

Volumen

I

DESIGNER 3.0: EL NUEVO ENTORNO DE GALILEO

Las novedades del entorno de desarrollo integrado

Guía de usuario de Designer 3.0

ENTORNO DE DESARROLLO PARA GALILEO 3.0

Guía de usuario de Designer 3.0



© ThyssenKrupp Ingeniería y Sistemas
Parque Científico Tecnológico, Edificio I+D • Gijón - Asturias
Teléfono (34) 985 095768 • Fax (34) 985 09 9220

DESIGNER 3.0

Tabla de contenidos

El nuevo sistema de desarrollo	3	Depuración del proyecto	30
Introducción	3	Plugins y Formularios	32
Cómo utilizar este manual	4	Propiedades de un Computador	32
Referencias	4	Solapa "General"	33
Datos sobre este documento...	5	Solapa "Máquinas"	34
Revisiones	5	Propiedades de una tarjeta	34
Instalando el Entorno de Desarrollo	6	Solapa "Configuración"	35
Requerimientos de Instalación.	6	Solapa "Configurar Bus"	36
Proceso de instalación	7	Edición de Bus	36
Trabajando con la aplicación: Guía rápida	13	Compilación	38
Menú Principal	14	Compilación de un secuenciador	39
Menú "Fichero"	14	Compilación del proyecto	39
Menú "Edición"	15	Usuarios del proyecto	41
Menú "Insertar"	16	Edición de Secuenciadores	43
Menú "Modificar"	16	Edición de Código	44
Menú "Compilar"	16	Mezcla de Proyectos	47
Menú "Ventana"	17	Mezclar métodos globales	48
Menú "Comunicaciones"	17	Mezclar Secuenciadores	49
Menú "Herramientas"	18	Importar Visualizaciones	50
Menú "Ayuda"	19	Importar código Libre	51
Los botones de acceso rápido	20	Importar Plantillas de Depuración	51
El Proyecto	22	Importar Computadoras	52
Creación de un nuevo proyecto	22	Importar Esclavos de Bus	53
Creación de una nueva computadora	25	Importar Máquinas y Zonas	53
Creación de una Tarjeta	26	Importar la configuración del Agente de	
Editar el Bus	26	Transportes	55
Crear los Secuenciadores	26	Secuenciadores	57
Crear el Simbólico	27	Modos de funcionamiento	57
Realizar las Visualizaciones	28	Elementos de un grafcet.	57
Configurar el Agente de Transportes	28	Herencia	62
Compilar el proyecto	29	Interfaz de diseño	63
Crear y dar permisos a usuarios de proyecto	29	Edición de Simbólico	76
	29	Nodos del árbol del simbólico	81

DESIGNER 3.0

El mapa de variables en la tarjeta	84	Tratamiento de Históricos	151
Campos de Formularios de Edición de variables.		Comparación de Secuenciadores	153
	86	Comprobación del proyecto.	154
Machines	86	Opciones del Entorno	156
Zonas	87	Problemas Típicos	158
Variables PLC	87		
Variables Shared	88		
Strings	89		
Componentes	90		
Constantes	90		
Configuración del Agente de Transportes	92		
Paquete PATHFINDER	92		
Estaciones, Ordenes, Trayectorias y Defectos.			
	92		
Configuración del Agente de Transportes	93		
Solapa "Estaciones"	94		
Solapa "Trayectorias"	95		
Solapa "Procesar Ordenes"	95		
Solapa "Averías"	96		
Solapa "Código Generado"	97		
Visualizaciones	99		
Comunicaciones	118		
Ver red Galileo	118		
Ver averías	121		
El Pupitre Virtual	122		
Depuración	124		
Estado de variables	124		
Grafcet en línea	128		
El Editor de Tracking	129		
Las solapas ocultas	135		
Repositorio de Secuenciadores	139		
El Repositorio Remoto	139		
Herramientas	142		
Generación de Informes de proyecto	142		
Solapa "Generación"	143		
Solapa "Visualización"	144		
Edición de mapas	144		
Interfaz de objeto nativo	148		
Conversión de proyectos 2.x a 3.0	150		
Registrar el tipo de archivo .TKR	150		
El Galileo Status Monitor	150		

DESIGNER 3.0

El nuevo sistema de desarrollo

Las versiones 1.x ó 2.x del Sistema de Control Galileo dejan paso a la nueva versión 3.0 del mismo, que supone un cambio radical en alguno de sus conceptos y filosofía. De igual manera, el Entorno de Desarrollo Integrado (DESIGNER), cambia para poder ofrecer acceso a toda esta nueva funcionalidad.

Este manual no es sólo una revisión de las versiones 1.x y 2.x de la herramienta de desarrollo, sino que ha sido reescrito completamente a fin de poder dar cabida y explicación a los nuevos conceptos que se han desarrollado para la versión 3.0 de Galileo, y que suponen un cambio radical en cuanto a modo de trabajo con las versiones previas. A lo largo de este documento encontrará no solo la descripción formal de la nueva herramienta, sino también información adicional sobre la nueva filosofía, ejemplos ilustrados, y comparativas con la vieja versión que le ayudarán a comprender y dominar la actual en un corto espacio de tiempo.

Introducción

El entorno de desarrollo y programación de instalaciones (también conocido como “Designer”) esta diseñado para ser el “Front-end” de usuario en todo el sistema de programación “Galileo”. Su funcionalidad puede aplicarse desde los primeros pasos de definición de la instalación (montaje eléctrico: definición del Bus) y seguir hasta la fase de seguimiento de producción una vez finalizada la puesta en marcha de la instalación (a través de la herramienta Status Monitor).

Como características principales de esta herramienta, podemos mencionar:

- Posibilidad de trabajar con varios proyectos a la vez.
- Se maneja desde un entorno Visual (Windows 9x, ME, NT 4.0, 2000, XP).
- Posibilidad de importar / exportar información desde / hacia otras herramientas de programación.

Permite el control total del sistema desde ella, sin tener que desplazarse a otros puntos de la instalación, siempre que estos se encuentren en una red accesible vía TCP/IP.

Cómo utilizar este manual

TIPO DE TEXTO	
Código	Este tipo representa código o texto que aparece en la pantalla.
Teclas	Con este tipo se indican teclas del teclado que el usuario puede pulsar para realizar ciertas acciones.
MENU	Con este tipo se indican opciones del menú principal de la aplicación.

A fin de que resulte más legible, este manual usa unas convenciones de tipo y de texto para resaltar ciertos elementos, tal y como se observa en la tabla de tipos de texto. Asimismo, se hace uso intensivo de imágenes para resaltar ciertas partes de información.

SDO. DE LOS ICONOS	
	Información importante
	Nota adicional
	Ejercicio para el lector
	Comparativa con versión anterior
	Opciones ocultas

A fin de ayudar en la tarea de ilustrar los conceptos mediante ejemplos y comparativas, se han dispuesto una serie de iconos que aparecerán a lo largo del texto y que indicaran ejercicios propuestos, notas adicionales, resúmenes e información importante para el lector.



Las imágenes

A lo largo de este manual, una serie de imágenes aparecen para situar al lector en la pantalla, formulario o plugin que se está comentando. En casi todas las ocasiones las imágenes corresponderán a formularios de la aplicación Designer, y casi siempre irán precedidas de la combinación de teclas o menús que es necesario utilizar para acceder a las mismas. Se puede encontrar una lista de todas las ilustraciones contenidas en este manual en la tabla de ilustraciones que aparece en los apéndices finales del mismo.

Las ilustraciones de formularios u objetos que no pertenezcan a la aplicación Designer serán indicadas puntualmente.

Referencias

Dado que este manual no puede contener toda la información necesaria para trabajar con el sistema Galileo, se indican a continuación una serie de manuales y enlaces a otros puntos de información donde será posible conseguir una información más detallada sobre algunos aspectos del sistema de desarrollo, que no serán por lo tanto explicados aquí. Puede usar estos enlaces como referencias para completar su formación.

 Para más información, consulte regularmente la página web del departamento de I+D: <http://www.tkmanutencion.com>

- “Guía de programación del Sistema Galileo”
- “Documentación sobre los componentes nativos para XANA”
- “Guía de instalación de la tarjeta de control Hilscher”

- Cualquier guía / tutorial sobre el lenguaje Visual JavaScript de Microsoft.
- Cualquier guía / tutorial sobre *Cascading Style Sheets* (hojas de estilo) presente en Internet.
- Guía de instalación del cliente de Oracle 8, 9 o 10.

Datos sobre este documento...

Autor	Manuel J. Fernández
Aprobado	Jose Luis Santiago
Revisiones	Bárbara Adaro
Fecha	02/02/2005
Versión	3.0
Proyecto	Galileo
Nº de Proyecto	174/ 9A00001
Cliente	Interno
Tipo Documento.	Confidencial



Revisiones

A continuación se detallan las diferentes revisiones que ha sufrido este texto desde sus inicios en las versiones 2.x hasta la actualidad.

FECHA	VERSIÓN	DESCRIPCIÓN	AUTOR
16/04/2001	0.0	Creación de esta guía.	MJK
20/04/2001	0.1	Primera versión (incompleta aún) entregada	MJK
18/05/2001	0.2	Primera versión completa entregada para revisión.	MJK
24/05/2001	1.0	Primera versión revisada y entregada a los usuarios	MJK
19/07/2001	1.1	Versión actualizada para reflejar cambios de la versión 1.7	MJK
18/10/2001	1.2	Versión actualizada con las nuevas características de la versión 2.0	MJK
24/01/2002	1.3	Actualizada para las nuevas características de la versión 2.1 del Designer.	MJK
19/04/2002	2.3	Actualización para reflejar los cambios en la versión 2.3 del Designer.	MJK
25/04/2002	2.4	Actualización para reflejar los cambios en la versión 2.4 del Designer.	MJK
26/04/2004	3.0	Recreación completa con las características de la versión 3.0 del Designer.	MJK

DESIGNER 3.0

Instalando el Entorno de Desarrollo

La tarea inicial para poder empezar a usar el sistema es la instalación. Esta tarea, a veces tan sencilla, requiere a veces de tomar algunas decisiones que pueden no serlo.

Es recomendable instalar siempre la versión más reciente del entorno. Dicha versión está disponible en la web <http://www.tkmanutencion.com>. Una vez obtenida, se procederá a instalarla en la computadora de trabajo apropiada. A continuación mostraremos los diversos diálogos que aparecerán durante este proceso y las opciones que podemos elegir en cada caso concreto.

Requerimientos de Instalación.

- Un ordenador con Windows 9x, ME, NT4.0, Windows XP o Windows 2000 versión Servidor o profesional instalada.
- El navegador Internet Explorer 5.0 (o una versión superior) debe estar instalado en el sistema.
- Si se desea conectar con las estaciones remotas (modo online), es necesario que el ordenador disponga de acceso vía TCP/IP a dicha estación.
- Si se desea compilar los programas que usen llamadas a componentes nativos en el código XANA, se deberá instalar también el sistema Galileo, a fin de que las librería de componentes nativos estén presentes en el sistema.
- Si se pretende configurar el agente de transporte, es necesario un cliente de Oracle (versión 8 o superior) instalado en el ordenador donde se instala el entorno de desarrollo y configurar una conexión con la base de datos de control.

Tanto esta aplicación como su hermana “Galileo Status Monitor”, requieren una licencia para su funcionamiento. Dicha licencia es gestionada mediante un programa llamado “Gestor de Licencias” y que debe ser instalado de forma independiente desde el setup “LicenceManager.exe”. Una vez instalado, se debe contactar con el servicio de soporte para obtener las licencias apropiadas

Para Obtener licencias, Lo más sencillo es llamar al teléfono del servicio de soporte o bien enviar un correo a la dirección galileo@tkmanutencion.com indicando la aplicación para la que se requiere la licencia y el número de referencia.

Proceso de instalación

Tras las pantallas de bienvenida, aparece el diálogo de Licencia, que debe ser aceptado por el usuario antes de continuar.

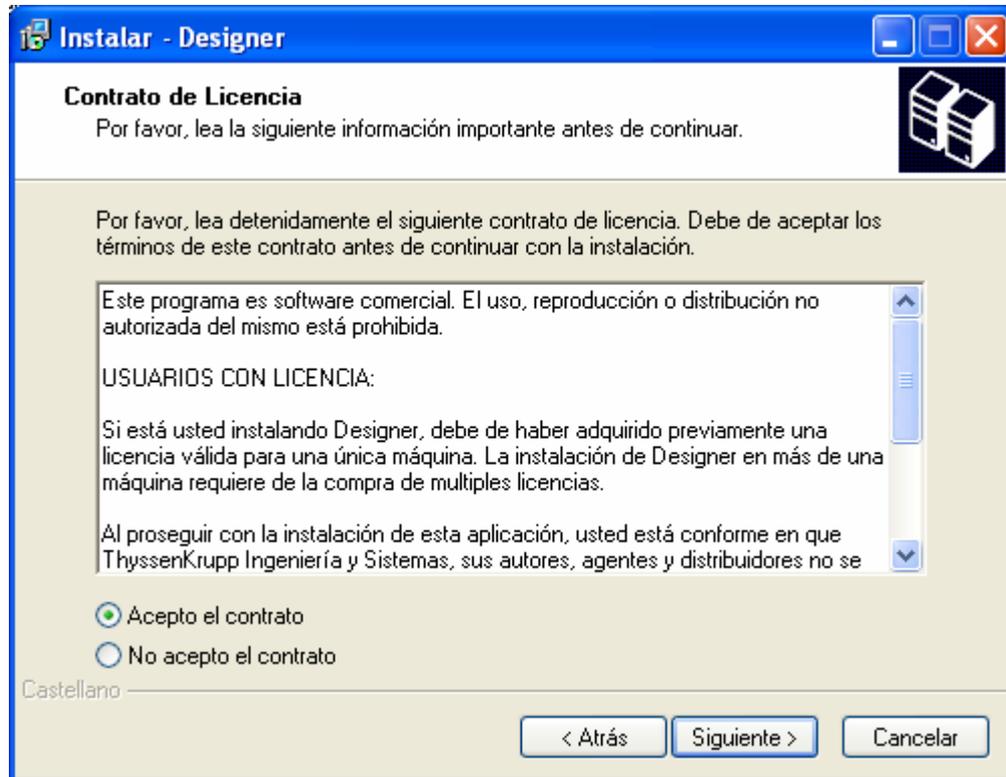


Ilustración 1: Diálogo de aceptación de licencia

A continuación, otro diálogo nos mostrará de información de última hora sobre la aplicación. Se recomienda leerlo, dado que siempre incluirá información útil sobre la aplicación que no estará incluida en este manual. Tras leerlo, debemos pulsar sobre el botón “Siguiete”.

DESIGNER 3.0

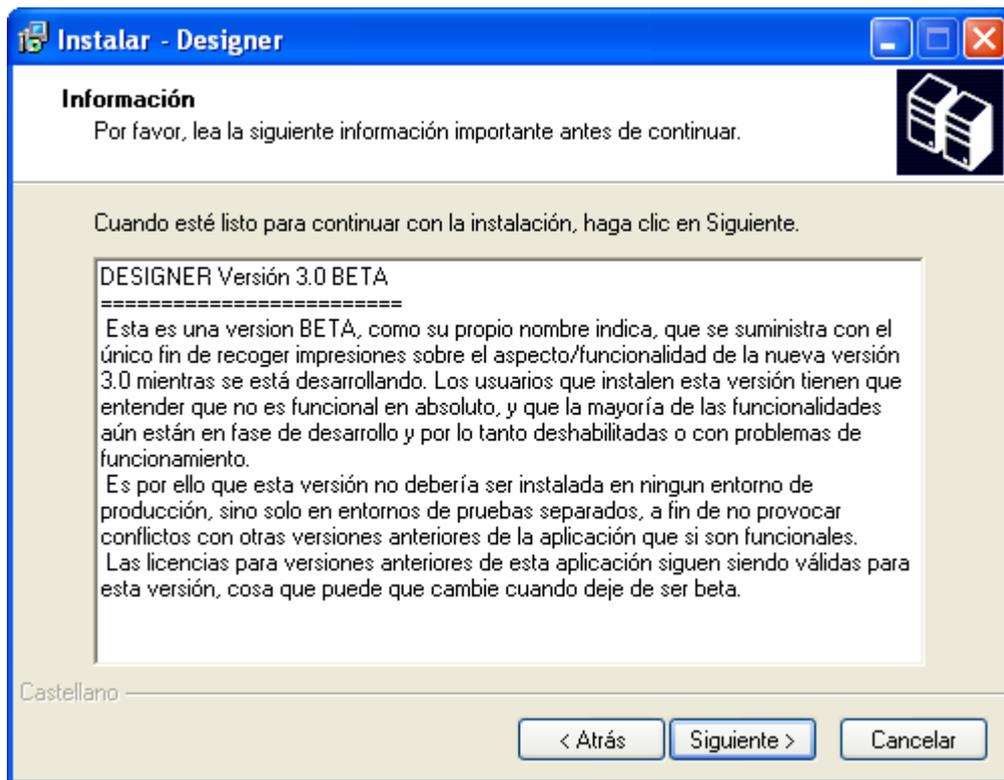


Ilustración 2: Diálogo "readme".

Se nos pide ahora que elijamos un directorio de nuestro disco donde instalar la aplicación. Podemos seleccionar el directorio por defecto que se nos muestra (c:\Designer) o escoger otro del disco duro. Sugerimos que si se cambia, se use un directorio de nombre corto y sin espacios en blanco.

DESIGNER 3.0

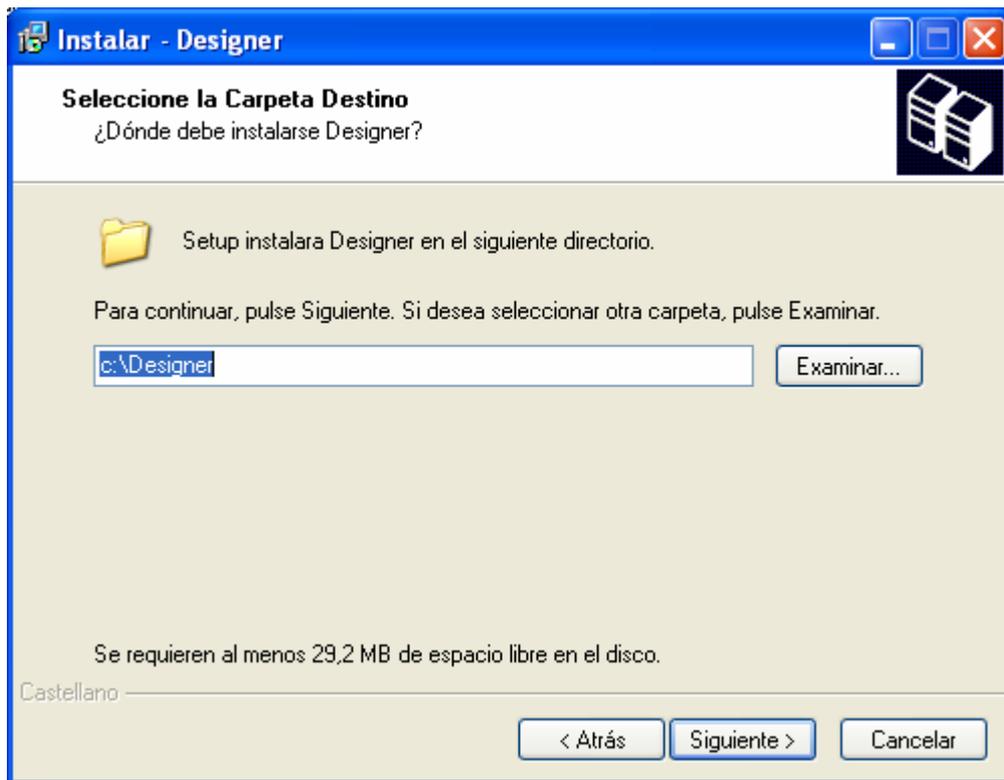


Ilustración 3: Diálogo solicitando un directorio donde instalar la aplicación.

La aplicación de instalación permite seleccionar entre varios tipos de instalación: Completa (incluye todos los componentes y aplicaciones extra), Mínima (solo se instalarán los componentes indispensables para funcionar y ninguna aplicación extra) o Personalizada (donde podemos elegir los componentes extras que queremos que se instalen). Podemos elegir entre estos tipos de instalación en el diálogo siguiente.

DESIGNER 3.0

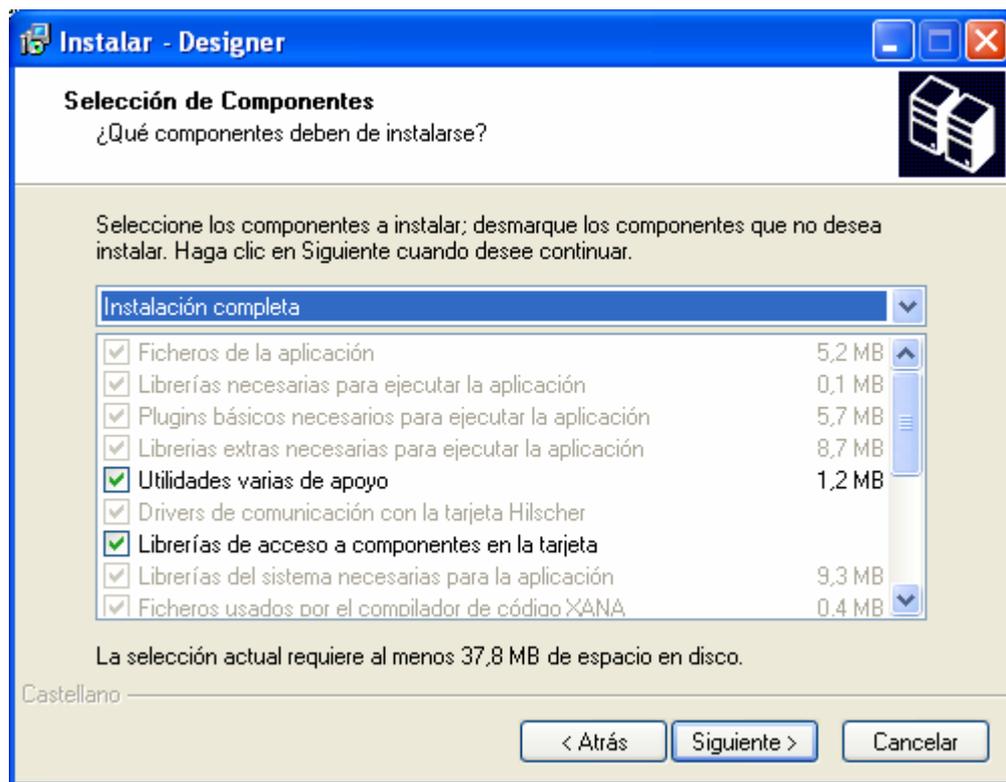


Ilustración 4: Selección de componentes a instalar.

El grupo de programas,
 En la versión 2, el instalador preguntaba por un grupo de programas donde instalar los accesos directos a los programas. En esta versión se instalan por defecto en el grupo de programas "Designer 3.0" del menú de inicio.

El siguiente diálogo nos permite decidir si queremos colocar accesos directos a la aplicación principal en el escritorio y en la barra de tareas, mediante la selección en un cuadro de chequeo.

DESIGNER 3.0

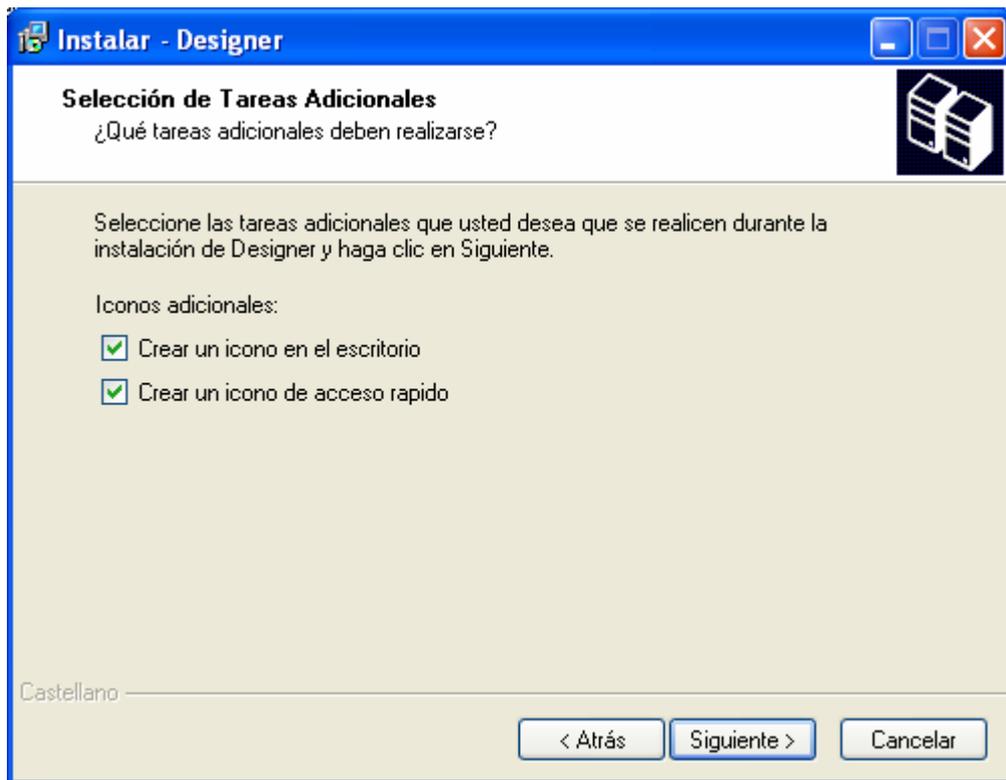


Ilustración 5: Diálogo de tareas adicionales.

