):single; // Convierte a radianes
```

6.3 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, incluyendo estas últimas tanto las primitivas de uso de la herramienta como las primitivas para la gestión de las entradas y salidas binarias del robot.

6.3.1 Primitivas para la importación de datos.

Mediante una serie de funciones habilitadas, podremos incorporar a nuestra aplicación robótica transformaciones que habrán sido previamente definidas mientras construíamos el escenario. Disponemos de las dos funciones siguientes:

```
function Importar_transformacion_relativa(nombre:string):TTransformacion;
function Importar_transformacion_absoluta(nombre:string):TTransformacion;
```

Como ya hemos explicado, el lenguaje de programación del robot no dispone de modelado del entorno, y el árbol que utilizamos para el diseño de la escena sí, por lo que podremos usar estas dos funciones y estructurar el escenario mediante la composición de transformaciones.

Ambas funciones nos devolverán un valor del tipo TTransformacion, y deberemos pasarle como parámetro el nombre que el objeto que queremos importar tiene en el árbol. La diferencia radica en que mediante *Importar_transformacion_absoluta* importamos la transformación absoluta del objeto, mientras que mediante la función *Importar_transformacion_relativa* lo que importamos es la transformación del objeto pero relativa a su antecesor en el árbol. Así pues, si utilizamos esta última técnica, para especificar la transformación total del objeto deberemos realizar una serie de composición de transformaciones, recorriendo el árbol, mientras que si importamos de forma absoluta toda esta serie de operaciones se realizará de forma interna.

6.3.2 Primitivas para el trabajo con transformaciones homogéneas.

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

```
function RotRPY(roll,pitch,yaw:single):TTransformacion;
```

Define una transformación a partir de los ángulos de roll, pitch y yaw.

```
function RotEuler(alfa,beta,gamma:single):TTransformacion;
```

Define una transformación a partir de los ángulos de Euler.

```
function RotX(angulo:single):TTransformacion;
```

```
function RotY(angulo:single):TTransformacion;
```

```
function RotZ(angulo:single):TTransformacion;
```

Rotaciones en torno a los ejes principales.

```
function RotGiro(eje:Tcoordenadas;angulo:single):TTransformacion;
```

Rotación en torno a un eje arbitrario. La especificación del eje de giro se hará mediante el parámetro coordenadas.

```
function NullTr:TTransformacion;
```

Definición de la transformación nula, o lo que es lo mismo, la matriz identidad.

```
function Comp(T1,T2:TTransformacion):TTransformacion;
```

Composición (multiplicación) de dos transformaciones.

```
function Invertir(T:TTransformacion):TTransformacion;
```

Devuelve la inversa de una matriz.

```
function Transf(x,y,z,roll,pitch,yaw:single):TTransformacion;
```

Define una transformación mediante x y z y los ángulos roll, pitch y yaw.

```
function Traslacion(x,y,z:single):TTransformacion;
```

Define una traslación.

```
function Shift(T:TTransformacion; x,y,z:single):TTransformacion;
```

Traslada una transformación respecto de la referencia absoluta.

```
function ShiftRel(T:TTransformacion; x,y,z:single):TTransformacion;
```

Traslada una transformación respecto de la propia referencia.

6.3.3 Primitivas de uso del robot.

```
procedure velocidad(cuanto:single);
```

Define la velocidad de movimiento del robot en tanto por ciento de la máxima permisible.

```
function donde_herramienta:TTransformacion;
```

función que 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.

```
function articulaciones:TArray;
```

Función que nos devuelve el valor de las articulaciones del robot.

```
procedure set_configuracion(conf:string);
```

Procedimiento para cambiar la configuración del robot.

```
procedure stop(ms:integer);
```

Detenemos la ejecución de la aplicación durante un tiempo expresado en milisegundos.

```
procedure MovArtic(destino:Tarray);
```

Movimiento coordinado articular con cuyo destino es un vector de coordenadas articulares que se le pasa como parámetro.

```
procedure MovCoor(destino:TTransformacion);
```

Movimiento coordinado articular cuyo destino es una transformación.

```
procedure MovLin(destino:TTransformacion);
```

Movimiento rectilíneo cuyo destino es una transformación.

```
procedure MovRelCoor(destino:TTransformacion);
```

Movimiento coordinado relativo a la herramienta.

```
procedure Drive(numero:integer;valor:single);
```

Mueve la articulación dada por el primer parámetro un ángulo/distancia dada por el segundo parámetro.

```
procedure Home;
```

Vuelve a la posición de reposo del robot.

Las siguientes instrucciones serán 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.

```
procedure Tool(T:TTransformacion);
```

Cambia la referencia asociada a la herramienta, y por tanto, la referencia asociada al extremos del robot.

```
procedure Abrir;
```

Abre la herramienta hasta la apertura máxima.

```
procedure cerrar(cuanto:single);
```

Cierra la herramienta hasta la apertura especificada por el parámetro (en mm).

```
procedure wait(i:integer);
```

Detiene la ejecución del programa hasta que se active el canal referenciado.

```
procedure signal(i:integer);
```

Activa o desactiva el canal referenciado dependiendo de su signo. Si el signo es positivo se activará el canal, mientras que si es negativo se desactivará.

```
function sig(i:integer):Boolean;
```

Lee el estado del canal de entrada correspondiente.