Tras aceptar este diálogo, comenzará la instalación del entorno. Mientras se produce la copia de archivos, aparecerá el siguiente diálogo, que le permitirá cancelar la instalación antes de que acabe si es necesario.

DESIGNER 3.0

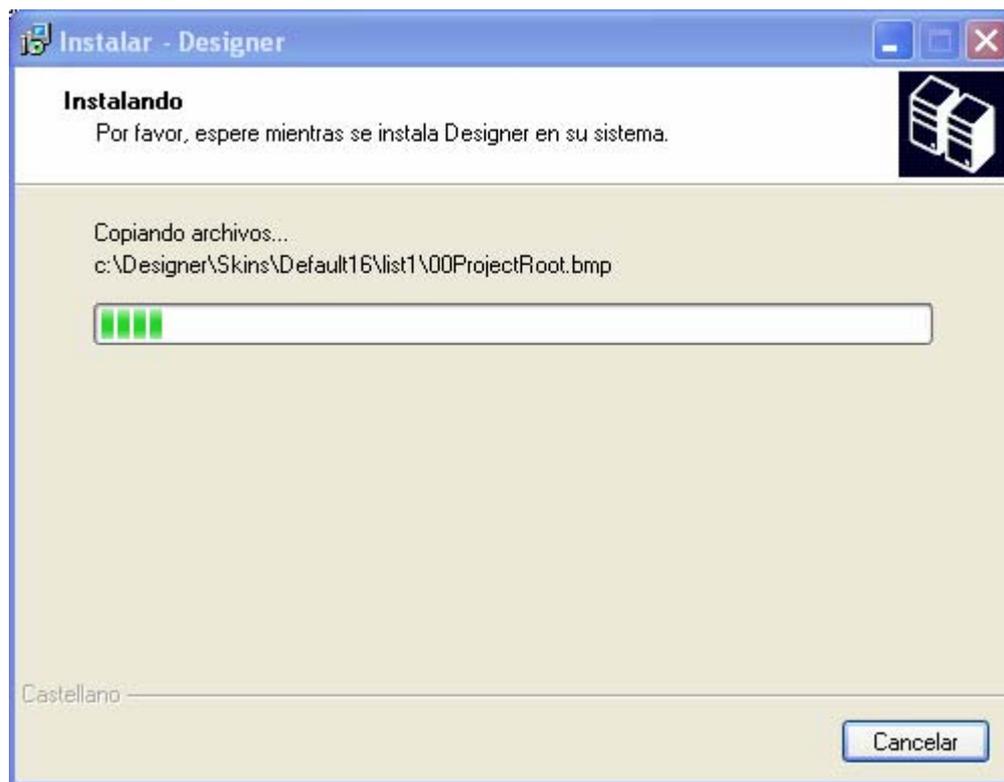


Ilustración 6: Diálogo con información de instalación.

El driver de comunicaciones, Ya no aparece como un elemento de la instalación, y no es necesario que sea registrado, como ocurría en las versiones 2.x, por lo tanto esta opción ya no aparece

Posteriormente un par de diálogos nos informarán de las últimas novedades que contiene la versión que acabamos de instalar y de los pasos finales de la instalación.

Tras esto, ya tendremos la aplicación instalada y lista para ejecutarse, pero aún no podremos usarla, dado que no tenemos instalada la licencia para la misma. Para obtener una licencia, ejecute la aplicación una vez, recibirá un mensaje de error y la ejecución se abortará. A continuación se debe ejecutar la aplicación “Gestor de Licencias”, que debe ser instalada de forma independiente) y comprobar que existe una entrada para DESIGNER III en la lista de códigos de producto. Al seleccionarla, en la parte inferior aparecerá una clave. Debe enviar dicha clave al departamento apropiado de ThyssenKrupp Ingeniería y Sistemas S.A. a fin de obtener otra clave que licenciará el producto, y que deberá introducir en este mismo formulario. Puede contactar por correo electrónico en la dirección: galileo@tkmanutencion.com o poniéndose en contacto por vía telefónica con algún integrante del departamento de soporte. En función del número de aplicaciones que necesite licenciarse, requerirá más o menos códigos a introducir. Tras esto, la aplicación debería funcionar sin ningún problema.

Nota

Los códigos de licencia están fuertemente asociados al hardware que usa el computador donde se instala la aplicación. Esto implica que un cambio de dicho hardware (cambio de tarjeta de red, formateo del disco duro, etc.) puede llegar a provocar que la licencia no sea reconocida como válida. En dicho caso, habrá que solicitar una nueva licencia.

Trabajando con la aplicación: Guía rápida

Si tu intención es describir la verdad, hazlo con sencillez y la elegancia déjasela al sastre..

Albert Einstein

A l ejecutar el entorno de desarrollo (programa “Designer”) de la versión 3.0 por primera vez, su pantalla principal se nos muestra tal y como se puede observar en la siguiente ilustración:

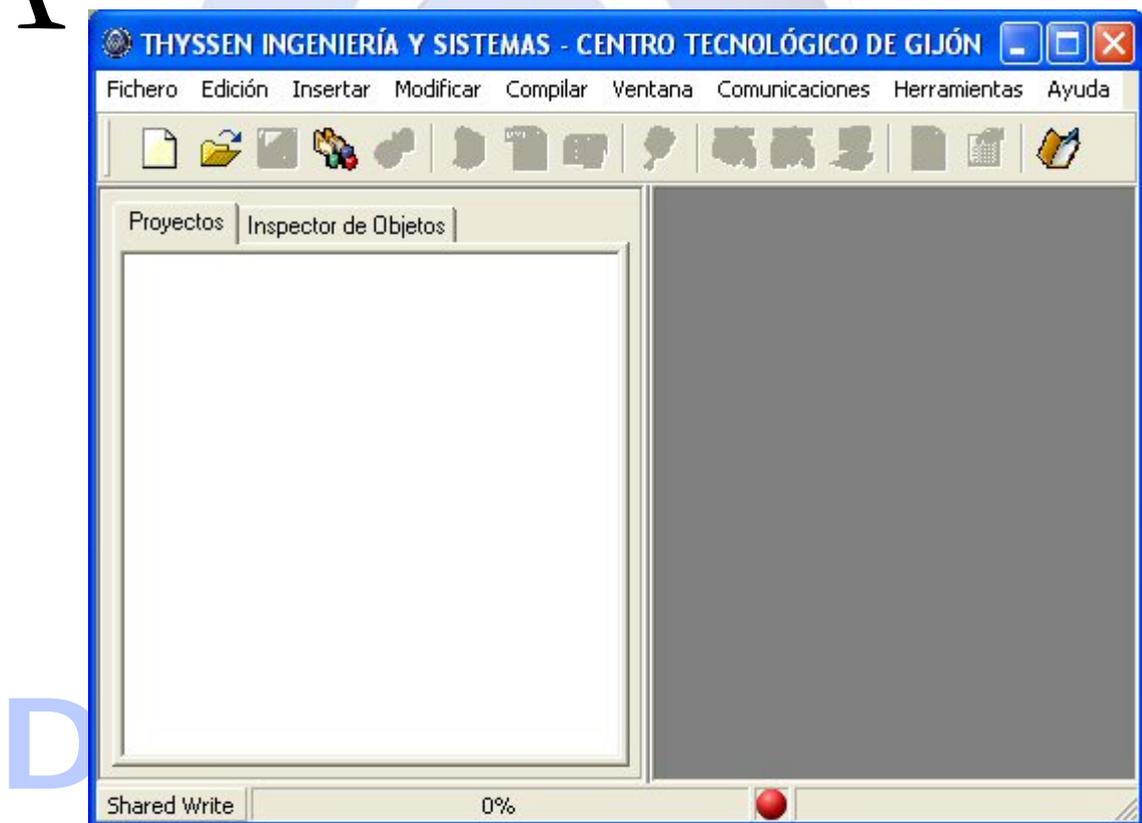


Ilustración 7: Pantalla principal de la aplicación.

Este aspecto es configurable mediante el uso de Skins, como veremos en el apartado correspondiente, pero la disposición de los elementos será siempre la misma: en la parte izquierda aparece el árbol de proyectos, donde se irán colocando los diversos elementos que vayamos creando / cargando, de

manera que al seleccionarlos con el ratón aparezcan en la parte derecha los formularios / detalles sobre dichos elementos. En la parte inferior, una barra de estado nos informa del estado del proyecto seleccionado. A izquierda de la misma, se nos muestra la ocupación de memoria Shared (read o write, pudiendo cambiar la vista si pulsamos sobre el primer panel) del proyecto. A continuación, un led informa del estado de las comunicaciones: si se encuentra en modo “Online”, el led se mostrará de color verde, y permanecerá de color rojo mientras se encuentre en modo “Offline”. La última parte de la barra de estado muestra un reloj con la hora actual del sistema.



A continuación revisaremos las opciones del menú principal y daremos una breve indicación sobre las mismas.

Menú Principal

Menú “Fichero”

Desde este menú se puede acceder a las siguientes opciones:

Nuevo Proyecto

Esta opción permite la creación de un nuevo proyecto, requiriendo para ello la introducción de los datos básicos del mismo (véase el apartado [Creación de un nuevo proyecto](#)).

Abrir Proyecto

Esta opción permite rescatar información sobre un proyecto, previamente salvada a disco desde esta aplicación, y mostrarla en el árbol de proyectos.

Acceso al Repositorio

Permite abrir el formulario de acceso al repositorio, desde el que se puede conectar con la base de datos que provee un repositorio de secuenciadores listos para usar, o bien abrirlos para su edición. Véase el apartado: Uso del repositorio.

Propiedades

Mediante esta opción se puede volver a editar la información básica definida para el proyecto. Véase el apartado [Creación de un nuevo proyecto](#).

Eliminar

La opción de Eliminar sirve para eliminar la información sobre un determinado proyecto del árbol de proyectos. Al usarla, se eliminará el proyecto actualmente seleccionado en el árbol. Si dicho proyecto tuviera cambios aún no salvados, se preguntará al usuario si desea salvar a disco dichos cambios o desecharlos. La eliminación del proyecto no implica el borrado del fichero donde se guarda.

Salvar

Salva los datos del proyecto activo a disco. Si es la primera vez que se salva el proyecto, se solicitará un nombre de fichero donde guardarlo, en caso contrario, se salva directamente al fichero de proyecto. El salvado de un proyecto provoca que se genere una copia de seguridad de su archivo, que tendrá su mismo nombre, pero con extensión

📖 La extensión de los ficheros de proyecto, Mientras que en las versiones anteriores los ficheros de proyecto tenían la extensión .THY, para la versión 3.0 la extensión por defecto de los ficheros pasa a ser .TKR

.BAK0. Se guardarán las 10 últimas modificaciones del proyecto, numeradas desde 0 hasta 9 (.BAK0, .BAK1,..., .BAK9) con lo que la más reciente copia será la número 0 y la más antigua la 9.

Salvar Como...

Idéntica a la opción [Salvar Proyecto](#), la diferencia radica en que siempre se pregunta el nombre de fichero donde salvar el proyecto.

Informe

Permite generar un informe en donde se puede incluir toda o parte de la información que contiene el proyecto. Véase el apartado [Generación de Informes de proyecto](#).

Mezclar Proyectos

Esta opción permite introducir en el proyecto elementos de otro proyecto, ya sean secuenciadores, variables, configuraciones varias, o cualquier otro elemento presente en otro proyecto. Es una manera rápida de sincronizar el trabajo entre varias personas.

Reopen

Permite la re-apertura rápida de los 10 últimos ficheros con los que se ha trabajado, sin necesidad de navegar en el árbol de directorios para seleccionarlos. Esta opción no aparece las primeras veces que se ejecuta la aplicación, por el simple hecho de que aún no se ha trabajado con ningún proyecto. Una vez que aparece, también aparecen 2 sub-opciones:

 **Abrir ficheros recientes,**
Esta opción va asociada al usuario Windows que trabaja con la aplicación en cada momento, es decir, que si varios usuarios trabajan con la aplicación, a cada uno le aparecerá en la lista los últimos elementos sobre los que ha trabajado (independientemente unos de otros).

Limpiar lista de proyectos recientes: Limpia la lista de proyectos recientes, dejándola vacía (como si fuese la primera vez que trabajamos con la aplicación).

Borrar proyectos obsoletos: Borra las entradas de la lista de proyectos cuyo fichero no exista ya en disco (o no se encuentre en la localización marcada).

Salir

Mediante esta opción saldremos de la aplicación. Si alguno de los proyectos cargados tuviese cambios sin salvar, se preguntará al usuario si desea salvarlos antes de salir.

La configuración del Agente de Transportes

En las versiones 2.x del Designer, la configuración del agente de transportes se hacía desde un menú de esta sección. En esta nueva versión, la configuración es directamente accesible desde el árbol de proyectos y desde otro menú, tal y como se verá más adelante.

Menú "Edición"

Este menú contiene opciones que se aplican al trabajo sobre el proyecto. En función de la parte del mismo que sobre la que se esté trabajando, estarán activados o no. En el caso de los plugins, algunos habilitarán submenús en este menú, lo que esto implique será indicado en el apartado correspondiente a cada plugin.

Buscar en Proyecto

Permite acceso al plugin de búsqueda de cadenas, el cual habilita la búsqueda de cadenas en los proyectos, pudiendo localizar partes del mismo con más facilidad.

Menú "Insertar"

A través de este menú accedemos a las opciones de inserción de nuevos elementos:

Nueva Visualización

Permite insertar una nueva visualización en el proyecto. Para más información véase el apartado [Realizar las Visualizaciones](#). Para poder usar esta opción es necesario tener un proyecto activo.

Nueva Computadora

Permite insertar una nueva computadora en el proyecto activo. Para más información véase el apartado [Propiedades de un Computador](#). Para poder usar esta opción es necesario tener un proyecto activo (seleccionado en el árbol de proyectos).

Nuevo Secuenciador

Permite insertar un nuevo secuenciador en un proyecto. Para más información véase el capítulo 6. Para poder usar a esta opción es necesario tener un proyecto seleccionado.

Menú "Modificar"

Este menú permite modificar aspectos concretos de un proyecto:

Generar configuración del Agente de Transportes.

Permite el acceso al plugin de edición de la configuración del Agente de transportes para un proyecto concreto.

Editar código

Provoca la aparición del plugin de edición de código (funciones) del proyecto. Para más información véase [Edición de Código](#).

Edición de Bus

Provoca la aparición del plugin de edición de bus para el proyecto. Previamente es necesario haber creado al menos un computador con una tarjeta, o no podremos hacer nada con la edición. Para más información véase [Edición del Bus](#).

Menú "Compilar"

A través de este menú se pueden realizar las tareas de compilación del código del proyecto.

Generar y Compilar

Este menú permite acceso al plugin de compilación, desde el cual es posible generar el código de control con los datos actuales y posteriormente, si la generación fue bien, compilarlo, mostrando los posibles errores o avisos generados por el compilador durante la compilación. Para poder usar esta opción, es necesario tener un proyecto abierto.

 Las otras opciones de compilación

En las versiones 2.x del Designer, era posible compilar y transferir el código directamente al sistema de control desde este menú. Para simplificar y evitar la confusión, estas opciones ahora se manejan desde el plugin de compilación, que es mucho más completo y permite un mayor control sobre la generación y compilación, como se verá en la sección correspondiente.

Menú "Ventana"

Típico menú de aplicación MDI Windows, a través del cual podemos colocar las ventanas hijas en determinadas posiciones, a gusto del usuario. La única opción que no es estándar de las aplicaciones MDI se detalla a continuación.

Mostrar / ocultar ventana de proyectos

Permite ocultar temporalmente o volver a mostrar el árbol de proyectos principal, a fin de tener un área de trabajo más amplia con la que trabajar. También se logra este efecto mediante la pulsación de la tecla F12.

Menú "Comunicaciones"

 **Computadoras o Tarjetas**, En las versiones anteriores, la comunicación se establecía entre el entorno de desarrollo y la tarjeta de control. A partir de la versión 3.0, se hace entre el entorno de desarrollo y el proyecto, que engloba las computadoras desde donde los diferentes Sistemas de control transmiten la información. Así pues, es transparente para el usuario con que computadora o tarjeta se esta comunicando: la información se obtiene de todos los que se encuentren activos.

Modo Online

Permite poner el proyecto seleccionado actualmente en modo "Online", lo que activa la comunicación entre el entorno de desarrollo y el sistema de control "Galileo". Es necesario que el computador desde donde ejecutamos la aplicación "Designer" pueda establecer una comunicación TCP/IP con las computadoras definidas en el proyecto. Véase el apartado [Comunicándose con el sistema Galileo](#).

Modo Offline

Pasa el proyecto seleccionado actualmente a modo offline (previamente dicho proyecto tuvo que ser puesto en modo "Online"). Esto implica que el entorno deja de recibir o enviar información con las computadoras de control definidas en el proyecto. Para más información, véase el apartado [Comunicándose con el sistema Galileo](#).

Averías

La activación de esta opción permite monitorizar las averías que son detectadas por el sistema de control Galileo. Véase el apartado [Seguimiento de averías](#).

Ver Red Galileo

Para acceder a esta opción, es necesario estar en modo "Online". Al usarla, se activará el plugin de control de la red Galileo, que permite monitorizar todas las computadoras que componen el proyecto y enviarles comandos para transferir ficheros, detener el sistema de control, arrancarlo o hacer copia de respaldo entre otras opciones. Para más información véase el apartado [Opciones de control remoto](#).

Status

Esta opción activa el plugin de status de variables en el proyecto. Para poder usarla es necesario tener un proyecto seleccionado y que además se encuentre en modo “Online”. Para más información véase el apartado [Visualizar el estado de las variables \(Status de Variables\)](#).

Mostrar Pupitre

Permite el acceso al Pupitre Virtual. Este es un formulario que actúa como un pupitre de control remoto, permitiendo la modificación de señales de control previamente seleccionadas. Véase el apartado sobre comunicaciones para más información.

Menú “Herramientas”

Mostrar Interfaz

Esta opción activa el plugin de información sobre componentes, que permite al usuario recabar información sobre los métodos que exporta un componente nativo, y que pueden ser usados desde el lenguaje XANA. Para más información véase el apartado [Mostrar la Interfaz de un objeto nativo](#).

Registrar Tipo de Fichero

Permite registrar la extensión de ficheros .TKR como asociada por defecto a la ejecución del entorno de desarrollo, para que sea posible abrir los ficheros de proyectos con un doble click desde el explorador de archivos de Windows.

 **No más importaciones**, En el menú de herramientas, en las versiones 2.x del entorno aparecía la opción “Importar/Exportar secuenciadores” que era la manera de copiar secuenciadores entre proyecto. A partir de la versión 3.0, esta opción deja de existir. Su cometido es realizado ahora por la opción de “Mezcla de proyectos” del menú “Fichero” y el repositorio de secuenciadores.

 **Informes de fallos**, En el menú de herramientas, en las versiones anteriores a la 2.5 del entorno aparecía la opción “Generar Informe de Fallo”, que servían para generar un informe del estado del entorno de desarrollo ante un fallo de la aplicación, a fin de comunicárselo al departamento de I+D. En esta nueva versión la filosofía es algo distinta: Ante la sospecha o evidencia de un fallo en el entorno, se debe proceder a recoger y enviar los ficheros de log que el entorno deja en el subdirectorio “logs” de la instalación al departamento de I+D para su análisis, junto con una explicación del problema y/o una posible manera de reproducirlo.

Instalar Status Monitor

Esta opción permite agilizar la instalación de la aplicación “Status Monitor” como parte final del proceso de diseño. Véase apartado [El Galileo Status Monitor](#)

Informe de averías y transportadores

Esta opción permite ejecutar, directamente desde el entorno, la aplicación “HistReport” para la generación de informes de averías e históricos de transportes que se puede instalar como un complemento extra del Designer. Para más información véase el apartado [Tratamiento de Históricos](#).

Cambiar a Status Monitor

Esta opción, que se encuentra desactivada normalmente, solo puede ser usada desde el Status Monitor, cuando hacemos que pase a modo “Designer” durante un tiempo por el motivo que sea. Al usarla volveremos al modo “Status Monitor” otra vez. Para más información véase el apartado [El Galileo Status Monitor](#).

Editar Mapas

Este menú permite acceder al plugin de edición de mapas del entorno. Como en las versiones anteriores, no es necesario tener ningún proyecto abierto para trabajar con esta herramienta. Para más información de esta opción, consulte el apartado [Edición de mapas](#).

Acceder a tablas Oracle

Este menú permite acceder a otro plugin que nos proporcionara información de las tablas Oracle que usa el sistema Galileo para trabajar y configurarse junto con un SGA. Para obtener más información, véase el apartado “Acceder a tablas de datos”

Opciones de configuración

En esta nueva versión del entorno de desarrollo es posible cambiar algunos aspectos del interfaz a gusto del usuario, para ello se debe usar esta opción del menú. Para más información véase el apartado “Configurando el entorno”.

Comparar Secuenciadores

Con esta opción se puede realizar una comparación a nivel funcional entre dos secuenciadores a fin de discernir los cambios entre uno y otro. Para más información véase el apartado “Comparación de Secuenciadores” del Capítulo 13.

Menú “Ayuda”

Acerca de...

Esta opción muestra una pantalla como la que aparece en la Ilustración 8, con información de los números de versión de la aplicación, así como de las librerías que se están usando. Es conveniente anotar esta información cuando se quiere hacer alguna consulta al servicio de soporte.

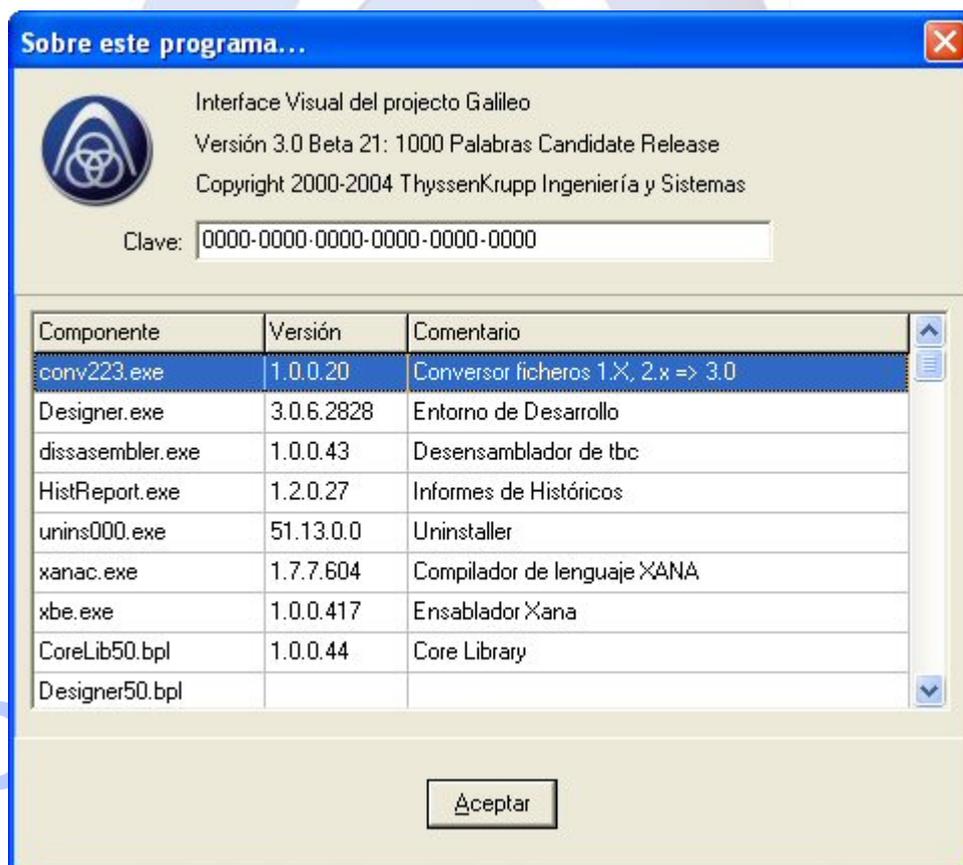


Ilustración 8: Ventana "Acerca de..."

Sugerencias...

Este menú ocasiona la aparición del diálogo que se muestra en la siguiente ilustración (este diálogo también aparece al arrancar el programa) y que ofrece pequeños consejos sobre como utilizar mejor el entorno de desarrollo.

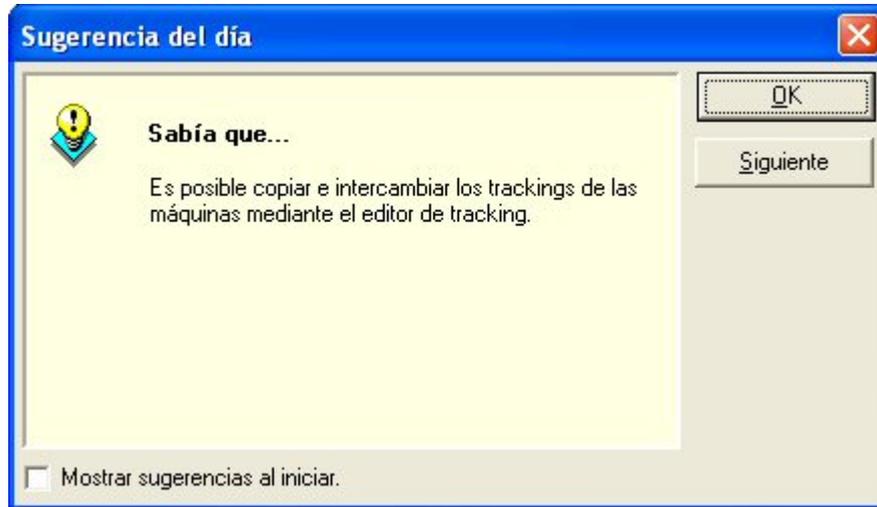


Ilustración 9: Diálogo "Sugerencia del día"

Si se desea activar o desactivar la aparición de este diálogo al arrancar el programa, basta con marcar la casilla “Mostrar sugerencias al arrancar” que aparece en la parte inferior del diálogo.

Los botones de acceso rápido

Debajo del menú principal de la aplicación aparece una barra de herramientas con una serie de botones que permiten un acceso más rápido a ciertas opciones que a través de los menús. A continuación los enumeraremos junto con sus equivalencias.



Ilustración 10: La barra de herramientas principal

Nuevo proyecto. Equivalente a “Fichero → Nuevo proyecto”

Abrir proyecto. Equivale al menú “Fichero → Abrir proyecto”

Salvar proyecto. Equivale al menú “Fichero → Salvar Proyecto”

Repositorio. Equivale al menú “Fichero → Acceso al Repositorio”

Mezclar Proyecto. Equivale al menú “Fichero → Mezcla de proyectos”

Insertar computadora. Equivale al menú “Insertar → Nueva computadora”

Insertar Secuenciador. Equivale al menú “Insertar → Insertar Secuenciador”

Insertar Tarjeta. Equivale al menú “Insertar → Insertar Tarjeta”

 **Botones inactivos.** Al igual que los menús, no todos los botones se encuentran activos desde el principio. Muchos solo se activaran cuando se dan las condiciones necesarias para que su función tenga sentido o sea posible. Por ejemplo, el botón de insertar una tarjeta solo se activara cuando haya una computadora activa en el árbol de proyectos.

Copiar. Equivale al menú “Edición → Copiar”

Cortar. Equivale al menú “Edición → Cortar”

Pegar. Equivale al menú “Edición → Pegar”

Compilar. Equivale al menú “Compilar → Generar y Compilar”

Online. Equivale al menú “Comunicaciones → Modo Online”

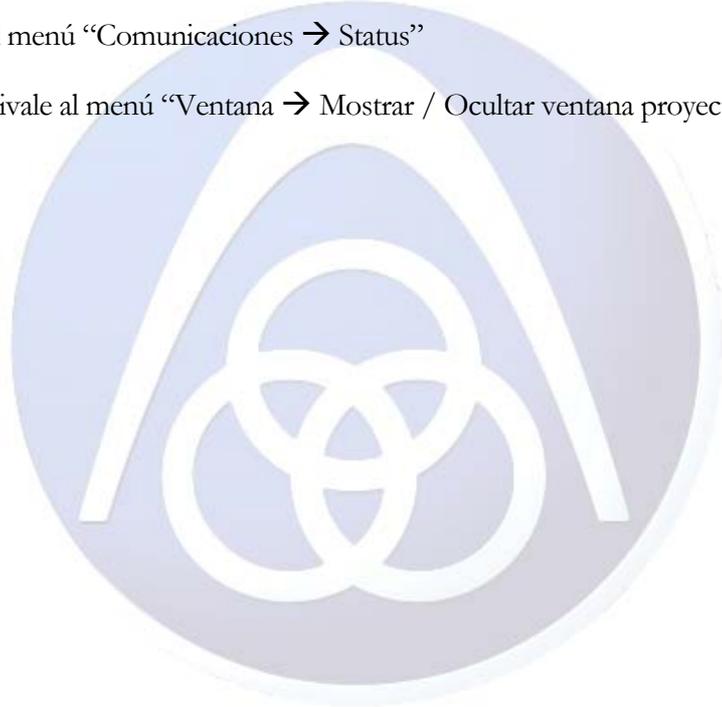
Offline. Equivale al menú “Comunicaciones → Modo Offline”

Red Galileo. Equivale al menú “Comunicaciones → Ver Red Galileo”

Averías. Equivale al menú “Comunicaciones → Ver Averías”

Status. Equivale al menú “Comunicaciones → Status”

Ver/Ocultar. Equivale al menú “Ventana → Mostrar / Ocultar ventana proyectos”



DESIGNER 3.0

El Proyecto

El proyecto es la base fundamental sobre la que se asienta todo el entorno de desarrollo. A partir de esta versión cobra aun más importancia, puesto que es la unidad que cohesionan todos los elementos de la instalación.

Creación de un nuevo proyecto

El primer paso para la creación de un nuevo proyecto es introducir la información básica del mismo. Para ello usaremos la opción **Fichero / Nuevo Proyecto** del menú principal, o pulsando sobre el icono de la barra de herramientas correspondiente. Esto provocará la aparición de la pantalla que vemos en la Ilustración 11.

Nuevo Proyecto...

Introduzca los datos del nuevo proyecto

Título

Nombre

Número

Autor

Empresa

Descripción

Ilustración 11: Información necesaria para la creación de un nuevo proyecto

Los campos que aparecen en este formulario son en su mayoría informativos, y sólo son usados a la hora de generar informes, por lo que no reviste mayor importancia su contenido. La excepción es el campo **Nombre**, único campo obligatorio de este formulario. Debería ser un nombre corto y descriptivo del proyecto, puesto que será el indicador que luego aparecerá en el árbol de proyectos y del que colgarán todos los nodos del proyecto. El resto de campos no están restringidos en longitud o formato.

 El campo "Número", En las versiones anteriores, este campo podía ser editado y modificado por el programador, sin embargo no tenía mayor uso. A partir de esta nueva versión, este campo contiene un identificador único para cada proyecto creado que lo identifica de forma unívoca, y no es editable. Sirve para asociar un "designer" con un proyecto, bloqueando su uso para otros proyectos.

Si posteriormente se desea cambiar el nombre a un proyecto, se puede editar el nombre que aparece en el árbol de proyectos, usando el puntero de ratón sobre la etiqueta del nodo de proyecto. Tenga en cuenta que esto no cambiará el fichero en que se esté salvando el proyecto, ni el resto de los campos de proyecto.

Si el proyecto posteriormente se actualiza para incluir alguna versión del repositorio de secuenciadores, la información de la versión incluida también podrá consultarse desde este formulario.

Objetos por defecto de un Proyecto

Llegados a este punto, la aplicación mostrará un aspecto similar al de la Ilustración 12. Se puede apreciar que los nodos que aparecen dan acceso a una parte concreta de la configuración del proyecto. Concretamente, cada nodo viene representado por un icono:



Señala el nodo raíz del proyecto



Señala el nodo de visualizaciones, de él colgarán todas las visualizaciones que se añadan al proyecto.



Señala una visualización concreta. Cada visualización del proyecto estará en uno de estos nodos.



Señala el nodo de variables. El simbólico completo del proyecto será accesible al pulsar sobre este nodo.



Señala el nodo de secuenciadores. Los secuenciadores que se creen dentro del proyecto aparecerán como hijos de este nodo.



Señala un secuenciador en concreto. Cada secuenciador presente en el proyecto tendrá este icono para distinguirlo del resto de nodos.



Señala el nodo que contiene la configuración del Agente de Transportes para el proyecto. Al pulsar sobre este nodo, se accederá al plugin de edición de la misma.



Señala el nodo de computadoras. Todas las computadoras que formen la red de control Galileo aparecerán como nodos hijos de este nodo.



Señala el nodo de configuración de Bus. Al pulsar sobre este nodo aparecerá el editor de dicha configuración.



Señala el nodo de depuración. Al pulsar sobre el mismo, aparecerá el plugin de estado de variables, que permitirá consultar o modificar el valor de las variables del proyecto. Para que este plugin sea funcional, es necesario que previamente el proyecto haya sido puesto en modo Online.



Señala el Nodo de edición de código general del proyecto. Al pulsar sobre él tendremos acceso al editor de código global del proyecto.

 El código Global, Hasta la versión 3, todo el código de un proyecto estaba disponible de forma global, con lo que cualquier secuenciador podía acceder al código de cualquier otro secuenciador. A partir de la versión 3, la filosofía cambia, y cada secuenciador contiene su propio código, de forma que los cambios en uno no afecten al resto (excepto en el caso de la herencia). Ello implica que el código global ahora debería ser inexistente o mínimo.



Señala el nodo de usuarios. A través de él se tiene acceso al nodo de edición de usuarios del proyecto, que permite crear entradas de usuarios que podrán trabajar con dicho proyecto.



Si posteriormente incluimos secuenciadores desde el repositorio, estos vendrán señalados por este nodo, para indicar que son de solo lectura y no pueden ser modificados.

Ninguno de estos nodos pueden modificarse ni borrarse, puesto que son propiedades inherentes al proyecto, aunque si se puede hacer con sus descendientes. Sin embargo, su aspecto puede variar, en función de la “Skin” usada por el usuario.

La aplicación mostrará un aspecto similar al de la Ilustración 12.

DESIGNER 3.0

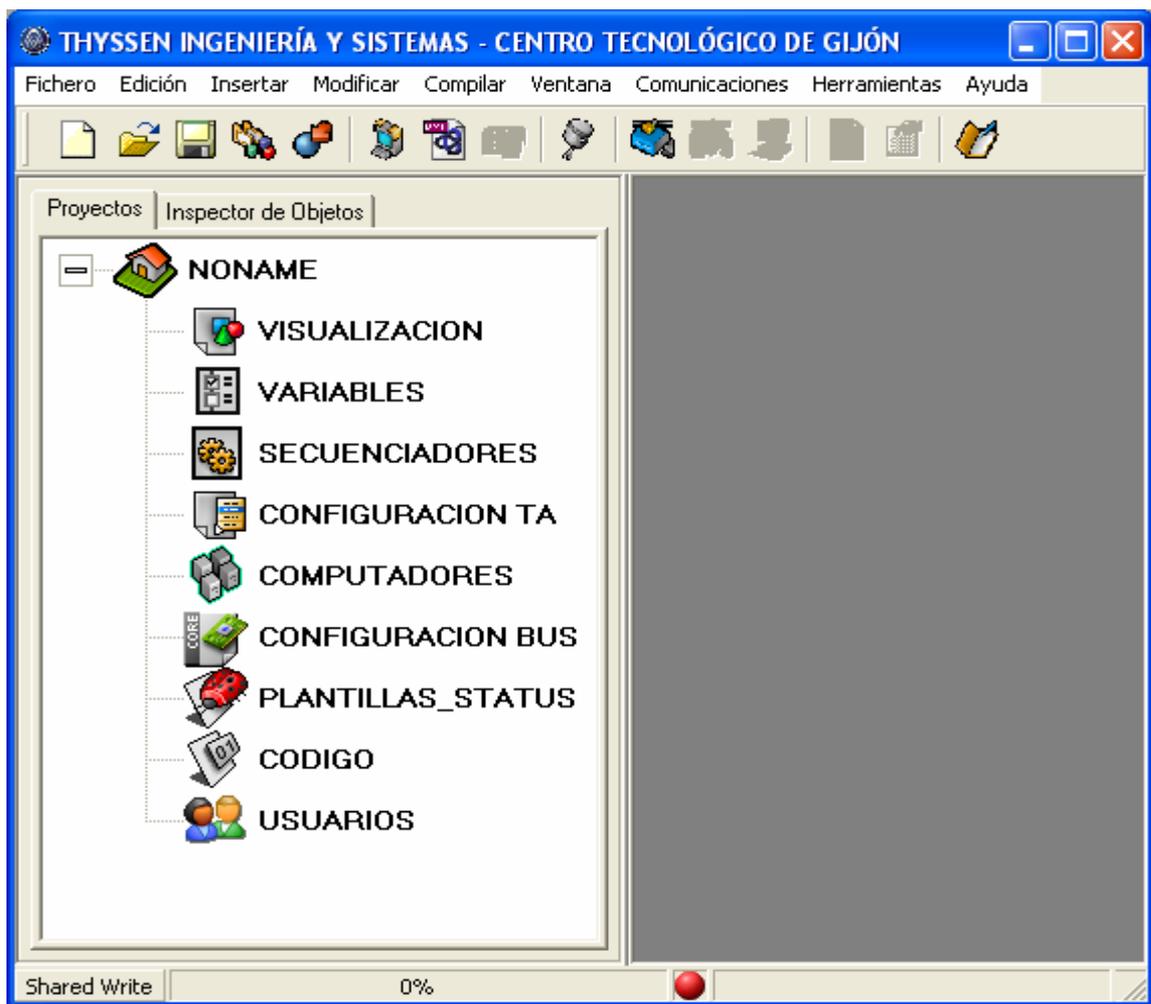


Ilustración 12: Objetos por defecto de un proyecto.

A partir de este momento podemos empezar a usar los plugins de diseño y configuración para crear y configurar los diversos elementos que constituirán y gobernarán la instalación. Vamos a ilustrar el ciclo de trabajo normal con esta versión, enumerando las fases de desarrollo en el orden en que deberían realizarse a partir de este momento.

Creación de una nueva computadora

Tras crear el proyecto, lo siguiente que se debe definir es la computadora o computadoras asociadas a dicho proyecto. Estas computadoras representan a los ordenadores en los que van a ir las tarjetas de control que se pretenden programar / gobernar, por lo que su nombre es un campo clave, ya que el entorno de desarrollo intentará comunicarse con ellas (a través del protocolo TCP/IP) usando esos identificadores. Como en el caso del nombre del proyecto, dichos identificadores pueden cambiarse posteriormente de la misma manera. Al seleccionar la opción **Insertar / nueva Computadora** aparece el cuadro de diálogo que se muestra en la Ilustración 13, solicitando el nombre de la nueva computadora.

Direcciones TCP/IP, A diferencia de la versión 2.x, en el nombre del computador ahora no se debe usar la dirección IP del mismo: aunque parezca que funciona, el uso de la dirección IP no funciona a todos los niveles de comunicaciones y puede resultar en pérdida de comunicación.



Ilustración 13: Diálogo de creación de una nueva computadora

De entre todos los computadores creados, el primero en crearse (o bien otro, si el usuario así lo designa), será el considerado “Computador por defecto”, más adelante explicaremos cual es su papel.

Creación de una Tarjeta

Para crear una nueva tarjeta de control, simplemente basta seleccionar una computadora a la que añadirsele y seleccionar el nodo “Tarjetas” que cuelga de ese nodo. Usando el menú del botón derecho del ratón, escoger la opción “Nueva tarjeta”. A continuación, el diálogo que se muestra en la Ilustración 14 aparecerá, solicitando el nombre de nueva tarjeta. Es aconsejable, por claridad, que dicho nombre no se repita en ninguna otra tarjeta del proyecto. Se pueden crear hasta un máximo de 4 tarjetas por computadora. El número de tarjeta se asigna de acuerdo al orden de creación de las tarjetas en la computadora (la primera creada será la 0, la segunda la 1, la tercera la 2 y la cuarta la 3). Como en los anteriores casos, el nombre de la tarjeta se podrá cambiar posteriormente editando el nodo correspondiente del árbol de proyectos.



Ilustración 14: Diálogo de creación de una nueva tarjeta

Editar el Bus

Tras tener claro cuantas computadoras y tarjetas necesitamos para acometer el proyecto, es necesario ahora introducir la información relativa a los módulos de bus que van a ser usados en la instalación, así como la repartición que vamos a hacer de los mismos en las tarjetas. La manera más sensata de realizar esta parte es guiándose por los esquemas eléctricos de la instalación, que nos darán la información precisa de dichos módulos, el tamaño de los mismos en entradas/salidas y su posición relativa en el bus desde el que se vayan a controlar. La forma de editar esta información la veremos en el apartado [Edición de Bus](#).

Crear los Secuenciadores

El siguiente paso, (que se puede realizar también antes que la edición del bus) es la definición de las estructuras de comportamiento de los transportadores que vayan a implantar en la instalación, también

conocidas como SECUENCIADORES. La edición de secuenciadores se deberá hacer siempre desde un punto de vista lo más ajeno posible a la instalación, dado que siempre deben de ser considerados como definiciones genéricas de los elementos que se pretenden usar, es decir, que idealmente deberían poder ser usados independientemente de la instalación donde se programen (y por lo tanto, independientes de las variables, bus, o computadoras que aparezcan en la misma). Un primer criterio para saber si cumplen unos requisitos mínimos de abstracción es que los secuenciadores puedan ser compilados independientemente de la instalación, mediante la opción que aparece en el plugin de compilación. Si la compilación de un secuenciador termina con éxito, esto garantiza una cierta portabilidad del mismo a otros proyectos, lo cual es significativo en cuanto al nivel de abstracción logrado.

La definición de un secuenciador incluye el conocimiento de numerosos conceptos y puede ser ciertamente compleja, por lo que existe todo un capítulo dedicado a la misma en este manual. Véase el capítulo [Edición de Secuenciadores](#) para obtener más información.

Crear el Simbólico

Una vez que tenemos el bus y las definiciones de los elementos (los secuenciadores) a usar en la instalación, es el momento de unir las dos partes: de lo teórico a lo físico. El puente que se extiende para realizar este paso es la definición del Simbólico.

La definición del simbólico consiste en hacer corresponder tantos elementos como haya en la instalación con la parte de bus que les corresponde a cada uno y a la vez dotarles de un comportamiento mediante la asignación de los patrones definidos con los secuenciadores. Para ello vamos definiendo máquinas (Machines), que implementan el comportamiento de un secuenciador. Dentro de estas máquinas aparecerán las variables, ligadas o no al bus de campo, que se asociarán con los parámetros que defina el secuenciador elegido para su máquina. Como el propio bus de campo ya está aislado dentro de la edición de módulos de bus, tanto el simbólico como los secuenciadores son insensibles a las modificaciones posteriores del mismo (siempre hasta cierto punto, como es lógico: la desaparición de una rama de bus puede implicar cambios en el simbólico, por motivos obvios).

Tras completar la definición del simbólico, es conveniente asignar las máquinas creadas a los computadores desde donde se pretende que funcionen. Aquí es donde cobra importancia la ligación entre el bus y las tarjetas del simbólico: Debemos ser cuidadosos de no asignar máquinas a un computador que controlan partes de bus que no están gobernadas por tarjetas que pertenecen a dicho computador. Si así lo hiciéramos, estaríamos direccionando partes de bus incorrectas desde esa máquina, pudiendo esto llevar a solapes que pueden causar malfunciones en la instalación o incluso accidentes más graves.

La asociación de Máquinas con computadores se realiza a través del formulario de propiedades del computador. Si dejamos sin asignar alguna máquina, esta se asignará al computador que tenga el rango de "Computador por defecto".

Además de las Máquinas y sus variables, en el simbólico también se pueden definir "Zonas", que son conjuntos de variables que son vistas por más de una máquina. A diferencia con las máquinas, las zonas no se asocian con ningún computador, sino que se "adhieren" al computador donde están las máquinas que la usan. Esto puede llevar a conflictos, si máquinas que están en distintos computadores usan la misma zona. Por defecto, la resolución de este conflicto implica que la zona estará en el

computador donde se ejecuta la primera máquina que la usa. Debemos tener en cuenta si esto nos interesa, y si no es así optar entre dividir las zonas o replantearnos el diseño.

Realizar las Visualizaciones

Antes de poder compilar y generar el código del proyecto, aun nos queda dar otro paso que relaciona esta vez el funcionamiento de los secuenciadores con las máquinas reales en función de su posición real en la instalación. Este paso consiste en definir el orden en que las máquinas se encuentran. Dicho orden permite a un comportamiento genérico, como es el de los secuenciadores, saber con quien tiene que comunicarse para poder tomar decisiones en ciertas circunstancias.

Para poder añadir esta información, se usa un aspecto esencial del proyecto: La visualización.

En principio, las visualizaciones tienen una función informativa, a nivel gráfico, que nos permitiría ver el comportamiento de los elementos en la instalación desde un computador cualquiera. Para realizar esta tarea, la visualización se comporta como un SCADA que permite “dibujar” dichos elementos y asignarles un enlace con la máquina que se pretende que representen para obtener información de su estado, en forma de consulta de valores o posibilidad de cambio de los mismos. Como un añadido, al asignarles esos enlaces, se puede especificar en que orden van, relacionando entonces las máquinas mediante las relación de conectividad “anteriores a” y “posteriores a”.

Estas relaciones no son de equivalencia. Concretamente, no deberían que ser reflexivas, ni simétricas ni transitivas. Sin embargo, la relación de simetría se puede establecer en forma de reciprocidad de ambas, concretamente de la forma “Si A es anterior a B, entonces B es posterior a A”. Aun así, no es obligatorio que se cumpla esta relación.

Una vez introducida toda esta información (de nuevo ayudándonos de los planos de la instalación), tendremos un proyecto listo para ser compilado y generar un fichero .PLC con la información que alimentará al sistema Galileo y le permitirá comenzar a controlar la instalación de la forma deseada. Si la instalación es lo suficientemente compleja que requiera un SGA, aun tendremos que definir un paso más para permitirnos trabajar con dicho SGA.

Configurar el Agente de Transportes

Si necesitamos generar los movimientos mediante un SGA, es preciso que interactuemos con él de alguna manera. Repartidos por la instalación encontraremos elementos definidos como PIE, Puntos de Picking, Puntos de Finalización de Orden, Puntos de Búsqueda de Orden... Todos ellos tienen unas características definidas que son comunes a todas las instalaciones. Lo que varía entre ellas es la forma en que la acometen, o más concretamente, la información en la que se basan para llevarlas a cabo. Dicha información suele estar en manos de un SGA, y los secuenciadores que hayamos realizado para las máquinas donde se encuentran esos elementos, tienen que haber sido pensados con ese objetivo concreto.

Para acabar de ligar esas máquinas con el SGA, es necesario que establezcamos esa conexión a través de la configuración del Agente de Transportes, que permite definir:

Estaciones → Serán nuestra manera de hablar con el SGA, indican puntos dentro del layout de la instalación, que suelen corresponder con la posición de una máquina físicamente.

Trayectorias → Funcionarán como un mapa de los posibles movimientos que se pueden realizar entre 2 ó más puntos de la instalación por el SGA. Esto ayuda al SGA a poder decidir los movimientos a realizar en cada momento.

Ordenes → Serán la manera de indicar que máquinas de nuestro proyecto necesitan una interacción más directa con el SGA a fin de poder realizar su ciclo de trabajo.

Adicionalmente, desde esta configuración se puede configurar otra información que es más útil para la parte de control, pero que el SGA puede tener interés en conocer también.

Averías → Lista de posibles defectos que causan un comportamiento anómalo de una máquina. Este comportamiento normalmente conlleva la parada de la máquina en cuestión y la aparición de una alerta que informa de este suceso al sistema de control y al SCADA de mantenimiento a fin de que sea solucionada de forma oportuna.

Compilar el proyecto

Una vez que tenemos todos los posibles elementos programados y configurados, deberemos compilar el proyecto a fin de generar el archivo .PLC que contiene la información necesaria para que Galileo pueda manejar la instalación. Deberemos compilar el proyecto tantas veces como Computadores hayamos definido, haciendo una compilación con cada computador como “Objetivo” de la misma. Si todas ellas son correctas, podremos pasar los ficheros “programa.plc” a la carpeta “c:\galileo” de cada computador asociado, y arrancar el servicio Galileo entonces.

También podemos realizar esta transferencia desde el propio formulario de compilación, pero se requiere que el computador tenga el servicio Galileo funcionando ya. Esto puede ser muy útil para actualizar la versión de programa corriendo tras un cambio.

Crear y dar permisos a usuarios de proyecto

Cuando la instalación ya esta funcionando y lista para entregar al cliente, es el momento de crear Usuarios de Proyecto. La funcionalidad última de este paso es la de poder controlar las interacciones de los usuarios finales potencialmente peligrosas.

Por seguridad, el Sistema Galileo requiere que toda acción que represente riesgo o implique un cambio en el funcionamiento habitual del programa de control lleve asociado un usuario y una computadora desde la que fue solicitada. Esto le permite llevar un “log” de dichas acciones que puede ser consultado en caso de que sea necesario.

A fin de poder ofrecer esos usuarios, y ampliando aun más la seguridad ofrecida por Galileo, desde el entorno de desarrollo se permite crear Usuarios con ciertos perfiles. Cada vez que el usuario solicite la realización de una tarea potencialmente peligrosa por parte del Sistema Galileo, se solicita un usuario y su clave de acceso. Si la clave coincide y el perfil asociado al usuario permite la realización de esa opción, se remitirá la petición de la tarea al Sistema Galileo, siendo registrada dicha acción por el Sistema Galileo como desencadenada por dicho usuario desde el computador donde se encontrara en ese momento. Si los privilegios son insuficientes o la clave incorrecta, simplemente se ignorará la petición.

La clave son los perfiles. Hay operaciones que son realizadas de forma más o menos rutinaria como parte del mantenimiento de la parte de control. Otras, sin embargo, requieren un conocimiento más extenso por parte del usuario.

Los niveles van de menos a más, y cada nivel incluye los privilegios de los niveles inferiores. Actualmente, los niveles existentes son:

Nivel	Permisos para...
None	Acceso a las visualizaciones sin restricciones.
Basic	Uso del pupitre virtual.
Intermediate	Resetear una máquina a la etapa inicial. Borrar un tracking de una máquina. Añadir vistas de tablas en el plugin de tablas Oracle.
Expert	Resetear un secuenciador a una etapa específica. Cambiar la información de un tracking. Borrar todos los trackings de una máquina.
Developer	Poder ver el grafo y el código en ejecución de una máquina desde el editor de tracking. Poder ver los valores de las variables de una máquina desde el editor de tracking. Poder forzar el valor de una variables desde el Status de Variables.
RemoteAdmin	Ver, y modificar los permisos de los usuarios de proyecto. Crear nuevos usuarios de proyecto. Eliminar usuarios de proyecto.
Admin	Establecer privilegios de sesión. Cambiar permisos de acceso a un formulario.

Tabla 1: Permisos de usuarios de proyecto

- ✦ Establecer privilegios, para evitar tener que teclear un usuario y clave cada vez que se de una situación que lo requiera, es posible establecer de forma global el usuario que realiza las operaciones (solo desde el Designer) Para hacerlo basta pulsar con el botón derecho del ratón sobre el nodo de proyecto y seleccionar la opción “Establecer Privilegios”. Si introducimos un usuario con perfil “Administrador”, se usará este usuario en todas las operaciones que requieran un usuario durante esta sesión.

Los dos últimos niveles de acceso (RemoteAdmin y Admin) sólo deberían ser otorgados a usuarios pertenecientes a Telemantenimiento o Mantenimiento de ThyssenKrupp.

Depuración del proyecto

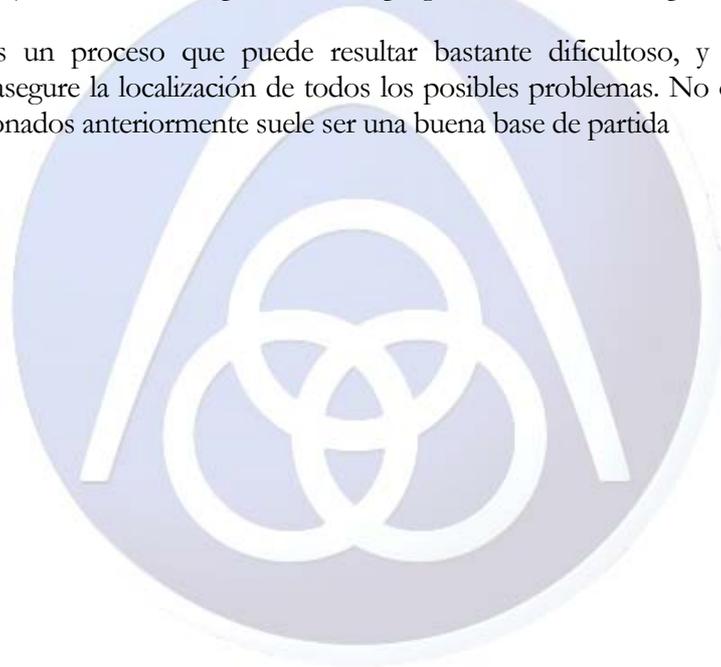
A la hora de poner en marcha el proyecto, es necesario ir depurando los posibles problemas que surjan a partir del código inicial. Para ello, el entorno de desarrollo dispone de una serie de herramientas que se explican en el Capítulo 11. Dichas herramientas van orientadas principalmente a poder realizar el seguimiento de una ejecución en zonas y lugares concretos. A partir de estas trazas, se modificará la parte del proyecto que corresponda. Normalmente los puntos de mayor interés a comprobar son:

- Comprobación de solapes: verificar que los solapes existentes sean deseados. Se puede comprobar los solapes mediante la opción Mapas de Memoria de la edición del simbólico.
- Seguimiento del funcionamiento de un secuenciador: Verificar que su grafo de ejecución corresponde con el funcionamiento deseado, y que condiciones anormales no lo saquen de

este funcionamiento. Se puede realizar desde el Editor de Tracking o desde la edición de secuenciadores.

- Comprobar valores de las variables en ejecución: Comprobar que las variables en ejecución tienen los valores esperados o cambiarlos para poder comprobar funcionamientos simulados. Se puede realizar desde el Editor de Tracking, el Pupitre Virtual o desde el Status de Variables.
- Comprobar el funcionamiento del Agente de Transportes: Verificar que los paquetes de procedimientos entregan ordenes cuando existen, comprobar la configuración del Agente en Galileo (conexión con la base de datos), comprobar la configuración del Agente de Transporte desde el entorno.
- Comprobar los ficheros de configuración: Es necesario verificar que los valores introducidos en ficheros de configuración (por ejemplo: mapas de posiciones) no causen comportamientos anómalos de la instalación (por ejemplo, ubicaciones en posiciones inexistentes, errores de deposito...). Para este fin se puede usar el plugin de edición de Mapas.

La depuración es un proceso que puede resultar bastante dificultoso, y no hay un patrón de seguimiento que asegure la localización de todos los posibles problemas. No obstante, el chequeo de los puntos mencionados anteriormente suele ser una buena base de partida



DESIGNER 3.0

Plugins y Formularios

El proyecto contiene muchas partes que deben ser puestas en común o programadas de forma separada o conjunta. Para especializar esta tarea se usan plugins que se especializan en la edición de una parte determinada del proyecto. Algunos de ellos, por su complejidad, requerirán de todo un capítulo aparte.

Los plugins son el sistema de trabajo que tiene el entorno de desarrollo para efectuar las tareas de configuración y creación de elementos asociados a un proyecto. En su mayoría, aparecen como una ventana hija de la aplicación, y se encargan de configurar un aspecto específico del sistema. Cada uno de ellos tiene su propia forma de funcionamiento, y pueden sufrir modificaciones independientemente del resto. Esto asegura una futura evolución por separado de estos plugins, sin necesidad de que el entorno principal sufra modificaciones severas, ganando en velocidad de desarrollo y calidad de las versiones generadas. Además permite detectar errores y generar los parches correspondientes de forma que tengan el mínimo impacto posible en la arquitectura global del sistema Galileo.

La programación de estos elementos queda fuera del ámbito de este documento, por lo que no se profundizará en el tema. No obstante, a continuación se comentarán los plugins por defecto que vienen integrados en la instalación del entorno de desarrollo, indicando su papel principal, modo de funcionamiento y condiciones de uso particulares.

Propiedades de un Computador

Este plugin permite configurar las opciones de un computador que forme parte del proyecto. Para acceder a él hay que seleccionar un nodo del árbol de proyectos que represente una computadora y pulsar con el botón derecho del ratón para acceder a un menú donde aparecerá la opción “Propiedades”. Al seleccionar dicha opción, aparecerá el plugin, con un aspecto similar al de la Ilustración 15:

DESIGNER 3.0

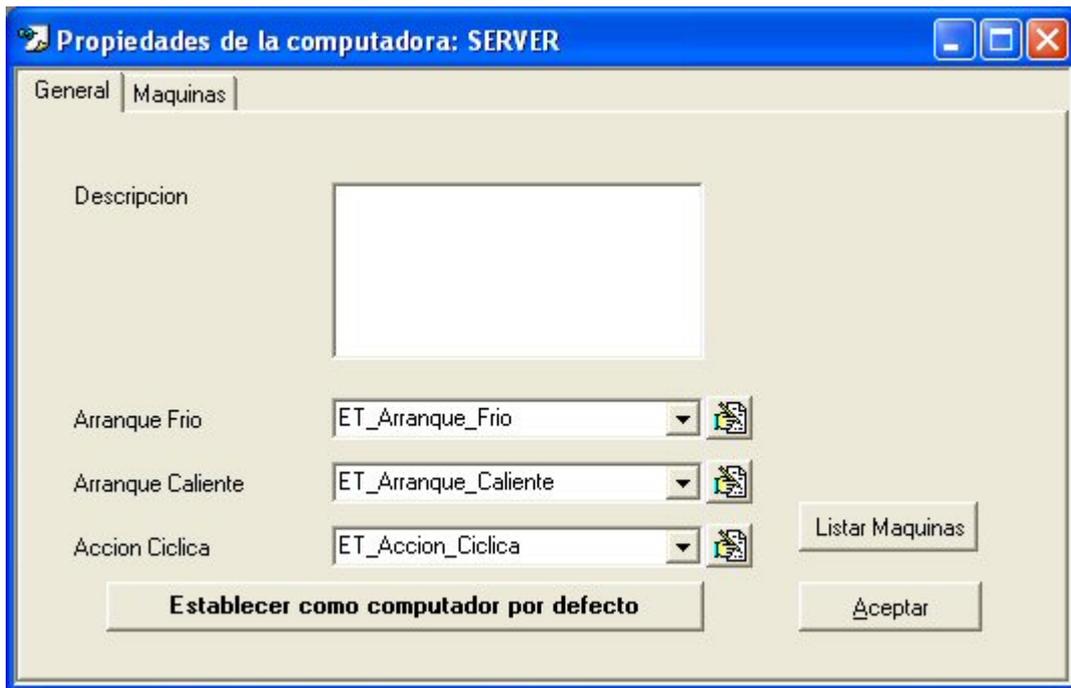


Ilustración 15: Propiedades de una computadora.

Hay dos solapas que permiten editar cada una parte de la configuración del computador. Veámoslas.

Solapa "General"

Desde esta solapa podemos establecer unas propiedades básicas de la computadora:

- Descripción: Permite que introduzcamos una descripción para el computador. No se mostrará en ningún otro sitio, así que sólo tiene un valor informativo para los desarrolladores.
- Arranque frío: Permite especificar una función global, de tipo Etapa, que se ejecutará cuando el Sistema Galileo arranque con este proyecto en esa computadora.
- Arranque Caliente: Permite especificar una función global, de tipo Etapa, que se ejecutará cuando el Sistema Galileo re arranque este proyecto (sin haber sido detenido o reiniciado antes) en esa computadora.
- Acción Cíclica: Permite especificar una función global, de tipo Etapa, que se ejecutará en cada ciclo de trabajo del Sistema de Control Galileo.

Los arranques y Acción Cíclica, En las versiones 2.x, estas funciones estaban a nivel de tarjeta, pero dado que las tarjetas pierden su protagonismo a favor de los computadores, es lógico que estos atributos pasen ahora a ellos. Su función, no obstante, es absolutamente la misma en ambas versiones.

Asimismo, tenemos la opción de establecer la computadora como "Computadora por defecto". Si lo hacemos, pulsando el botón, el icono de la computadora en el árbol cambiará, para resaltar el hecho de que esa es la nueva computadora por defecto. El hecho de que una computadora sea considerada "Computadora por defecto" significa que toda máquina que no está asociada a un computador se asociará a esta computadora a la hora de compilar el proyecto. Si no marcamos ninguna computadora con esta propiedad, el entorno le asocia este rol a la primera computadora definida en el proyecto.

Para ver mejor lo que significa asociar máquinas con computadoras, hay que irse a la siguiente solapa:

Solapa “Máquinas”

Desde aquí podemos ver las máquinas que han sido asociadas para ejecutarse dentro de la computadora, tal y como vemos en la Ilustración 16:



Ilustración 16: Máquinas asociadas a la computadora.

A la derecha aparecerá un listado de las máquinas aun no asociadas a una computadora, podemos filtrar también para ver solo las máquinas que tengan variables que usen módulos de bus que pertenezcan a tarjetas de esta computadora (que debería ser siempre la realidad), o bien mostrarlas todas (menos las ya asignadas a esta computadora). En cualquier caso, si asociamos una máquina previamente ligada a otra computadora, esta se “moverá” de la otra computadora a la actual, ¡pero hemos de recordar que las tarjetas de control no lo harán!, de forma que si la máquina requiere de módulos de bus que pertenecen a la otra computadora, tendremos problemas con la configuración del bus.

Para añadir una máquina, basta con seleccionarla de la lista de la derecha y pulsar el botón “Añadir”. Para sacar una máquina de esta computadora, basta seleccionarla de la lista de la izquierda y pulsar el botón “Borrar”. En ambos casos es posible realizar una selección múltiple manteniendo pulsadas las teclas **Shift** y **Ctrl**.

Propiedades de una tarjeta

Este plugin permite, entre otras cosas, establecer las propiedades más elementales asociadas una tarjeta. Se accede a él seleccionando una tarjeta de un computador en el árbol de proyectos principal, o bien al pulsar con el botón derecho sobre una tarjeta, el menú contextual nos permitirá borrarla o ver sus propiedades.

La primera opción elimina la tarjeta seleccionada del computador. Nótese que esto provoca que cualquier módulo asociado a esa tarjeta queda desligado automáticamente, y por lo tanto debemos volver a ligarlo desde la edición de bus o no podremos generar código que use variables en ese módulo.

Si seleccionamos la segunda opción, aparecerá el plugin de propiedades con la información de la tarjeta seleccionada, con un aspecto similar al de la Ilustración 17.

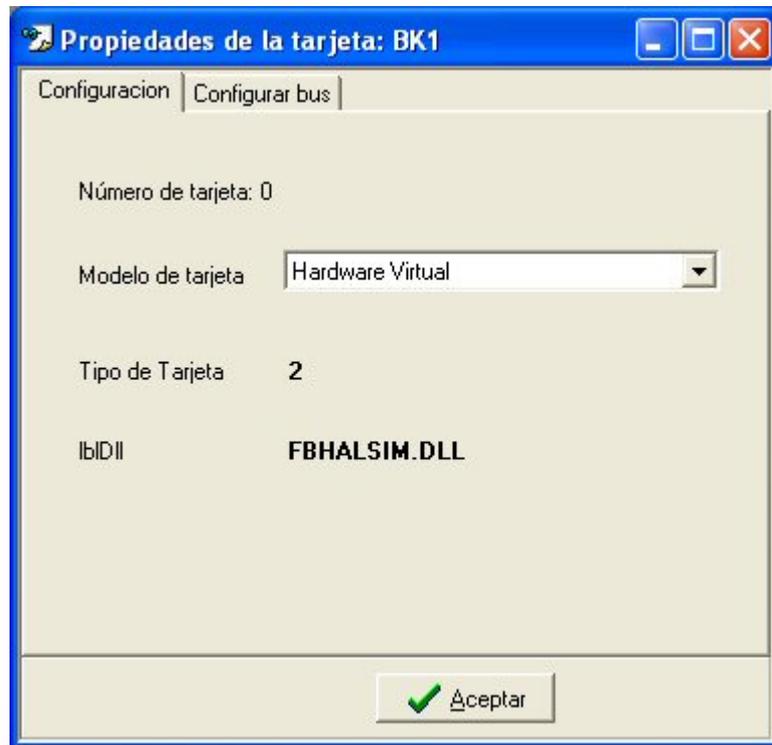


Ilustración 17: Plugin de configuración de las propiedades de una tarjeta.

Como se puede comprobar, el plugin consta de varias solapas que permiten acciones específicas, que se comentarán a continuación.

Solapa "Configuración"

Esta solapa es la que aparece por defecto, y muestra información relevante de la tarjeta, como por ejemplo, el número de esta (dicho número sirve para asociarla con el Slot PCI donde se inserta en la estación de control). También aparece un combo donde se puede especificar el tipo de tarjeta que se encuentra en el slot. Actualmente los modelos disponibles se muestran en la tabla siguiente:

TARJETA	TIPO	DESCRIPCIÓN	DLL
Hardware Virtual	2	Sin tarjeta: Se emulan las entradas y salidas por software.	FBHALSIM.DLL
Hilscher CIF-50 IBM	1	Tarjeta de control Hilscher Interbús	FBHAL.DLL
Hilscher CIF-30 IBM	1	Tarjeta de control Hilscher Interbús	FBHAL.DLL
Hilscher CIF-104 IBM	1	Tarjeta de control Hilscher Interbús	FBHAL.DLL
Hilscher COM IBM	1	Tarjeta de control Hilscher Interbús	FBHAL.DLL
Hilscher CIF-50 PB	3	Tarjeta de control Hilscher Profibus	FBHALPB.DLL
Hilscher CIF-60 PB	3	Tarjeta de control Hilscher Profibus	FBHALPB.DLL
Hilscher CIF-80 PB	3	Tarjeta de control Hilscher Profibus	FBHALPB.DLL
Hilscher CIF-100 PBM	3	Tarjeta de control Hilscher Profibus	FBHALPB.DLL
Hilscher CIF-104 PB	3	Tarjeta de control Hilscher Profibus	FBHALPB.DLL
Hilscher CIF-30 COM	4	Tarjeta de control Hilscher CMBUS	FBHALCAN.DLL
Hilscher CIF-50 COM	4	Tarjeta de control Hilscher CMBUS	FBHALCAN.DLL
Hilscher CIF-60 COM	4	Tarjeta de control Hilscher CMBUS	FBHALCAN.DLL
Hilscher CIF-81 COM	4	Tarjeta de control Hilscher CMBUS	FBHALCAN.DLL
Hilscher CIF-104 COM	4	Tarjeta de control Hilscher CMBUS	FBHALCAN.DLL

Tabla 2: Tipos de Tarjeta de Control

Solapa "Configurar Bus"

Esta solapa permite configurar el tamaño en bytes de las entradas/salidas que se le van a asociar a esta tarjeta. Para ello se deben especificar las posiciones de inicio/fin de las entradas y salidas que serán gobernadas por la tarjeta. Hay que tener en cuenta que esas posiciones limitan el tamaño de los elementos que se pueden insertar en el bus de la tarjeta, con lo que si escogemos un rango muy restrictivo, al editar el bus recibiremos de forma repetida mensajes indicándonos que los elementos sobrepasan el rango de entradas/salidas asignadas para la tarjeta que controla estos módulos.

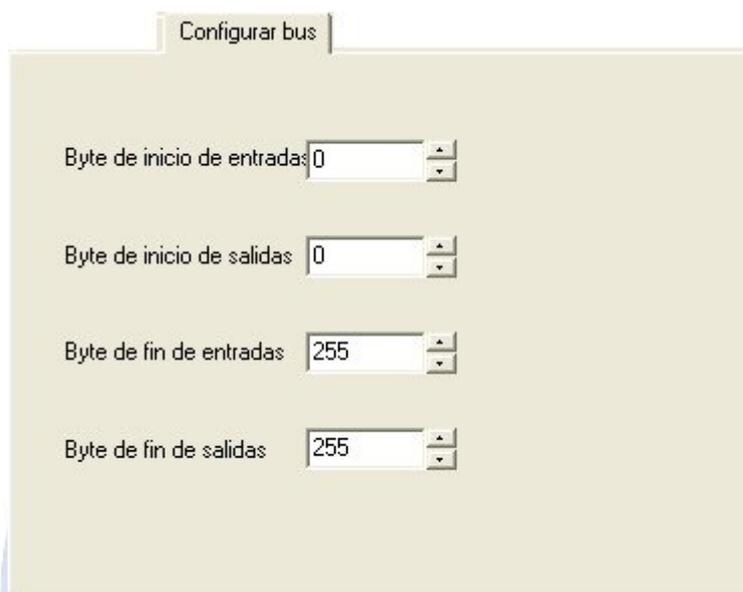


Ilustración 18: Configuración de entradas y salidas de una tarjeta.

Las direcciones de inicio y fin son absolutas, es decir, que siempre una dirección de inicio debe ser menor que la dirección de fin correspondiente.

Edición de Bus

Mediante este plugin se permite la identificación y etiquetado de los elementos del bus que componen la instalación y que van a ser gobernados desde una tarjeta de control específica, sita en una computadora concreta.

Para tener acceso a este plugin, basta con tener una tarjeta seleccionada, y utilizar la opción **Modificar / edición de Bus** del menú principal, momento que aparecerá el formulario de la Ilustración 19.

DESIGNER 3.0

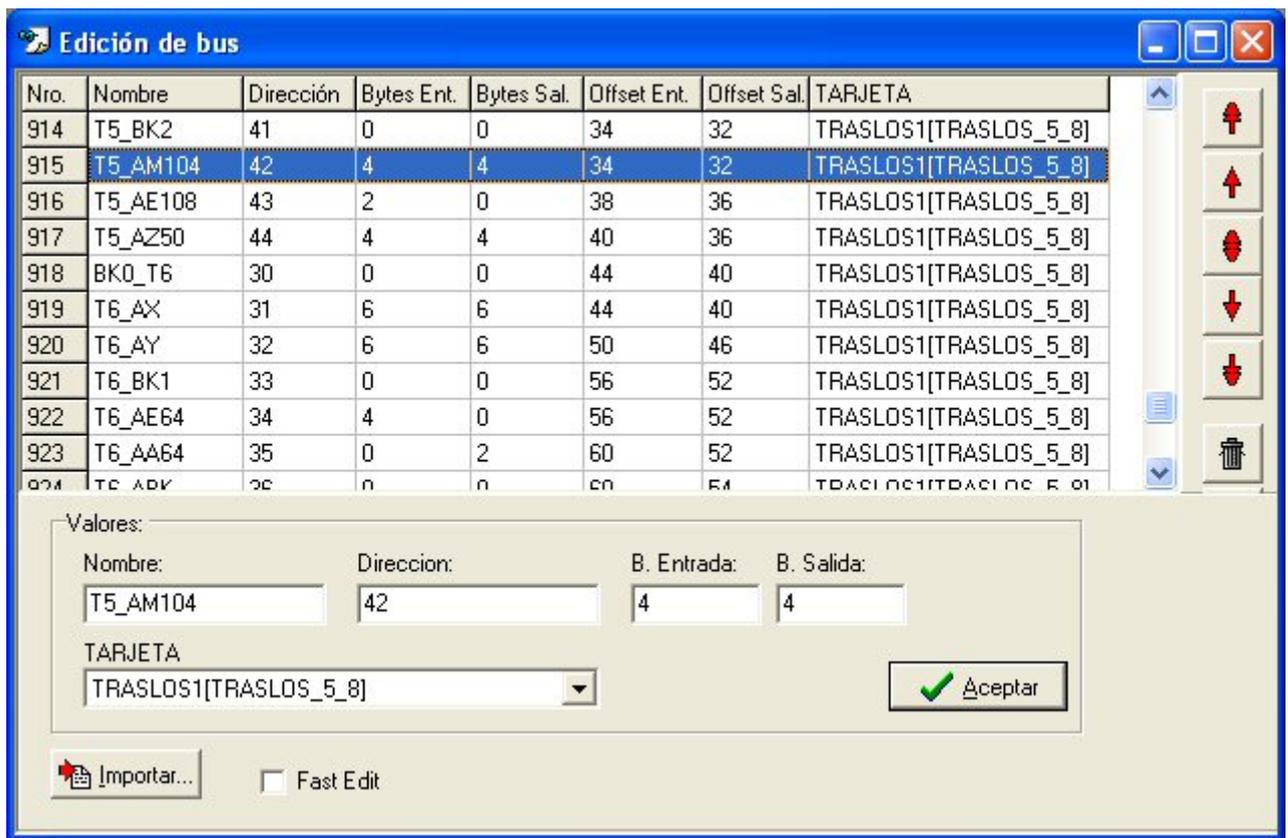


Ilustración 19: El plugin de edición de bus.

La tabla tiene una serie de columnas en las que se almacenan: el nombre de un elemento de bus, su dirección (campo descriptivo), los bytes de entrada que ocupa, los bytes de salida que ocupa, el desplazamiento de entrada y salida que le corresponde y la computadora y tarjeta en el que se encuentra. Dichos valores pueden modificarse mediante los campos que se encuentran en la parte inferior, pulsando el botón OK tras la modificación para validar los cambios, a fin de que se integren en la tabla.

Nótese que los campos “desplazamiento” no se pueden modificar, ya que vienen calculados por su orden respecto al resto de los elementos del bus y los bytes de entrada / salida ocupados por los elementos que van delante de ellos en la lista (y que pertenezcan al mismo computador).

La posición de un módulo en la lista es la que determina sus desplazamientos de entrada y salida, y cada vez que movemos un elemento por la lista implica que dichos desplazamientos deben recalcularse (al igual que los de sus elementos adyacentes). Dicho recalcule puede resultar una operación larga cuando la configuración contenga un número importante de elementos. Para hacerla más ágil, se puede pulsar el check de la parte inferior, lo que activará el modo “Fast Edit”.

Fast Edit, En este modo, el usuario va añadiendo cambios a la configuración, pero estos no se aplican hasta que se pulse el botón que aparece a la derecha del check de “Fast Edit”. Esto hace que la edición sea mucho más rápida, al no tener que calcular los desplazamientos, pero debemos de recordar que si salvamos el proyecto sin haber dado antes al botón de aplicar, esos cambios no se aplicaran, y lo mismo pasa si cerramos el plugin sin salvarlos.

Si se quiere añadir un elemento de bus nuevo, basta con seleccionar una columna vacía y editarla de forma normal.

Si se desea borrar un elemento del bus de la lista, basta con usar el botón  situado en la parte derecha de la pantalla.

Para modificar la posición en el bus de un elemento (esto puede afectar a su desplazamiento), basta con seleccionarlo en la lista y usar los botones con flechas de dirección  y  situados en la parte derecha, para adelantarlo o retrasarlo una posición en la lista respectivamente. Para moverlo a las posiciones inicial o final de la lista, se pueden usar los botones con los iconos  y  respectivamente. Si queremos mover el módulo a una posición más concreta, se puede usar el botón con el icono , que preguntará por el número de la posición a donde queremos mover el módulo, y lo moverá a esa nueva posición (desplazando los elementos hacia arriba o hacia abajo según sea necesario). Es posible mover varios módulos a la vez, seleccionándolos con la tecla **Shift**, pero deben estar situados de forma seguida en la tabla.

También se puede observar un botón  “Importar” en la parte inferior izquierda, que permite importar información de bus desde un archivo de configuración generado con la herramienta “Sycon” de Hilscher.

Compilación

A la hora de realizar la compilación, se invoca a un plugin que muestra un formulario como el que aparece en la siguiente figura:

 El Plugin de compilación, En las versiones anteriores, la compilación se realizaba directamente desde la pantalla principal de la aplicación. Al aumentar la complejidad del proyecto, ha sido necesario añadirle más opciones y posibilidades a la compilación, así que para tener mayor control de las mismas, se han concentrado en este nuevo formulario.

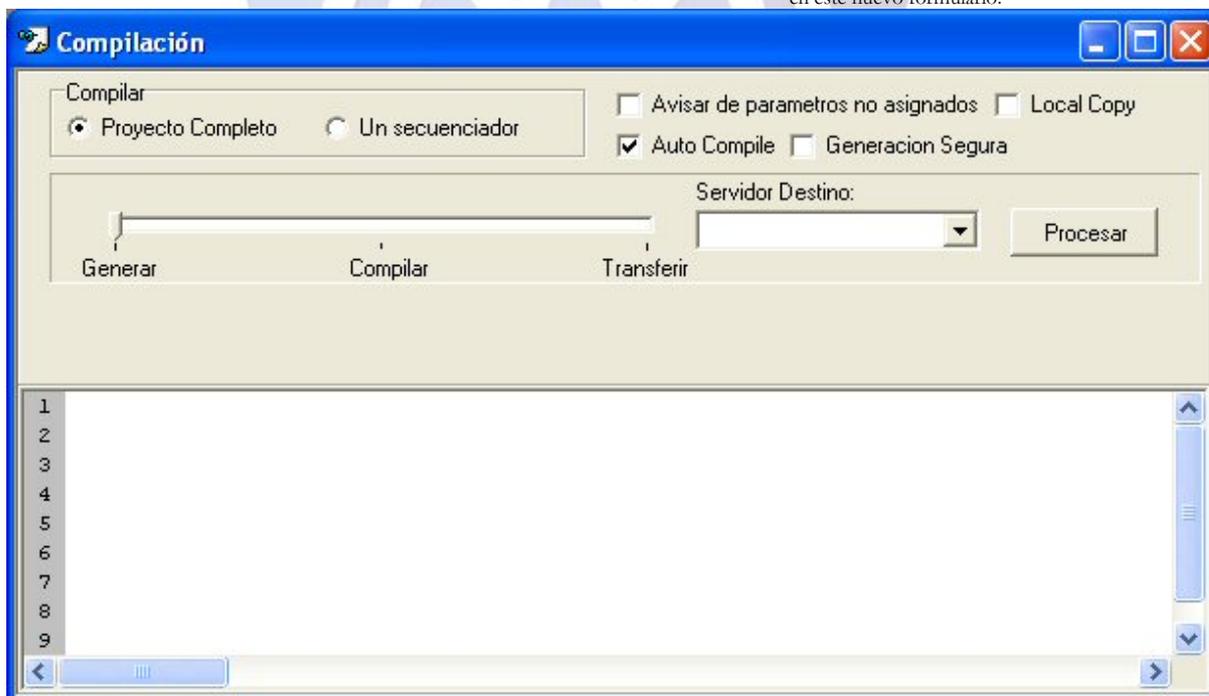


Ilustración 20: Formulario de compilación.

Desde este formulario se puede realizar la compilación del proyecto (para una computadora) o de un secuenciador del mismo (a fin de comprobar su validez).

Si el proceso de compilación termina con error, aparece un botón que al pulsarlo intenta localizar el código donde se produjo el error y mostrarlo de manera que el programador pueda modificarlo de inmediato. Dependiendo del tipo de error producido (por ejemplo, un error en la generación de variables), puede ser posible que no sea posible localizar este código.

Compilación de un secuenciador

A fin de poder comprobar la validez y abstracción “teóricos” de un secuenciador, es posible compilarlos como objetos independientes del proyecto. Al seleccionar esta opción, la parte superior del formulario muestra el siguiente aspecto:

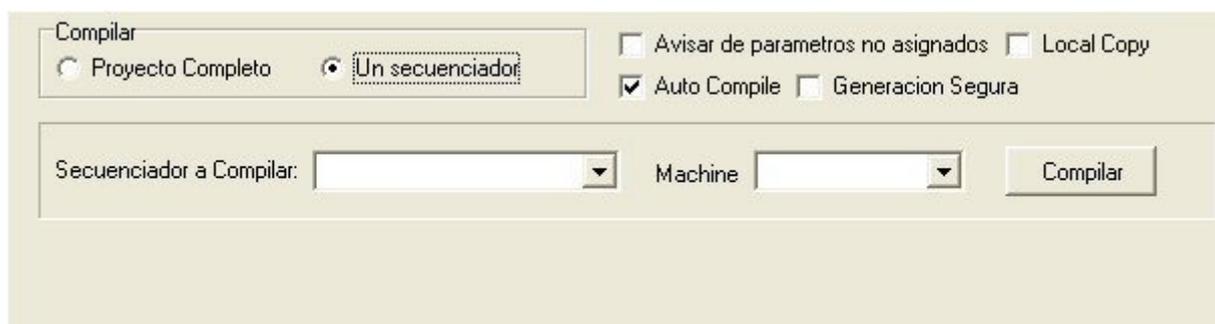


Ilustración 21: Opciones de compilación de un secuenciador

En el combo de la izquierda aparece la lista de secuenciadores existentes en el proyecto, debemos elegir el secuenciador que nos interese comprobar, para a continuación pulsar el botón de “Compilar”.

Para compilar un secuenciador, no se usa la información del simbólico, por lo que los parámetros actúan como si fueran variables ficticias ya asignadas. En algún caso concreto (por ejemplo, con el uso de la palabra reservada `inline` en el código del secuenciador), esto puede no ser suficiente para generar el código de forma apropiada, sin que esto sea indicativo de una mala programación del secuenciador. En estos casos, lo más conveniente es usar el combo de la parte derecha y asignar una máquina “real” para hacer la generación... en este caso se tomará la máquina para generar el código, y las cadenas, constantes o código `inline` se expandirán con los valores de dicha variable. Esto permitirá saber si los fallos de compilación son reales o no.

Por la misma razón, la compilación de un secuenciador se realiza en un solo paso: generando y compilando en un solo proceso, que no se puede interrumpir hasta que se complete.

Para la compilación de un secuenciador, los pulsadores de la parte superior derecha no se aplican, dado que no se genera un fichero de resultado con ella.

La compilación de un secuenciador por tanto, solo tiene un valor informativo, no se genera nada útil al finalizar la misma. Es, por tanto, una herramienta más para el programador.

Compilación del proyecto

La compilación de un proyecto es un proceso mucho más lento que en el caso de un secuenciador. La parte superior del formulario muestra ahora este aspecto:

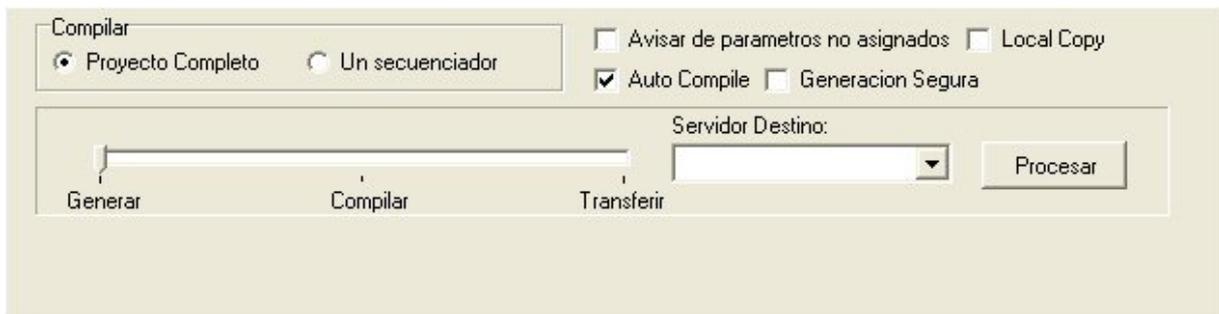


Ilustración 22: Opciones de compilación de un proyecto

Para empezar se debe elegir la computadora del proyecto para la que queremos realizar la generación y compilación. Este paso es importante, porque los ficheros generados serán distintos en función de la computadora seleccionada, dado que cada computadora lleva sus propias máquinas y también sus propios módulos de bus (determinadas por las tarjetas que contiene).

La compilación de un proyecto se realiza en 2 pasos: Generar el código Xana fuente del proyecto y compilarlo y empaquetar el código máquina resultante en un fichero .plc que es el que contiene toda la información necesaria para que Galileo pueda ejecutar ese programa. Se puede hacer que esos pasos se produzcan de forma simultanea usando la opción de auto-compilación, que provocará que se pase al siguiente paso de forma automática si el primero terminó con éxito.

La parte de generación de código tendrá ahora en cuenta la información del simbólico para generar el código de todas las máquinas que pertenezcan a la computadora. Por eso el fichero generado puede ser de un tamaño mucho mayor que para cualquier secuenciador. Si la generación y compilación se realizan con éxito, podremos realizar el paso final de transferir el fichero generado a la computadora de destino.

La transferencia del fichero “programa.plc” puede ser llevada a cabo sólo en el caso de que nos encontremos en modo “online” y que dicho computador tenga el servicio Galileo corriendo. En este caso, se establece una conexión de red entre Galileo y el entorno de desarrollo para transferirle el fichero, que deberá ser localizado siempre a la carpeta c:\Galileo del computador de destino.

En el caso de que el entorno de desarrollo se encuentre en el mismo computador de destino, se puede usar la opción “local compile”, que realiza la copia del fichero de forma local, sin necesidad de que estemos en modo “online”, y por tanto, mucho más rápidamente.

Mientras la generación o compilación se estén realizando, una barra de progreso en la parte inferior del formulario nos permitirá ver como avanza la operación, sirviendo de estimación para saber cuando va a terminar. También se muestra un botón que nos da opción de “Cancelar” la compilación o generación que se esté realizando en ese momento. Por desgracia, este no es un proceso sencillo, y puede que su detención lleve algún tiempo.

Si está configurado el uso de sonidos, se reproducirá un sonido para indicar el fin de un proceso de compilación o generación. Este sonido varía en función de que el proceso haya terminado bien o con errores, que se mostrarán en el cuadro de texto que ocupa la mayoría del formulario, junto con otras notas informativas (tiempos de generación o compilación) o avisos que puede ser conveniente revisar.

Si queremos más información sobre las variables generadas, podemos pulsar la comprobación de “Avisar de parámetros no asignados” que adicionalmente comprobará que todos los parámetros definidos por el secuenciador tienen una variable asignada en las máquinas cuyo código se está generando. Si en algún caso no fuese así, se generaría una línea de aviso. Esto no tiene porque provocar el fallo de la compilación, puede darse el caso que dicho parámetro no se esté usando y por lo tanto podemos descartar el aviso de forma segura (esto es normal en el caso de parámetros heredados por un secuenciador del padre, pero que no se usan en su código).

El botón de comprobación que aun no hemos explicado es el denominado “generación segura”. Este botón está disponible a fin de solucionar un posible error en lo que es la visión de los métodos heredados de un secuenciador. Mediante el uso de métodos heredados, es posible que un secuenciador tenga asociado un método perteneciente a un ancestro no directo, pero que ha sido sobrecargada en un ancestro más directo. Como los métodos son sobrecargados por orden de herencia (es decir, la función de un descendiente siempre oculta a la del padre), esto lleva a que la fase de generación de código no es capaz de encontrar el método buscado, y por lo tanto, dicho código no aparecerá en el programa (con los consiguientes problemas que esto puede acarrear). Las causas por las que esta situación puede darse son variadas, y tienen que ver en su mayoría con la forma y orden en que se hayan editado los métodos y secuenciadores.

Si se sospecha de esta situación, al activar esta comprobación, la fase de generación de código usará un algoritmo diferente para acometer la tarea, y si no encuentra el método referenciado, intentará suplirlo con el más cercano (ignorando la sobrecarga de métodos). En la mayoría de los casos esto solucionará el error, pero es necesario revisar el código de los secuenciadores a continuación para tratar de corregir esa anomalía. Por lo tanto, esta opción no debería usarse de continuo, ya que evitaría que el programador se apercibiera de este error.

Usuarios del proyecto

Como ya se explicó anteriormente, los usuarios de proyecto son perfiles automáticamente creados que permiten limitar el uso de ciertas opciones peligrosas sobre el sistema Galileo a través del entorno de desarrollo. Estas limitaciones pueden servir para evitar que una manipulación accidental del proyecto ocasionen daños importantes en la instalación por parte de un usuario inexperto. Para una explicación más avanzada del papel de estos perfiles, véase el apartado [Crear y dar permisos a usuarios de proyecto](#).

A la hora de crear, modificar o borrar estos usuarios, es necesario acceder a este plugin, lo cual se logra pulsando sobre el nodo de USUARIOS que cuelga del cada proyecto en el árbol de proyectos. Al hacerlo, aparecerá el formulario que se muestra en la siguiente ilustración:

DESIGNER 3.0

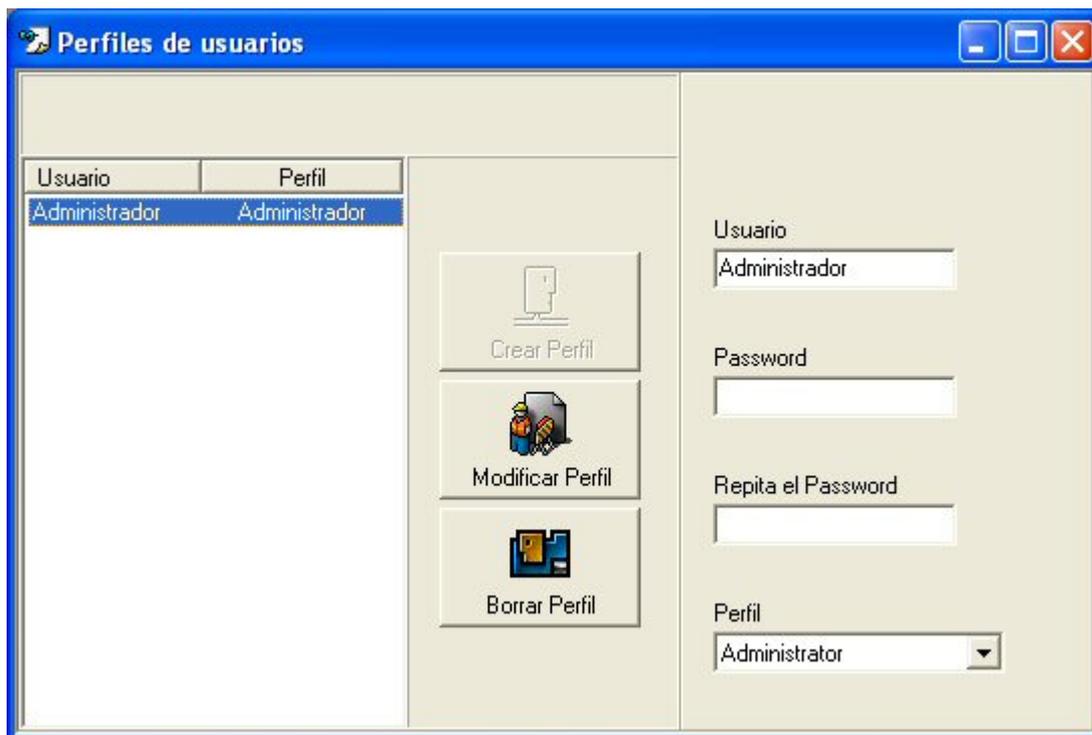


Ilustración 23: Formulario de edición de perfiles

En la parte izquierda aparecen listados los usuarios ya creados y los perfiles que tienen asociados. A la derecha de esta lista aparecen los siguiente botones:



Crear Perfil

Permite crear un nuevo usuario. Antes de pulsarlo deberemos haber introducido los datos para su creación en los campos de la parte derecha del formulario. Si los datos son correctos, el nombre y perfil del nuevo usuario se añadirán a la lista de los ya existentes.



Modificar Perfil

Permite modificar los datos de perfil de un usuario. Cuando pulsamos sobre un perfil de usuario, sus datos se muestran en los campos de la derecha del formulario (excepto la clave de usuario, que aparece con asteriscos). Si modificamos alguno de esos datos (cambiar el perfil o la clave de usuario), pulsando sobre este botón los aplicaremos al proyecto actual.



Borrar Perfil

Permite borrar un usuario de la lista. Previamente tendremos que haberlo seleccionado de la lista.

Por supuesto, editar estos datos o tener acceso a ellos requiere a su vez de un cierto nivel de privilegios. Cuando se pretenda alterar estos datos, se nos pedirá un usuario de proyecto que tenga perfil de Administrador. Si no introducimos los valores adecuados, no podremos cambiar nada de esta configuración (ni siquiera ver los datos de los usuarios, solo la lista).

Evidentemente, la primera vez que abrimos este formulario en un nuevo proyecto, no existe ningún usuario creado, y mucho menos con perfil de Administrador. Aun así, se nos preguntara por uno... en este caso deberemos introducir como nombre de usuario "Administrador" y dejar la clave de usuario

en blanco. Entonces veremos que aparece Administrador en la lista de usuarios, con perfil “Administrador”. Es conveniente cambiarle la clave de acceso en ese momento. Y luego proceder a crear los usuarios que estimemos oportunos con los permisos adecuados para cada uno.

Edición de Secuenciadores

Se accede a este plugin pulsando con el botón derecho del ratón sobre el icono de cualquiera de los secuenciadores ya creados en el árbol de proyectos y eligiendo la opción “Propiedades” del menú o bien usando la opción **Insertar / nuevo secuenciador** del menú principal, momento en que el diálogo de la Ilustración 24 aparecerá para solicitar un nombre para el nuevo secuenciador. Dicho nombre, además de significativo, deberá cumplir las reglas de los identificadores XANA.

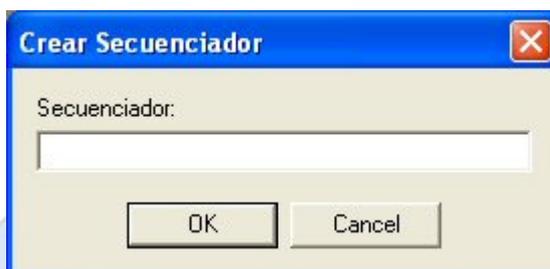


Ilustración 24: Diálogo de creación de un nuevo secuenciador.

Al igual que el resto, este nombre puede ser cambiado posteriormente sin más que editar el nombre del nodo que aparece en el árbol de proyectos.

Los secuenciadores que son “hojas” del árbol principal también se abren con solo pulsar con el botón izquierdo del ratón. Para los que son padres de otros secuenciadores, es necesario abrirlos usando el menú.

Los secuenciadores se muestran en el árbol de proyectos representados en función de su relación de herencia (los secuenciadores hijos como nodos de sus padre).

Los secuenciadores que pertenecen al repositorio no pueden ser editados, pero se pueden abrir en modo solo-lectura para revisarlos.

En el menú del botón derecho, también existe la opción de “Eliminar” el secuenciador por completo, haciendo que desaparezca del proyecto. La eliminación de un secuenciador tiene varias consecuencias importantes:

- Los hijos del nodo eliminado pasan a ser hijos del padre del nodo eliminado. Si no existiera ese padre, dichos hijos quedarían como secuenciadores raíz.
- Los métodos y parámetros del secuenciador desaparecen. Si sus hijos los usaban, puede ocurrir que empiecen a usar los de su nuevo padre, o bien que desaparezcan esas referencias en etapas o transiciones.
- En la edición del simbólico, todas las máquinas que implementaban ese secuenciador pasan a tener secuenciador vacío. Por lo tanto, al no tener parámetros, todas las asociaciones

parámetro-variable también son eliminadas. Ni la máquina ni sus variables son eliminadas, pero es necesario editarlas para asignarles un secuenciador válido.

La edición de Secuenciadores es explicada en profundidad en el Capítulo 6

Edición de Código

Como ya se ha comentado antes, la parte de control se basa en el código XANA que el usuario programa para efectuar el control de los transportadores en la instalación. Este código se divide en métodos que después se invocan desde las acciones de etapa o transición. Para editar este código de la forma más cómoda posible, la aplicación incorpora un formulario con un entorno que ofrece sintaxis resaltada (la del lenguaje XANA), opciones de búsqueda, reemplazar, imprimir y las funciones más básicas de cualquier editor de código.

Dicho formulario puede aparecer integrado dentro de la edición de Secuenciadores, donde solo podrán editarse los métodos pertenecientes al secuenciador, o bien se puede usar la opción **Modificar / Editar código** del menú principal, o pulsando el nodo "CÓDIGO" que cuelga del raíz del cada proyecto del árbol de proyectos. También es posible invocar este formulario desde la edición de propiedades de una computadora, al pulsar sobre los iconos de las funciones de arranque y cíclicas. Cuando aparece, muestra un aspecto similar al de la Ilustración 25.



Ilustración 25: Plugin de edición de código.

En la ilustración ya se está editando una función, pero normalmente aparecerá vacío. La parte superior del plugin contiene una paleta de herramientas que se explica a continuación.

La paleta de herramientas

Desde aquí se gobiernan las funciones que se pueden aplicar al código y las que controlan el plugin. Se describen de izquierda a derecha:



Este botón permite crear una nueva función. Al pulsarlo aparecerá el diálogo que se muestra en

la Ilustración 26.

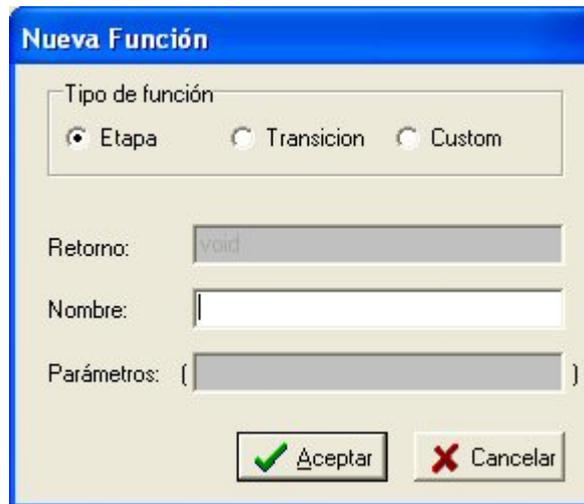


Ilustración 26: Creación de una nueva función.

Según sea una función para usar como acción de etapa, de transición, o totalmente distinta, elegiremos el tipo de función y le daremos un nombre, que no debe coincidir con el nombre de ninguna función ya definida en el proyecto o en el secuenciador. Si lo que queremos es una función de tipo “custom”, tendremos también la posibilidad de elegir el tipo de retorno y los parámetros que llevara la función. Al escribir dichos parámetros y tipos, hay que observar las reglas de la sintaxis del lenguaje XANA, pues de otra manera aparecerán errores durante la compilación posterior del código generado. Cuando hayamos completado dichos parámetros, basta con pulsar el botón “OK” y aparecerá una pestaña nueva con el prototipo (en XANA) de la función que acabamos de crear.

 Causa la aparición de Minibrowser de funciones el cual, básicamente, permite abrir una función ya creada para su edición. Como su funcionamiento tiene bastantes posibilidades, se explica de forma más completa en el apartado [El Minibrowser de funciones](#)

  Proporcionan la misma funcionalidad: salvar los cambios hechos en funciones en el proyecto o en el secuenciador (depende de donde estemos viendo el formulario). Mientras que el primero sólo salva la función que estemos editando actualmente, el segundo salva los cambios en todas las funciones que tengamos abiertas actualmente en el formulario de edición.

 Permite renombrar una función, cambiando su nombre a otro que no debe existir previamente.

  Proporcionan las herramientas “buscar” y “buscar y reemplazar” de sobra conocidas por cualquier programador. Cuando se pulsa sobre ellas, aparecen los diálogos de dichas operaciones, que también pueden ser accesibles usando las combinaciones de teclado **ctrl.+F** y **ctrl.+R**.

 Borra el código de la función actual, pasando a ser una función vacía. Un diálogo solicitará confirmación antes de llevar a cabo esta acción.

 Elimina la función y cierra la ventana de edición de la misma. La función ya no estará disponible en el secuenciador o proyecto. Como antes, un diálogo solicitará confirmación antes de aplicar

esta opción, avisando de las consecuencias de la misma. Estas pueden ser importantes, por ejemplo, si estamos en el caso de edición de un secuenciador, dicho código desaparecerá de las etapas y transición, y lo mismo ocurrirá en los secuenciadores hijos del editado.

 Cierra la edición de código de la función que se está editando. Esto provoca que la solapa donde se encuentra desaparezca, pero el código no se borra, solo deja de estar en edición.

 Permite la edición del **código libre** asociado al proyecto. El código libre es un conjunto de funciones especiales que el usuario puede introducir en la tarjeta, y que se mantiene por compatibilidad con las primeras versiones de la aplicación. Se recomienda usar funciones “Custom” en lugar de escribirlas en el código libre. Al invocar esta opción, aparece una nueva pestaña en la edición de código, que contiene todo el código libre definido en el proyecto, y se edita como cualquier otra función. Este icono no aparece cuando lo que edita son las funciones de un secuenciador. Esta opción aparece por compatibilidad con las versiones 2.x, y debería desestimarse su uso.

 Permite la impresión de la función que se está editando actualmente. El usuario puede variar alguna de las características de la impresión en el diálogo que aparecerá a continuación de pulsar esta opción, tales como la impresión del número de línea o imprimir la sintaxis.

 Al pulsar este botón, el método mostrado actualmente sufrirá un intento de alineación a fin de “embellecer” su código. Esta alineación se basa en el sangrado del texto a partir de que se encuentren los elementos clave de la sintaxis XANA, por ejemplo, llaves y paréntesis. Si el código no está balanceado (es decir, faltan o sobran llaves o paréntesis, lo cual es un fallo de sintaxis), al finalizar el proceso se mostrará un aviso sobre este particular. El código, una vez modificado, se muestra en pantalla, pero no se salvará hasta que pulsemos sobre uno de los botones de salvado.

El Minibrowser de funciones

Cuando el minibrowser aparece, muestra una lista con los prototipos XANA de las funciones definidas en el proyecto o secuenciador hasta el momento, tal y como se muestra en la Ilustración 27.

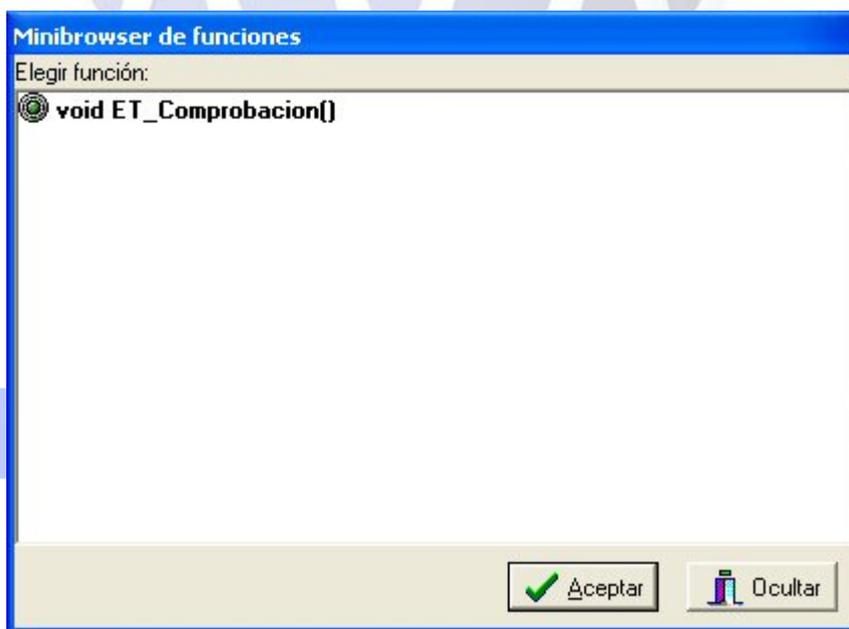


Ilustración 27: Minibrowser de funciones.

Para pasar a editar una función de las listadas, se puede o bien hacer una doble pulsación con el ratón sobre su prototipo, o bien seleccionarla y pulsar el botón “OK” de la parte inferior. también es posible seleccionar varias funciones a la vez y abrirlas todas con una única pulsación de ratón sobre “OK”.

El minibrowser no se cierra cuando se selecciona una función. Sigue abierto, permitiendo editar más. Si se desea cerrarlo, hay que pulsar el botón “Ocultar” de la parte inferior del mismo, o volver a pulsar el segundo icono de la paleta de herramientas del plugin.

Las funciones definidas llevan asociadas iconos de colores que proporcionan alguna información sobre las mismas:

 Las funciones que llevan asociado este led de color verde intenso son funciones de Etapa. Tienen un tipo predefinido de retorno como “void” y no usan parámetros. Pueden usarse como acciones de etapa o propiedades de secuenciador.

 Las funciones con este símbolo amarillo son de tipo “transición”, tienen un tipo de retorno predefinido a “bool” y no llevan parámetros. Pueden usarse como evaluaciones de transiciones.

 Las funciones con este led rojo son funciones de tipo “Custom”. No tienen tipo ni parámetros predefinidos, por lo que el usuario debe establecerlos. Solo pueden ser usadas para invocarlas desde otras funciones.

En el caso de la edición de código de un secuenciador, estos mismos iconos también pueden aparecer pero en un tono más claro y “tachados” por líneas del mismo color. Esto indicaría que el secuenciador “hereda” estas funciones de su secuenciador padre, y que por lo tanto no debería editarlas desde este punto. Si se intenta editar una de estas funciones, aparecerá el siguiente aviso:

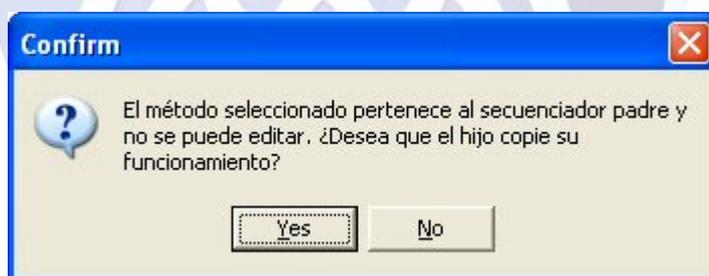


Ilustración 28: Aviso de edición de código heredado

Manifestando que las funciones heredadas no deben ser manipuladas desde los secuenciadores hijos. Si contestamos que no, la función aparecerá, pero estará en modo solo-lectura, y no nos será posible cambiar nada de su código. En cambio, si contestamos que si al diálogo, estaremos realizando una sobrecarga de esa función para el secuenciador. Para una explicación más extensa sobre este concepto, véase el capítulo sobre edición de secuenciadores.

Mezcla de Proyectos

No es inusual que más de una persona trabaje a la vez en un proyecto concreto. Habitualmente, varias personas trabajaran en partes distintas del proyecto, llegando un momento en que tendrán que poner en común todas las partes desarrolladas por separado. En este punto es donde entra en juego la Mezcla

de Proyectos. Su cometido es integrar partes de varios proyectos en el proyecto actualmente en edición, para lograr un trabajo en equipo más efectivo.

Para acceder a este plugin, una vez tenemos abierto un proyecto en el que queremos integrar partes de otros proyectos, basta con pulsar el botón  de la barra de tareas, o bien a través del menú “**Fichero** → **Mezclar Proyectos**” del menú principal. Esto hará que aparezca el siguiente formulario:

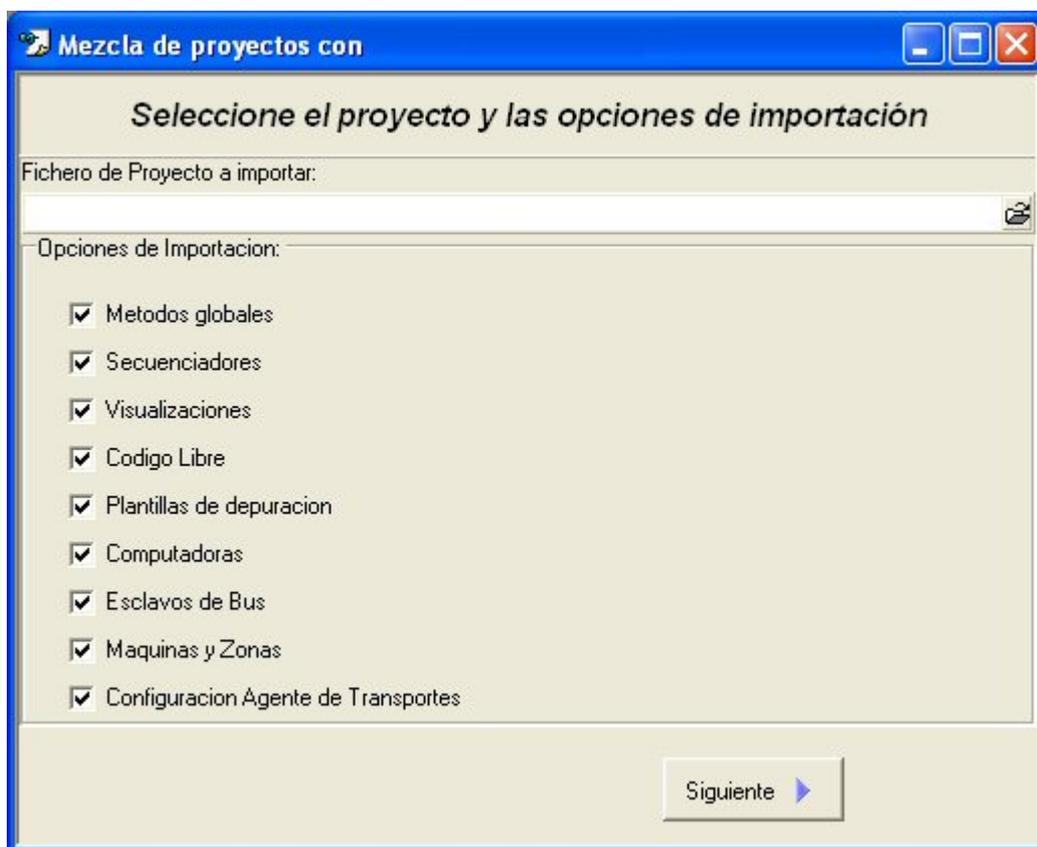


Ilustración 29: Formulario de mezcla de proyectos

En el se puede observar que se nos pregunta por el fichero desde donde leer el proyecto y que partes del proyecto a cargar nos interesa mezclar con el actual. En cuanto hayamos seleccionado el fichero y pulsado las opciones que deseemos integrar, pulsaremos el botón “Siguiente”.

 **Cargando el proyecto,**
Hay que tener en cuenta que al seleccionar el proyecto en el primer formulario y pulsar “siguiente”, el plugin debe de cargar ese proyecto de disco, lo que puede resultar algo lento, dependiendo de la información que contenga.

Esta dinámica se repite durante todo el proceso de mezclado, hasta que aparezca el botón de finalizar. Los formularios que mostramos a continuación aparecen solo si en el primero se ha marcado la opción para mezclar la parte correspondiente.

Mezclar métodos globales

En primer lugar tenemos la opción de mezclar los métodos globales del proyecto a importar.

En este caso la mezcla se realiza como sigue: todos los métodos seleccionados serán **añadidos** al proyecto actual. Si se encontrara algún método con el mismo nombre en el proyecto actual, el nuevo método se añadiría cambiándole el nombre para añadirle el sufijo “imported_”. No se realizarán otros cambios en las asignaciones de métodos a lo largo del proyecto.

En este formulario, como en la mayoría de los siguientes, tendremos 2 botones en la parte superior que nos permitirán seleccionar todos los elementos de la lista, o deseccionarlos todos a la vez.

Mezclar Secuenciadores

La siguiente parte a importar son los secuenciadores. El formulario en este caso será el siguiente.

En este caso, los secuenciadores seleccionados serán **añadidos o sustituirán** a los ya existentes si estos tienen el mismo nombre.

Los secuenciadores aparecen representados en forma de árbol, para reflejar su estructura de dependencia. Hay que notar aquí que en la mezcla de proyectos, existen dependencias que pueden o no ser forzadas a importarse según las elecciones que haga el usuario. Por ejemplo, si importamos un

secuenciador que es hijo de otro que elegimos no importar, durante la importación pueden ocurrir dos cosas:

- 1) El entorno detecta que no tenemos ese secuenciador padre, y como tampoco existe en el proyecto actual, nos informa que debe ser importado también.
- 2) El entorno detecta el secuenciador padre, y encuentra también un posible sustituto del mismo nombre en el proyecto actual. Un cuadro de diálogo nos ofrece la posibilidad de actualizarlo o de dejar el actual. La elección depende del usuario.

La importación de secuenciadores es la parte más delicada de toda la mezcla, puesto que los cambios afectan a la mayoría de los objetos de un proyecto. Concretamente, al sustituir un Secuenciador por otro, ocurren los siguientes eventos:

- Se importa el secuenciador. Todas las funciones del viejo secuenciador son sustituidas por las funciones del nuevo secuenciador. Consecuentemente, los hijos de ese secuenciador ven ahora esas nuevas funciones. Si un hijo usase funciones del viejo secuenciador no presentes en este nuevo secuenciador, se perderían esas funciones.
- Las variables son revisadas. Toda máquina que implementa el secuenciador actualizado sufre un chequeo de sus asociaciones entre variables y parámetros. Si el tipo de un parámetro no coincide tras la importación, esa asociación se rompe. Si un parámetro desaparece, también desaparece sus asociaciones. Si aparecen nuevos parámetros, estos quedan pendientes de asignar.

Importar Visualizaciones

La siguiente parte a importar son las visualizaciones.

En este caso, al igual que con los métodos globales, las visualizaciones seleccionadas son añadidas al proyecto actual. Si existiera alguna otra con el mismo nombre, las nuevas se renombrarían añadiéndoles el prefijo “Imported_”.

 **Visualizaciones no idénticas,** Si importamos una visualización con el fin de actualizar una ya existente, hemos de procurar borrar a continuación la vieja visualización, por dos motivos: Reduciremos el tamaño del proyecto y evitaremos confusión al compilar, dado que la información de anteriores/posteriores viene de las visualizaciones. Por esta última razón, también es conveniente abrir la nueva visualización y ponerla en modo edición y volver a modo runtime antes de realizar una compilación tras una mezcla.

DESIGNER 3.0

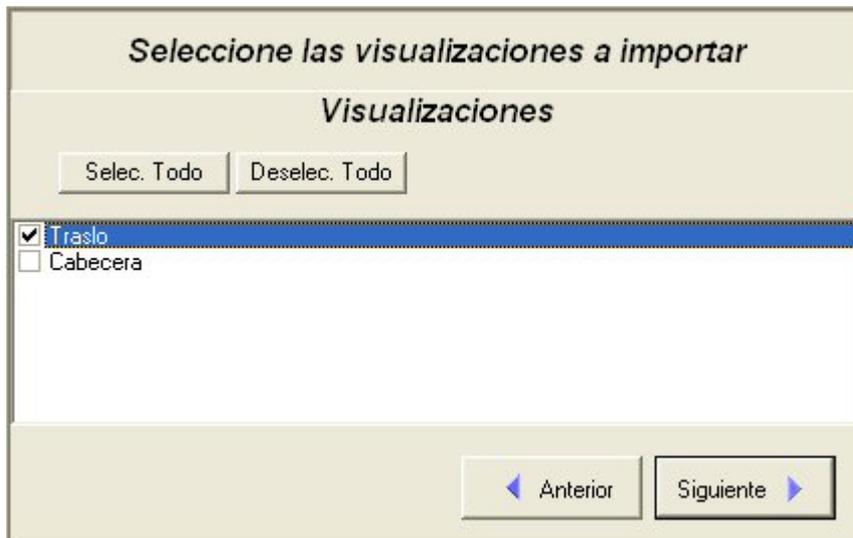


Ilustración 30: Mezclando visualizaciones.

Importar código Libre

La siguiente parte a mezclar, si ha sido seleccionada, será la opción de importar el código libre. Este es un bloque de código monolítico, así que no hay ninguna opción más que la de importarlo o no, no existiendo más formularios. Si se ha marcado para mezclarse, el código libre del proyecto a mezclar **se añadirá** al final del código libre del proyecto actual.

Importar Plantillas de Depuración

Se pueden mezclar también las plantillas de depuración contenidas en el otro proyecto, que se nos mostrarán en un formulario similar al de la siguiente figura:

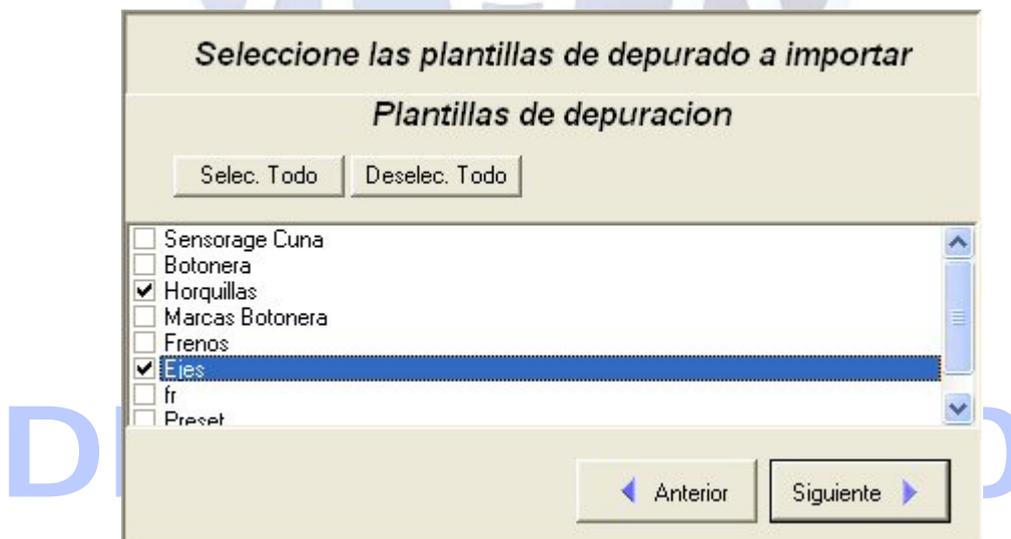


Ilustración 31: Mezclando plantillas de depurado

Hay que tener en cuenta que las plantillas van ligadas a las variables del otro proyecto, por lo que si no importamos las mismas, y estas no existen en el proyecto actual, las plantillas importadas quedarán

invalidadas. Las plantillas **se añaden** a las actuales. Si existe alguna plantilla con el mismo nombre, la nueva se renombra añadiendo el prefijo “Imported_”.

Importar Computadoras

Los computadores contenidos en otro proyecto se pueden añadir al actual a través del siguiente formulario de selección:

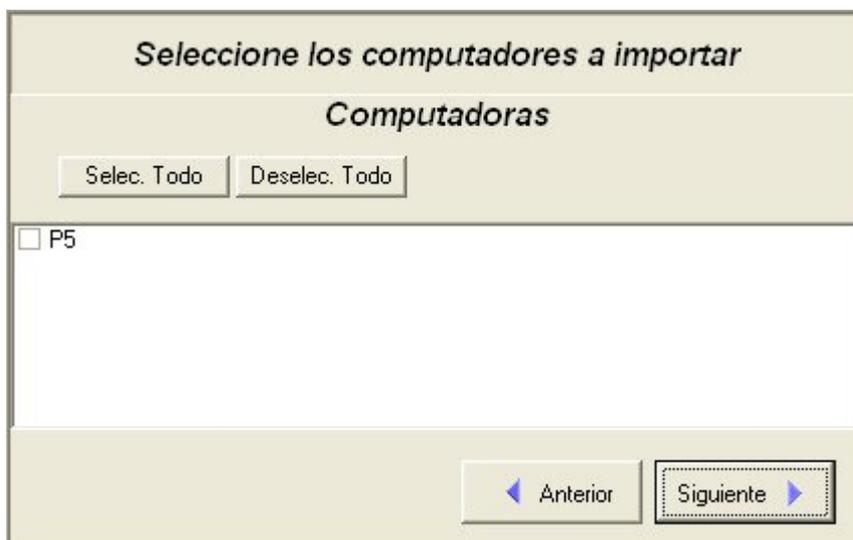


Ilustración 32: Mezclando computadoras

La importación de computadoras es un paso previo imprescindible si pretendemos importar también esclavos de bus: Solo podremos importar esclavos de bus cuyas computadoras hayamos importado también previamente. El resto aparecerá desactivado y no podrá ser seleccionado.

Las computadoras **se añaden o sustituyen** a las ya existentes. Hay que tener en cuenta un efecto secundario de esto: **las computadoras importadas no tienen máquinas asociadas**. Por lo que habrá que asociarles las que sean necesarias. Esto es valido incluso para el caso de que la computadora importada ya existiese previamente.

Las computadoras importadas arrastran automáticamente las tarjetas que lleven definidas al proyecto mezclado, y posibilitan la importación de los módulos de bus definidos en ellas.

Al igual que con los secuenciadores, la importación de una computadora puede tener efectos negativos si no se tiene cuidado cuando se sustituye otra ya existente:

Si la nueva computadora tiene menos tarjetas que la antigua (o las tarjetas han sido renombradas), y no se importa el bus correctamente, los módulos de bus pueden quedar desasociados (sin tarjeta).

Si quedan módulos desasociados del bus, algunas variables de tipo PLC del simbólico pueden perder la referencia del desplazamiento. Esto provocara fallos en tiempo de compilación.

De la misma forma, si las tarjetas importadas cambian su configuración de tamaño de entradas/salidas, los módulos de bus pueden acabar sobrepasando el tamaño asignado de entradas salidas de la tarjeta donde están definidos. Esto provoca fallos que son detectados al comprobar la edición de bus.

Importar Esclavos de Bus

El paso siguiente a importar son los módulos de bus, tal y como se observa en la siguiente figura:

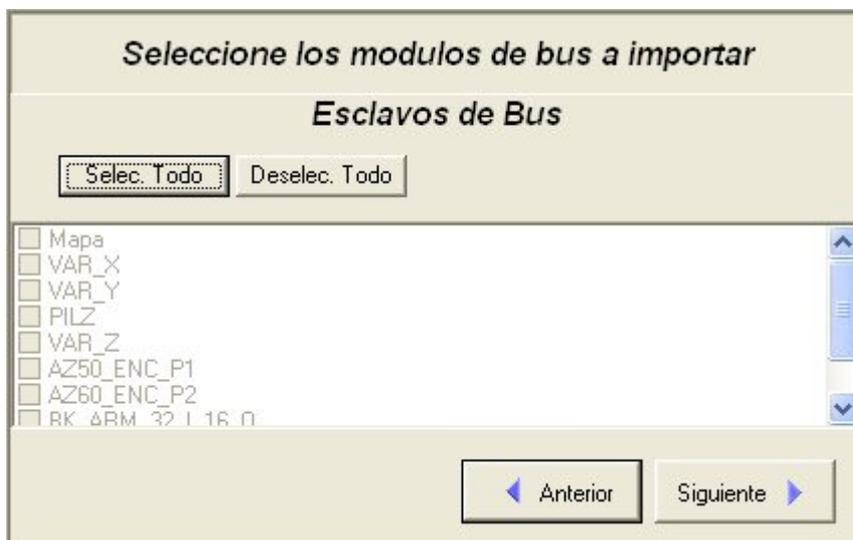


Ilustración 33: Mezclando esclavos de bus

En esta ocasión hay que notar que no podemos seleccionar todos los módulos que queramos, sino solo los que pertenezcan a computadoras que previamente hayamos elegido para importar. Si no hemos seleccionado ninguna, no podremos seleccionar tampoco ningún módulo.

Los módulos seleccionados *se añaden o sustituyen* a los ya existentes. La sustitución se realiza a nivel de que se actualizan los tamaños de las entradas y salidas de los elementos existentes por los de las importadas. También cambia la tarjeta a la que pertenece, si se diera el caso, sin embargo, la posición no es actualizada, con lo que importar una secuencia de módulos no garantiza que terminen con el mismo orden con el que se importan.

No es posible realizar esta “actualización” porque en el caso de seleccionar varios módulos al azar, sería imposible reasignar el nuevo orden y hacerlo “casar” con los módulos ya existentes. Por lo tanto, al abrir una importación de este tipo, observaremos un “desorden” aparente del bus, que se corregirá en cuanto realicemos cualquier operación. Las implicaciones que ello conlleva es que se pierde la información de orden en la importación. Es por ello que utilizar la mezcla de proyectos para remapear el bus no es posible (el resultado será una mezcla caótica de ambos).

Tras cualquier importación de elementos del bus, es aconsejable abrir el editor de Bus y comprobar el nuevo listado de los módulos.

Importar Máquinas y Zonas

Junto con los secuenciadores, las máquinas son los elementos más susceptibles de ser importados a la hora de poner en común un trabajo de equipo. El formulario que nos permitirá realizarlo se muestra en la siguiente figura:

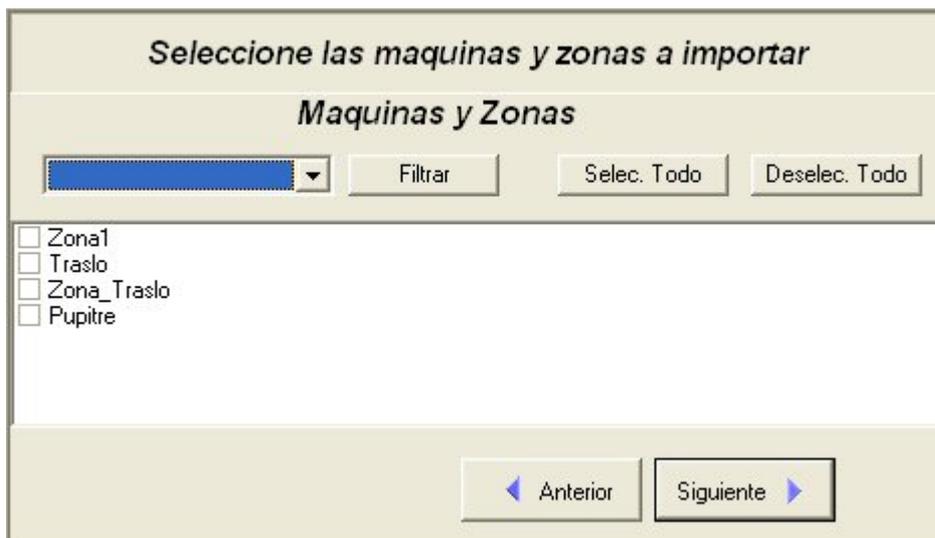


Ilustración 34: Mezclando máquinas y zonas

La parte superior tiene ahora un nuevo aspecto, con un combo que muestra la lista de computadores del proyecto a importar y un botón que nos permite filtrar la lista por el computador, permitiendo elegir así todas las asignadas a un computador.

No es necesario, sin embargo, que dicho computador haya sido seleccionado para importarse.

No se pueden importar variables globales.

Las variables a importar se **añaden o sustituyen** a las ya existentes. Las asociaciones de parámetros existentes intentan conservarse dentro de lo posible. Por ejemplo, si en la nueva máquina desaparecer la variable asociada a un parámetro, este queda desligado en la máquina resultante tras la mezcla. Por contra, si el parámetro estaba ligado a otro parámetro nuevo, se conserva esta nueva asociación.

Si al importar una máquina el entorno detecta que el secuenciador que implementa (o uno con el mismo nombre) no se encuentra en el proyecto, un cuadro de diálogo nos indicará, como nota informativa, que es necesario importar también dicho secuenciador, que será incluido automáticamente en la lista de secuenciadores a importar. Por contra, si dicho secuenciador ya existe, y no ha sido seleccionado para importar, la nueva máquina sufre un chequeo buscando asociaciones que no resulten válidas tras la importación.

De forma similar, las variables de tipo Machine dependen de la zona a la que están asociadas, por lo que si dicha zona no es importada, se importa automáticamente, salvo el caso de que ya existiera una con el mismo nombre. En este último caso, la máquina vuelve a sufrir una nueva comprobación en busca de asociaciones de parámetros de zona que dejan de ser válidas por las diferencias entre ambos proyectos.

Fragmentación de memoria, A fin de evitar posibles solapamientos, las variables de tipo shared importadas sufren una reasignación de memoria. Esto implica que si ya existían, la memoria que ya ocupaban queda libre, produciendo un efecto conocido como "Fragmentación de memoria". Este efecto puede agotar la memoria disponible para variables Shared en poco tiempo, por lo que siempre, tras una importación, es aconsejable abrir la edición del simbólico y realizar un remapeo de variables shared, que provoca la compactación de la memoria.

Hay que ser cuidadosos también con los esclavos de bus: si las variables que importamos contienen variables de tipo PLC cuyos módulos de bus no existan en el proyecto actual, y no importamos los

nuevos módulos, las variables PLC quedarán desligadas de dichos módulos, lo que evitará que el proyecto pueda compilarse con éxito.

Importar la configuración del Agente de Transportes

La última posibilidad de mezcla es la parte de la configuración del agente de transportes. Se puede ver a continuación el formulario para realizarla:

Ilustración 35: Mezclando la configuración del agente de transportes

Podremos seleccionar por separado estaciones, trayectorias y ordenes a importar. Dicha configuración **se añade** a la actual. Si algún elemento de los anteriores ya existiera, el nuevo se renombraría añadiéndole el prefijo “Imported_”.

La lista de errores sólo da opción a importarse completamente, siendo sus elementos añadidos a la lista de errores actuales. Si los errores tienen el mismo número, aparecerán duplicados (salvo que se active la marca “Sobrescribir códigos de error”) y dicha configuración no podrá cargarse en base de datos. El usuario debería seleccionar los que le convengan y borrar los que no le interesen desde el editor de la

configuración del Agente de Transportes. Si se pulsa la marca de sobreescritura, se actualizarán todos los códigos de error repetidos con los nuevos valores que se encuentren.



DESIGNER 3.0

Secuenciadores

Los secuenciadores son el corazón del sistema Galileo ya que con ellos se definen los comportamientos de los diferentes elementos a controlar. Es por eso que su edición es sin duda el paso más importante del desarrollo y que requiere de más pericia por parte del usuario.

Un secuenciador es un autómata finito y determinista por definición. Sin embargo, dentro del sistema Galileo, los secuenciadores extienden esta definición para añadirle diversas propiedades que aumentan sus posibilidades hasta el punto que permiten modelar comportamientos de gran complejidad.

Empezaremos con una serie de definiciones:

Un **Secuenciador** supone la descripción genérica de un elemento que sigue un patrón de comportamiento cíclico y variable. La potencia de este concepto estriba en definir un patrón que con un único código se podrá aprovechar para definir el comportamiento de todas aquellas máquinas que sigan la misma pauta.

Modos de funcionamiento

Un secuenciador puede tener varios **Modos de funcionamiento**, con diferentes prioridades y selectores, que le permiten cambiar su forma de actuar en cuando cambien ciertas condiciones de entrada. Esta posibilidad de cambio le permite adaptarse a situaciones completamente distintas sin necesidad de complicar el grafo hasta niveles incompresibles. Cada modo de funcionamiento tiene sus propias Etapas y Transiciones y se muestra como un grafo separado del resto. Genéricamente, se pueden considerar los modos de funcionamiento como graficets independientes del mismo secuenciador.

📖 Modos de Funcionamiento, En las versiones 2.x, los secuenciadores sólo tenían un único modo de funcionamiento. Cualquier otro modo de funcionamiento era simulado mediante Etapas ANY. Ahora las etapas ANY son los llamados Modos de Funcionamiento por Interrupción, y funcionan del mismo modo.

Elementos de un graficet.

Una **Etap**a consiste en un estado en el que una acción o acciones se están desarrollando.

Las etapas se representan gráficamente como un rectángulo de bordes redondeados y pueden tener hasta 4 acciones asociadas a ellas, tal y como se observa en la Ilustración 36 .



Ilustración 36: Representación de una etapa.

En una **Transición** se definen las condiciones necesarias para el paso de una etapa a otra.

Una transición indica la posibilidad de evolución entre etapas. El franqueo de una transición provoca el paso de una etapa a otra. Una transición entre dos etapas se representa mediante una línea perpendicular a la base de los rectángulos que representan las etapas que se unen, también lleva una línea de menor tamaño, paralela a dichas bases. A cada transición va asociada una condición que puede ser evaluada a verdadero o falso, además, y de forma opcional, se le puede asociar una Acción Anterior.

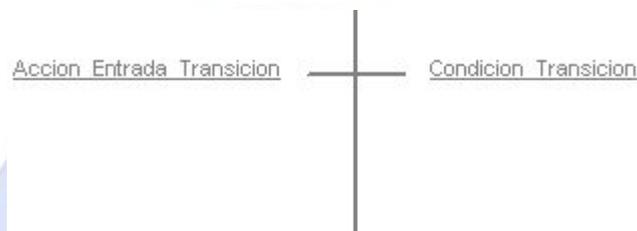


Ilustración 37: Representación de una transición.

Las etapas pueden tener un atributo extra que les señala una característica especial, estos atributos son:

- **STEP:** Etapa normal, sin más características.
- **INITIAL_STEP:** Etapa inicial, indica la etapa por defecto del grafo y a donde se reiniciará cuando termine un ciclo o sea inicializada por el usuario.
- **WAIT_STEP:** Etapa de espera. Significa que dicha etapa no se activa hasta que se cumplen todas las transiciones que llegan a ella.
- **FINAL_STEP:** Etapa final. Al salir de esta etapa, el grafo de ejecución salta de nuevo a la etapa definida como INITIAL_STEP.
- **WAIT_FINAL_STEP:** Combinación de una etapa de espera que además es final. Es decir, no se activará hasta que se cumplan todas las transiciones que llegan a ella, y al cumplirse su transición de salida, el grafo en ejecución regresaría a la etapa definida como inicial.

Cada modo de funcionamiento debe de tener una y solo una etapa definida como inicial, y una y solo una etapa definida como final (puede ser una FINAL_STEP o una WAIT_FINAL_STEP).

Además de este atributo, las etapas pueden ser de 4 tipos diferentes, en función del número de transiciones de entrada/salida que acepten:

A una etapa **Pipe** (de color azul) únicamente le entra una transición y le sale otra.

Una etapa **Fuente** (color verde) recibe una sola transición mientras que de ella puede salir más de una transición.

En una etapa **Sumidero** (color rojo) puede entrar más de una transición pero únicamente puede salir una transición de la misma.

Una etapa **Concentrador** (color marrón) es aquella que puede recibir más de una transición y de la también pueden salir varias transiciones.

El orden de ejecución de las acciones de etapas y transiciones sería el mostrado en la Ilustración 38

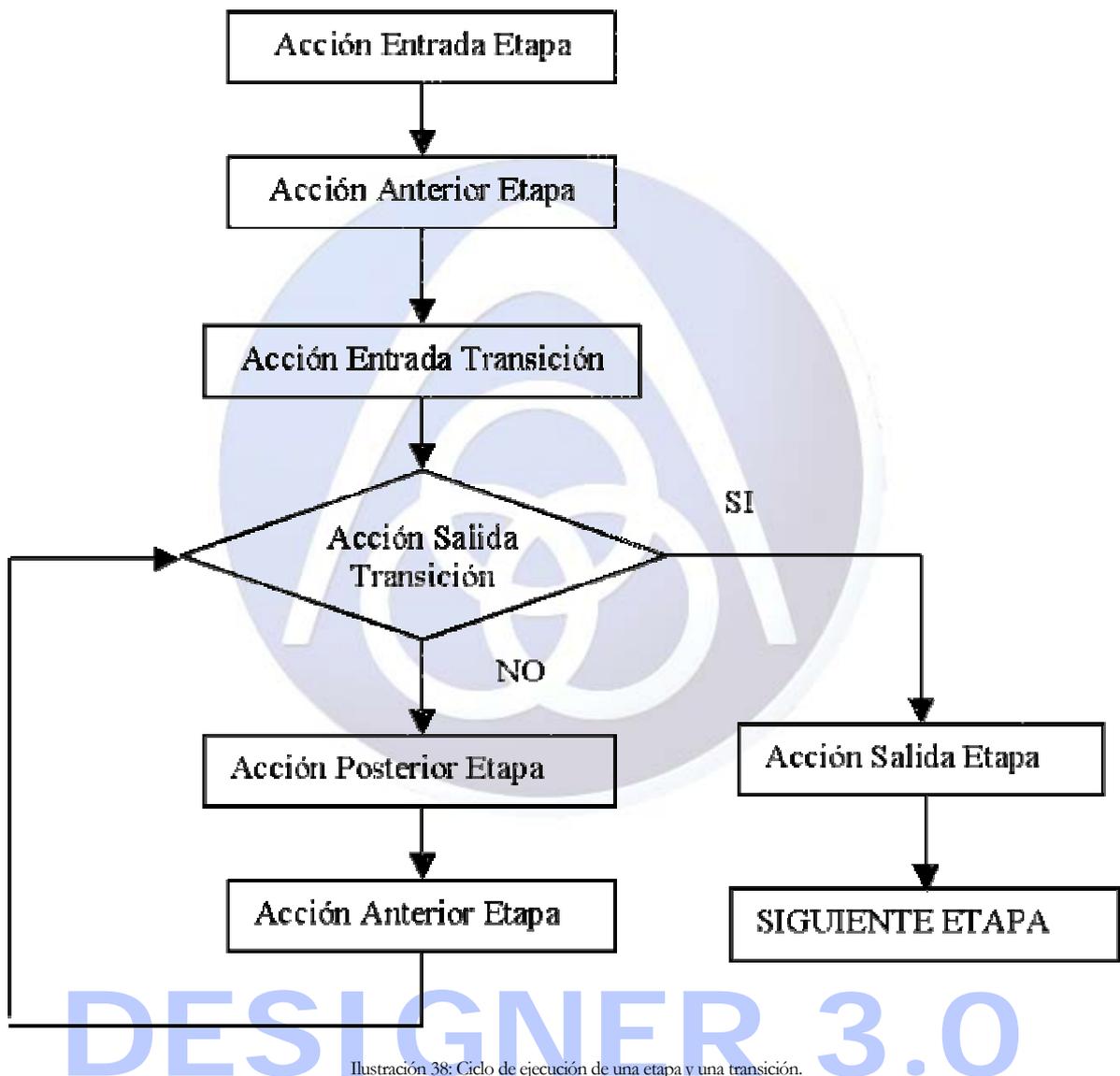


Ilustración 38: Ciclo de ejecución de una etapa y una transición.

Vamos a detenernos un momento en los distintos tipos de acciones mencionadas:

- *Acción de Entrada de Etapa:* Esta acción se ejecuta cuando la etapa en ejecución se activa, bien porque la condición de transición le da paso, bien porque se regresa de un modo de

funcionamiento por interrupción (ANY). Su prototipo será una función que no retorna ningún tipo de valor (void) y no requiere de ningún parámetro para ser llamada.

- *Acción de Salida de Etapa:* Esta función se ejecuta cuando la etapa en cuestión deja paso a la ejecución de cualquier otra etapa, ya sea porque se cumple la transición de salida o porque se cumple la condición de entrada de un modo de funcionamiento por interrupción. Su prototipo será una función que no retorna ningún tipo de valor (void) y no requiere de ningún parámetro para ser llamada.
- *Acción Anterior de Etapa:* En cada ciclo de ejecución Galileo, esta función será llamada antes de evaluar la condición de transición. Su prototipo será una función que no retorna ningún tipo de valor (void) y no requiere de ningún parámetro para ser llamada.
- *Acción Posterior de Etapa:* En cada ciclo de ejecución Galileo, esta función será llamada después de evaluar la condición de transición, siempre que esta última haya resultado negativa, pues en otro caso, no se llegará a ejecutar. Su prototipo será una función que no retorna ningún tipo de valor (void) y no requiere de ningún parámetro para ser llamada.
- *Acción de Entrada de transición:* Esta función es invocada por Galileo en el mismo ciclo que la acción de Entrada a la etapa, justo antes de invocar por primera vez a la condición de transición. En los subsiguientes ciclos no es llamada (salvo en los mismos casos en que se vuelve a llamar a la acción de entrada a la etapa) Su prototipo será una función que no retorna ningún tipo de valor (void) y no requiere de ningún parámetro para ser llamada.
- *Condición de Salida de transición:* Esta es una función que se evalúa en cada ciclo de ejecución. Su prototipo será una función que retorna un booleano indicando si se debe cambiar de etapa (valor **true**) o si no es así (valor **false**). No debe llevar parámetros de ningún tipo.

Los modos de funcionamiento normales llevan asociados un número llamado **Selector** que los identifica y una característica llamada **Prioridad**. En todo secuenciador debe haber al menos un modo de funcionamiento cuyo selector valga 0. Este modo de funcionamiento será el **Modo de Funcionamiento por Defecto** y será el que esté activo si no se especifica otro desde código. Una vez creado el grafo de selector 0, no puede cambiarse dicho valor. En todos los modos de funcionamiento normales, la prioridad debe de valer 0. Los identificadores de selector no pueden repetirse entre los distintos modos de funcionamiento normales.

Además de los modos de funcionamiento normales, existen otros Modos de Funcionamiento por Interrupción (también llamados Etapas ANY), que funcionan de una manera un poco diferente, asemejan en su mayor parte a los modos de funcionamiento normales (por selector), pero existen varias diferencias:

- Su campo prioridad es un número mayor que 0. Dicho número indica también su orden de evaluación: cuanto más alta es la prioridad, antes se evaluarán. Dicha prioridad también debe ser única, no pudiendo repetirse entre distintos modos de funcionamiento por Interrupción.
- Su selector no es usado, así que puede ser 0.
- Sus grafos no tienen Etapas iniciales o finales, ni tampoco transiciones finales. En su lugar, tienen **Transiciones de Entrada ANY** y **Transiciones de Salida RETURN**.

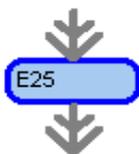
La diferencia fundamental con los modos de funcionamiento normales es que los modos de funcionamiento por interrupción no necesitan activarse por código, sino que se activan automáticamente en cuanto se cumple la condición de su transición de Entrada ANY. Asimismo, también dejan de estar activos en cuando se cumple su transición de Salida RETURN.

Para intentar integrar los dos modos de funcionamiento con el gráfico anterior, imaginemos que antes de cada ciclo de ejecución se evalúan, por orden de prioridad, todas las transiciones de entrada de todos los modos de funcionamiento por prioridad. Si ninguna de estas transición se cumple, seguiremos con el funcionamiento visto para los modos de funcionamiento por selector. Si una de esas transiciones se cumpliera, ocurriría que:

- El modo de funcionamiento actual se detendría.
- Se guarda el estado actual del modo de funcionamiento por selector
- Se entra en el modo de funcionamiento por interrupción.

Dentro de este modo de funcionamiento, el grafo de prioridad se ejecuta “normalmente”, pero siempre con la salvedad que se siguen evaluando las transiciones de entrada de los modos de funcionamiento por prioridad cuya prioridad sea más alta que la del actual. Si alguna de estas transiciones se activara, volvería a producirse un cambio, idéntico al anterior, con la excepción de que el estado del grafo no se salva en este caso, dado que los funcionamientos por interrupción no “recuerdan” su estado.

Las transiciones de entrada de un modo de funcionamiento por interrupción, se representan de la siguiente manera:

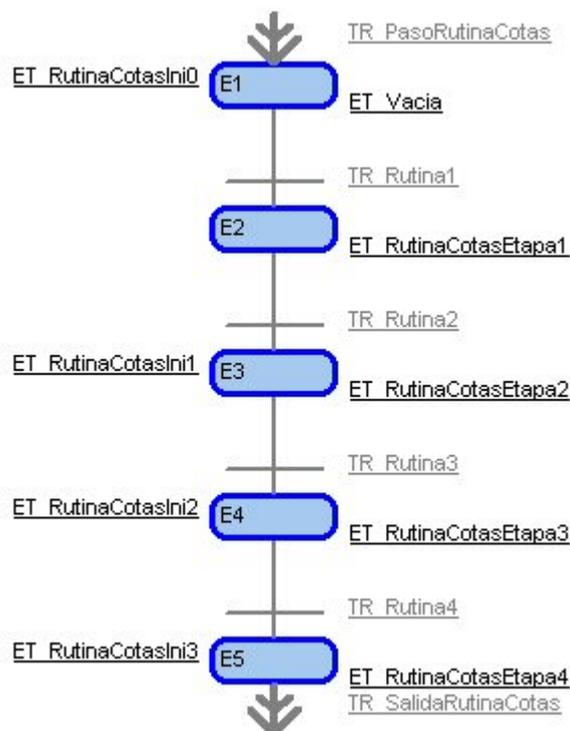


En este caso, que es el más común, la interrupción consta de una sola etapa. Las transiciones de entrada y salida se representan por una flecha de doble cabeza. En las propiedades de transición, veríamos que la transición de entrada tiene de tipo ANY y la de salida de tipo “Return”. La etapa es, a todas luces, una etapa normal.

Un caso algo más complejo podría ser el siguiente:

DESIGNER 3.0

Donde el funcionamiento por interrupción realiza una serie de tareas de forma secuencial, de forma semejante a un modo de funcionamiento por selector, pero con las salvedades antes comentadas.



Herencia

Otro de los conceptos nuevos introducidos en la versión 3 con respecto a los secuenciadores es el de Herencia. Los secuenciadores ahora pueden compartir propiedades entre ellos, basándose en una relación de herencia directa. Dicha relación es transitiva, lo cual implica que los grados de herencia pueden ser múltiples.

Antes de seguir explicando la herencia, es necesario introducir un nuevo concepto: los **Parámetros**. Estos son abstracciones de tipos de variables que permiten implementar una plantilla a la que las máquinas que implementen un secuenciador deberán adaptarse a fin de poder usarlo.

Los parámetros especifican el tipo básico de variable, pero no ninguno de sus modificadores. Es decir, podremos definir parámetros de tipos Bit, Byte, Word, Dword, String o Componentes, independientemente de si son de tipo Shared, PLC, de entradas o de Salidas.

Los parámetros se subdividen sin embargo en 2 categorías: parámetros normales y parámetros de zona. Los parámetros normales afectan a una instancia del secuenciador (y en el código de las funciones deberían ser usados mediante la fórmula **this.nombre_parametro**), mientras que los parámetros de zona pueden afectar a varias instancias del secuenciador a la vez (en el código deberían aparecer como **this.zone.nombre_parametro_zona**). El concepto de zona se explicará con más detalle durante la explicación del Simbólico en el capítulo 7.

Una correcta realización de un secuenciador implicaría que en su código no aparecen referencias a variables del simbólico, sino a sus parámetros. Esto puede comprobarse fácilmente si se logra compilar el secuenciador sin especificar ninguna máquina, tal y como se explicó en el apartado dedicado al plugin de compilación, en el capítulo 5.

Al contrario que los modos de funcionamiento, los parámetros y las funciones de un secuenciador son heredables. Es decir, que un secuenciador hijo tiene acceso a todos los parámetros y funciones definidos en el padre o uno de sus ancestros.

Sobrecarga de funciones en secuenciadores

La sobrecarga de funciones en los secuenciadores es una herramienta poderosa, pero debe usarse con cuidado. La sobrecarga en este ámbito se puede definir como una función que tiene distinto código dependiendo del contexto en que se use.

Concretando más esta definición inicial, consiste en crear funciones en un secuenciador que ya han sido previamente declaradas en el secuenciador padre (o alguno de sus ancestros). Estas nuevas funciones, con el mismo prototipo que las de su padre, pueden contener código completamente diferente, y a su vez “ocultan” la visión de la función que sobrecargan para ese secuenciador y para todos sus descendientes. Esto permite especializar los descendientes de un secuenciador sin perder la claridad de concepto que ya se tenía al conocer al secuenciador base.

Al contrario que las funciones, los parámetros no se sobrecargan, es decir, no pueden cambiar de significado entre un padre y sus descendientes.

En las representaciones de los graficet, se distingue si una función usada es heredada o no en que las funciones propias siempre se dibujan subrayadas, mientras que las heredadas no lo están.

Interfaz de diseño

Tras introducir las definiciones básicas, pasamos ahora a explicar como trabajar con el formulario de edición de secuenciadores. Cuando abrimos una ventana de edición de secuenciadores por primera vez, observamos una vista similar a la siguiente:



DESIGNER 3.0

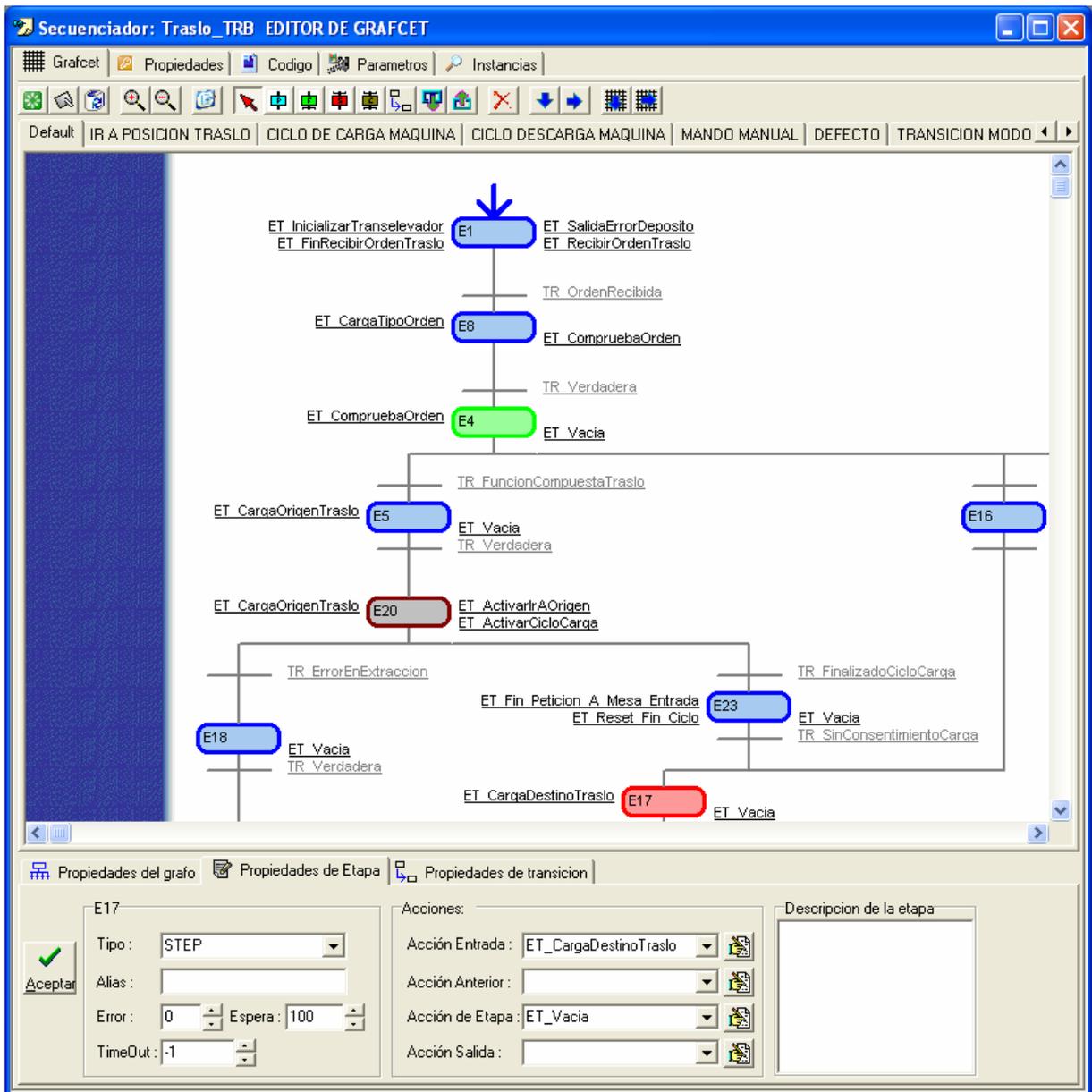


Ilustración 39: Plugin de edición de secuenciadores.

Como se ve, el formulario está dividido en 5 pestañas, cada una muestra una faceta concreta de la definición de un secuenciador:

- Grafcet** Permite definir los modos de funcionamiento del secuenciador, apareciendo los diferentes grafkets separados en pestañas dentro de esta.
- Propiedades** Permite establecer parámetros inherentes al secuenciador, así como información sobre el mismo. Desde aquí también se puede controlar la herencia y algunas funciones de depuración para el secuenciador.
- Código** Desde esta pestaña podemos ver/modificar/eliminar las funciones contenidas en el secuenciador. Dentro de ella veremos un editor de código muy similar al editor de

código del proyecto, y su funcionamiento es básicamente el mismo. Las diferencias puntuales se explicarán más adelante.

 **Parámetros** Esta pestaña nos da acceso al formulario desde donde se pueden crear, modificar o eliminar los parámetros del secuenciador.

 **Instancias** La última pestaña no es de diseño propiamente, sino de consulta. Nos permite tener en cada momento un listado de todas las máquinas que usan el secuenciador en edición en este proyecto. Si hacemos doble click con el cursor del ratón sobre cualquiera de los elementos de esta lista, el editor del simbólico entrará en escena y se mostrará apuntando directamente a la definición de la máquina señalada.

A continuación pasaremos a describir más concienzudamente cada uno de estos elementos, así como la barra de herramientas de la parte superior del formulario.

La solapa de Grafcet

La pestaña de grafcet permite definir los modos de funcionamiento. Inicialmente sólo se mostrará una barra de herramientas, puesto que no existe ningún modo de funcionamiento predefinido para los nuevos secuenciadores.



Ilustración 40: La paleta de herramientas de edición del grafcet.

 Crea un nuevo modo de funcionamiento. Al pulsarlo un cuadro de diálogo aparecerá solicitándonos un nombre para el nuevo modo de funcionamiento y el tipo del mismo, tal y como se aprecia en la siguiente figura:

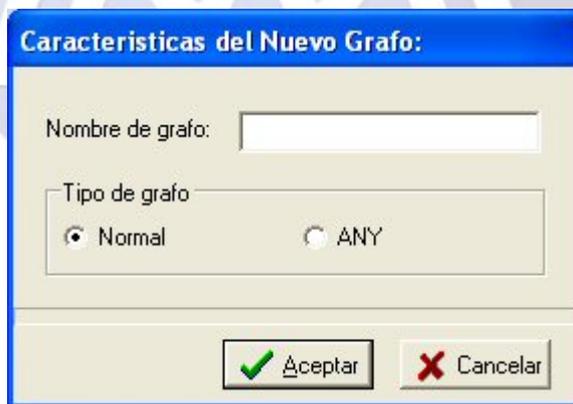


Ilustración 41: Diálogo de creación de nuevo grafo

 Al pulsarlo solicitamos que el modo de funcionamiento mostrado actualmente sea eliminado del secuenciador. Si confirmamos este punto en el diálogo que aparecerá a continuación, dicho modo desaparecerá del secuenciador. La única excepción a esta regla es el modo de funcionamiento por defecto (que tiene selector y prioridad 0), que no puede ser nunca eliminado.

 Permite cambiarle el nombre al modo de funcionamiento. El nuevo nombre no puede estar siendo usado por otro modo de funcionamiento del mismo secuenciador.

 Permite realizar una operación de acercamiento sobre el graficet mostrado. Este cambio es solo temporal y a nivel estético, puesto que no afecta nada más que a la manera de ver el grafo. Cada vez que se pulse este botón, las etapas pasarán a ocupar el doble del tamaño, hasta un tamaño máximo de 4 veces el inicial.

 Este icono causa el efecto inverso al anterior. Al pulsarlo los elementos del graficet actual pasarán a dibujarse a la mitad del tamaño actual (hasta un mínimo). Puede resultar útil si un grafo tiene muchos elementos y resulta difícil obtener una imagen global del mismo.

 Este icono realiza una tarea de “reagrupamiento” del graficet mostrado que intenta colocar las etapas de una forma ordenada. No obstante, se trata de una ordenación un tanto mecánica, puesto que no tiene en cuenta el tamaño de los textos, así que los resultados pueden no ser del gusto del usuario.

Los siguientes iconos sirven para actuar ya de forma directa sobre el dibujado del graficet:

 Pone el cursor en modo “selección”. En este modo se pueden seleccionar los diferentes elementos del graficet para ver sus propiedades en la parte inferior del panel.

 Insertar una etapa “Pipe”. Permite insertar una etapa de este tipo en el graficet.

 Insertar una etapa “Source”. Permite insertar una etapa de este tipo en el graficet.

 Insertar una etapa “Sink”. Permite insertar una etapa de este tipo en el graficet.

 Insertar una etapa “Hub”. Permite insertar una etapa de este tipo en el graficet.

 Insertar una transición. Permite insertar una transición nueva entre dos etapas.

 Inserta una transición ANY de entrada en el graficet.

 Permite insertar una transición Any de retorno en el graficet.

 Este icono permite eliminar un objeto del graficet, ya sea etapa o transición. Para hacer el borrado basta con seleccionar el elemento a borrar y a continuación pulsar la tecla “Supr” o pulsar con el ratón sobre este icono. Un diálogo nos solicitará confirmación avisando de las posibles consecuencias del borrado del elemento en cuestión. Por ejemplo, borrar una etapa también implica borrar las transiciones que salen o entran en ella

 Desplaza los elementos del graficet hacia abajo. El desplazamiento se realiza a partir del último elemento seleccionado. Todas las etapas por debajo de ese punto se moverán dos bloques hacia abajo.

 Desplaza los elementos del graficet hacia la derecha. El desplazamiento se realiza a partir del último elemento seleccionado. Todos los elementos a partir de ese punto son desplazados hacia la derecha dos bloques.

 Permite insertar espacio en el área de edición del graficet. Se añaden cinco bloques en blanco por debajo del último. Nos permite hacer crecer este área si el graficet es muy alto.

 Permite insertar espacio en el área de edición del grafcet. Se añaden 3 bloques en blanco por la parte derecha del grafcet. Esto nos permite editar más cómodamente si el grafcet es muy ancho.

Estos botones y el área de dibujo componen la mitad del área de trabajo de un grafcet. La otra parte es el panel situado en la parte inferior, donde se muestran las características de los elementos que se van seleccionando.

Al pulsar sobre la pestaña de un grafcet, se muestran las propiedades del mismo, que básicamente, son su selector y la prioridad que tiene.

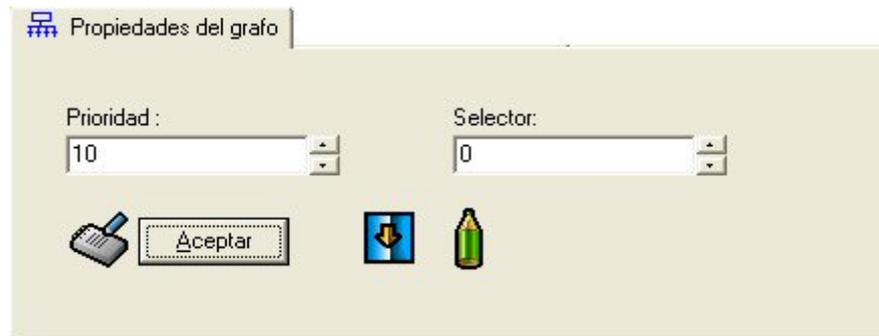


Ilustración 42: Propiedades de un grafo

Aunque ambos pueden tener valores, lo habitual es que uno valga 0. Si es un modo de comportamiento por prioridad, el valor que manda es el de la prioridad (que debe ser mayor que 0), mientras que si es un modo de funcionamiento (su prioridad es 0), el valor que lo define es el del selector, que si tiene un valor 0 indica que es el modo de funcionamiento por defecto.

Los dos iconos que aparecen a la derecha del botón de aceptar, nos dan información sobre el grafo. Concretamente, cuando aparece el primero quiere decir que el grafo es **completo**, mientras el segundo indica que es **usable**. La primera propiedad indica que en el grafo hay tantas transiciones de entrada como de salida, lo que garantiza que no existen ramas muertas. La segunda propiedad indica que todas las transiciones que aparecen tienen una función de transición asociada. Si alguna de estas propiedades no se diera, un mensaje en rojo aparecerá indicando cuales elementos provocan que no se cumplan dichas condiciones.

Si el elemento que hemos seleccionado es una etapa, se nos muestran sus propiedades:

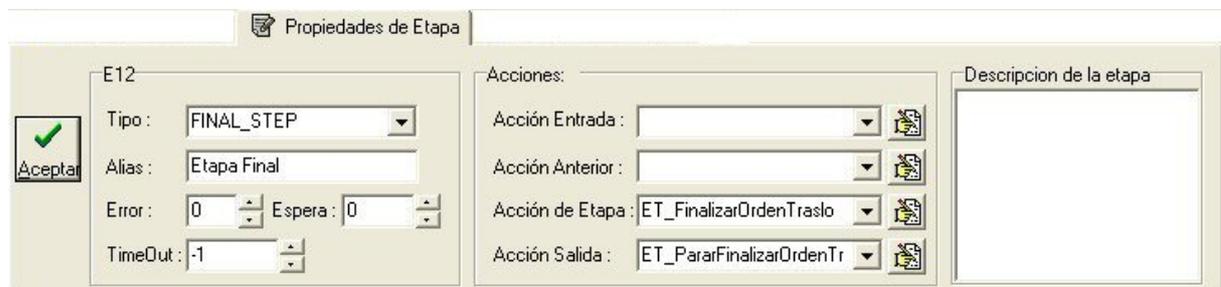


Ilustración 43: Propiedades a especificar para una etapa.

Una vez abierta la ventana que nos permite el diseño del GRAFCET, éste se va construyendo con los elementos disponibles en la paleta situada en la parte superior. Obligatoriamente hay que definir una

etapa del graficet como **INITIAL_STEP** y otra como **FINAL_STEP**, características que se deben fijar dentro de las propiedades de dichas etapas.

Mediante el combo “Tipo” se puede cambiar la forma de comportarse de la etapa respecto a sus predecesoras. Hay 4 tipos: **STEP**, **WAIT_STEP**, **INITIAL_STEP**, **FINAL_STEP** o **WAIT_FINAL_STEP**. Antes de poder definir una etapa como **FINAL_STEP** o **WAIT_FINAL_STEP**, es necesario haber definido la etapa inicial (es decir, asignarle como tipo **INITIAL_STEP**). Si cambiamos una etapa y hacemos que su tipo sea **WAIT_STEP** o **WAIT_FINAL_STEP**, también lograremos que todas las transiciones que llegan a ella pasen a ser de tipo AND.

El campo “Alias” es un nemotécnico que podemos asignar a una etapa, a fin de señalarla para que pueda ser referida posteriormente desde el formulario de edición de tracking.

El campo “Espera” sirve para introducir un periodo mínimo de ejecución de la etapa, que implica que aunque la transición de salida se cumpla, no se cambiará a la próxima etapa hasta que la actual haya consumido al menos ese tiempo (medido en milisegundos). Si ya ha transcurrido ese tiempo de espera desde que la etapa se inicio, se cambiaría de forma normal.

El campo “TimeOut” es exactamente el complementario al anterior. Establece un tiempo máximo de ejecución de la etapa en milisegundos. Si transcurrido ese tiempo la etapa aun no ha cambiado, se disparara un error cuyo número será el indicado en el campo “Error”. Este error no es bloqueante, ni implica que se salga de la etapa, sino que es el programador el que debería programar un modo de funcionamiento por interrupción para detectar este caso y actuar en consecuencia. El uso de estos campos, por ejemplo, es muy importante para evitar movimientos continuados de motores que pueden acabar quemándolos.

Mediante los combos de edición podremos establecer el nombre de las funciones que deseamos sean sus diferentes propiedades de entrada, salida, anterior o acción de etapa. Por último, cuando cambiemos alguno o varios de estos parámetros, la modificación sólo se tendrá en cuenta tras pulsar el botón “Aceptar”.

Si lo que se selecciona es una transición, el panel que aparecerá será el que se muestra en la Ilustración 44.

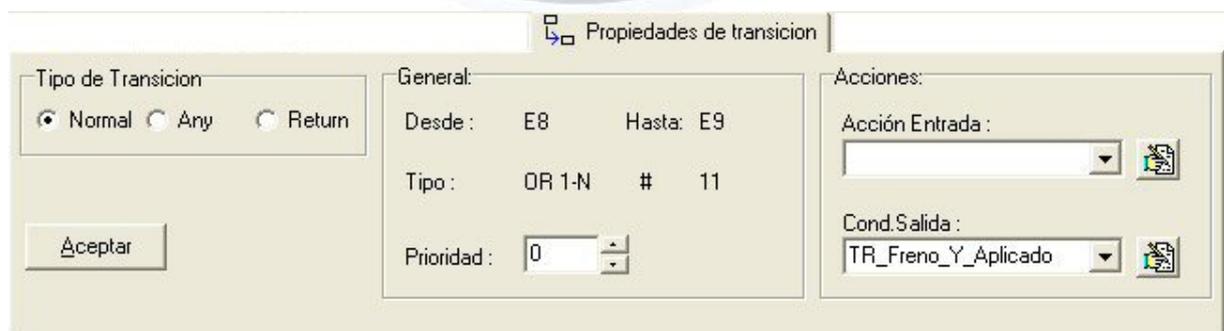


Ilustración 44: Propiedades a especificar para una transición.

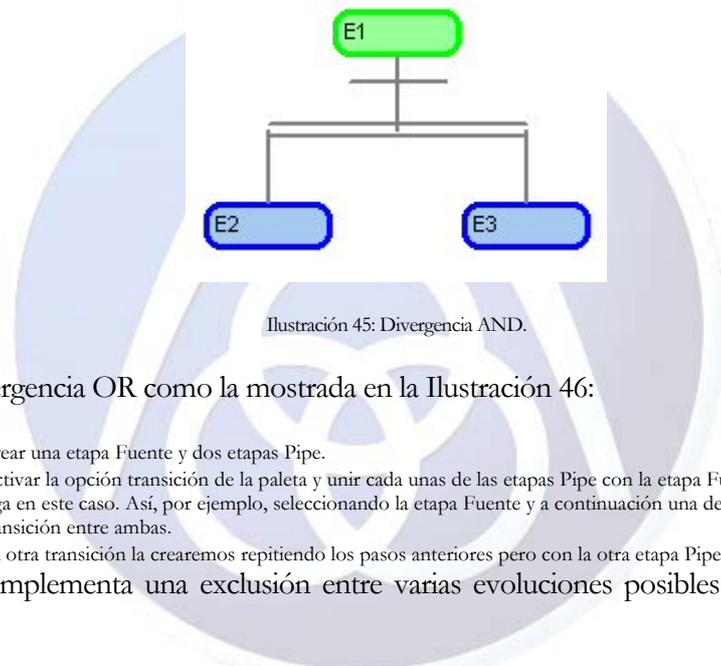
Como se ve, desde las prioridades no se puede cambiar el tipo de la transición (OR o AND), ya que esto se hace directamente cuando se dibuja la transición o incluso la etapa.

El campo “Prioridad” permite establecer una prioridad a la transición que será la que marque su orden de evaluación en el conjunto formado por ella y sus transiciones hermanas. Es decir, para el caso de una divergencia en OR, todas las transiciones hijas de una etapa son evaluadas en el orden especificado por esta prioridad. A menor valor de la prioridad, antes se evaluará. Si una de las transiciones se cumple, el resto de sus hermanas ya no se siguen evaluando (puesto que implica ya un cambio de etapa).

Para fijar una divergencia AND como la mostrada en la Ilustración 45:

- Colocar una etapa Fuente y luego 2 etapas Pipe.
- A continuación se debe crear la transición de la siguiente forma: activar la opción transición de la paleta, pulsar sobre la etapa fuente y luego sobre la Pipe (aparecerá una transición), seguidamente pulsar sobre la transición creada y posteriormente sobre la otra etapa Pipe.

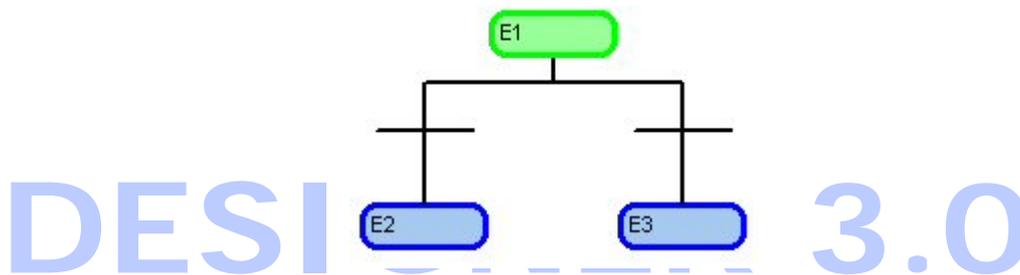
El resultado debe ser una figura como la señalada en la Ilustración 45. Una vez que la transición sea validada, se ejecutarán de forma simultánea las ramas que une.



Para fijar una divergencia OR como la mostrada en la Ilustración 46:

- Crear una etapa Fuente y dos etapas Pipe.
- Activar la opción transición de la paleta y unir cada una de las etapas Pipe con la etapa Fuente, no importa el orden que se siga en este caso. Así, por ejemplo, seleccionando la etapa Fuente y a continuación una de las etapas Pipe, aparecerá la transición entre ambas.
- La otra transición la crearemos repitiendo los pasos anteriores pero con la otra etapa Pipe.

En este caso se implementa una exclusión entre varias evoluciones posibles a partir de una misma etapa.



Para fijar una convergencia OR como la mostrada en la Ilustración 47:

- Crear dos etapas Pipe y una etapa Sumidero.

- Activar la opción transición de la paleta y unir cada una de las etapas Pipe con la etapa Sumidero, no importa el orden que se siga en este caso. Así por ejemplo, seleccionando la etapa Fuente y a continuación una de las etapas Pipe, aparecerá la transición entre ambas.
- La otra transición la crearemos repitiendo los pasos anteriores pero con la otra etapa Pipe.

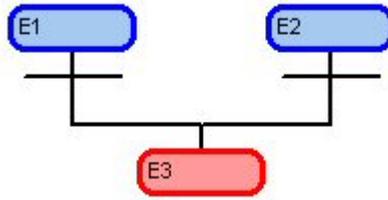


Ilustración 47: Convergencia de transiciones OR.

Para fijar una convergencia AND como la mostrada en la Ilustración 48:

- Crear dos etapas Pipe y una etapa Sumidero.
- Activar la opción transición de la paleta y unir cada etapa Pipe con la etapa Sumidero, no importa el orden que se siga en este caso. Así por ejemplo, seleccionando la etapa Fuente y a continuación una de las etapas Pipe, aparecerá la transición entre ambas.
- La otra transición la crearemos repitiendo los pasos anteriores pero con la otra etapa Pipe.
- A continuación, editando las propiedades de dicha etapa, hay que definirla como **WAIT_STEP**.

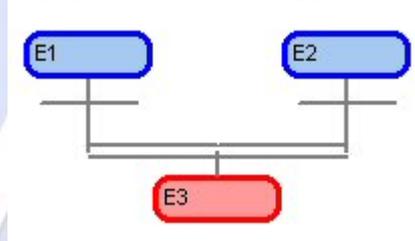


Ilustración 48: Convergencia de transiciones AND.

La pestaña "Propiedades"

La tercera pestaña del plugin de edición de secuenciadores permite definir el secuenciador con un texto descriptivo, como se ve en la Ilustración 49

Junto a esta descripción, también podemos especificar una serie de funciones que actúan siempre sobre el secuenciador, independientemente del modo de funcionamiento activo. Estas son:

Acción Anterior. Especifica una acción que siempre se ejecutará antes de evaluar el estado activo del secuenciador.

Acción Posterior. Especifica una acción que siempre se ejecutará después de evaluar el estado activo del secuenciador.

Arranque frío. Esta función es invocada, para cada máquina que implementa el secuenciador, cuando el Sistema Galileo inicia un arranque frío (inicio del sistema). Típicamente se hacen las inicializaciones en esta función (carga de datos de disco, por ejemplo).

Arranque caliente. Idéntica a la anterior, pero para los arranques en caliente (reinicios del sistema Galileo que no implican una parada del sistema). Su cometido es muy similar a la anterior y puede usarse en los mismos casos, aunque suele realizar una inicialización más ligera que aquella.

Modo de depurado y Trigger de Depurado: Permiten especificar dos funciones, que deben ser declaradas como funciones de tipo transición, que si están definidas gobiernan la ejecución de una máquina en modo depuración. Cuando ambas señales están definidas, la función definida como “Activación de depurado” se ejecuta en cada ciclo de programa, y si devuelve un valor de “true”, eso indica que la máquina se encuentra en modo de depuración, y por lo tanto la máquina no cambiará de estado mientras que la evaluación de la función definida como “Transición de disparo” no resulte en “true”. Cuando se cumpla dicho caso, la máquina actuara como si se hubiesen habilitado las transiciones de salida de las etapas actuales en que se encuentra la máquina, produciendo un cambio de etapa. Es importante resaltar que este modo de depuración solo es válido cuando la máquina se encuentra en el modo por funcionamiento por selector, no cuando está en modos de funcionamiento por prioridad (ANY).

Secuenciador Padre: El Secuenciador padre es el elemento que permite relacionar secuenciadores entre si, dando lugar a la herencia de funciones y parámetros. Al establecer un padre por primera vez o cambiarlo, notaremos como el nodo del secuenciador en el árbol de proyectos se desplaza para reflejar el cambio en la jerarquía que hemos realizado.

Todas estas funciones, así como las de las propiedades de etapas y transiciones del secuenciador, son accesibles desde la tercera solapa de edición de código, simplemente con pulsar el botón que aparece a la derecha de los campos de nombre.

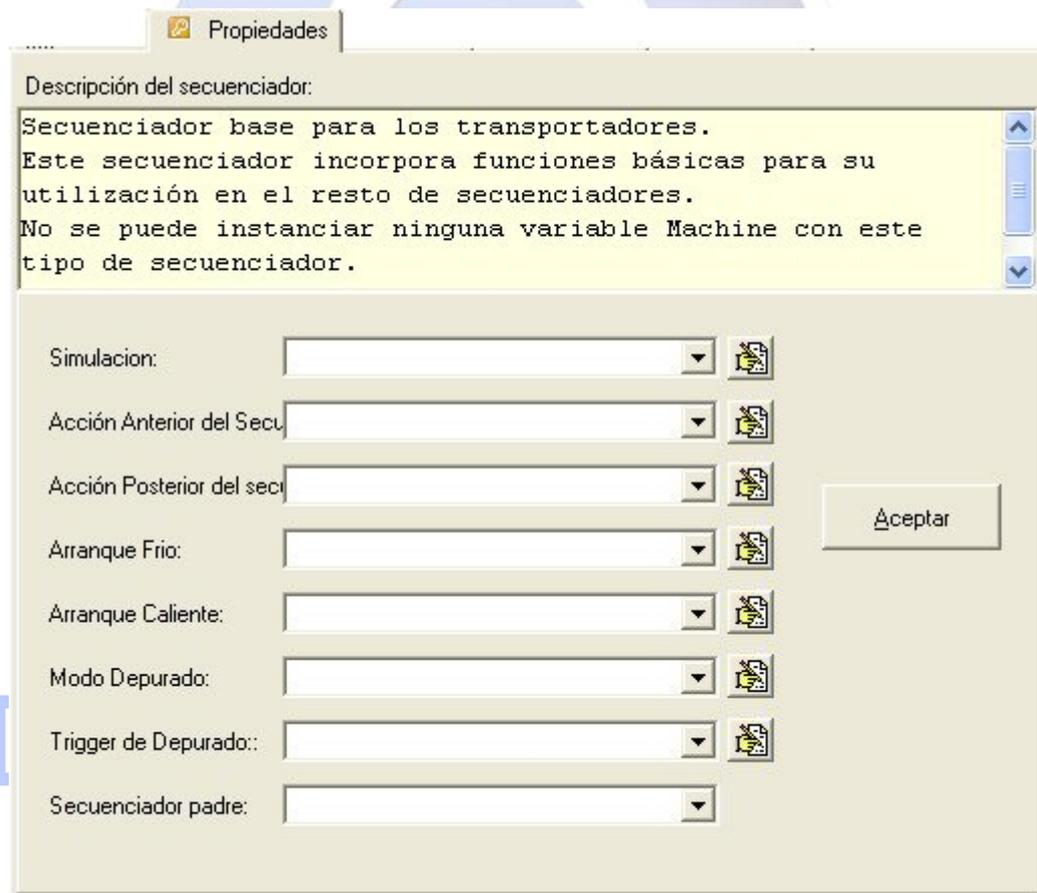


Ilustración 49: Solapa de propiedades genéricas de un secuenciador

La única diferencia aquí es que en los grafos de solo-lectura, el campo “descripción” no aparece, siendo sustituido por un botón que permitirá un acceso directo a la ayuda entregada con la documentación de la versión del repositorio correspondiente.

La pestaña “Código”.

El editor mostrado es prácticamente el mismo que el editor de código del proyecto, con algunas salvedades:

- No existe el concepto de “código libre” para los secuenciadores. Por lo tanto, no se puede editar. Existen, sin embargo, funciones “Custom” que suplen este papel.
- Si una función pertenece al padre, no es posible editarla desde este secuenciador. Se nos ofrece, sin embargo la posibilidad de “sobrecargarla”, tal y como se explicó antes. A partir de que la sobrecarguemos, podremos trabajar con ella de forma normal, ya que ahora sí que pertenece al secuenciador actual.
- Si una función es lo suficientemente general como para considerar que puede ser interesante que el resto de “hermanos” de un secuenciador la puedan necesitar también, tenemos la opción de pasarla al padre, usando el menú del botón derecho del ratón sobre su nombre en el minibrowser de funciones. A partir de ese momento, la función pasa al padre, ya está disponible para todos los secuenciadores “hermanos” del actual, así como sus descendientes.
- En el minibrowser de funciones pueden aparecer iconos levemente diferentes a los anteriores, a fin de diferenciar cuales funciones son propiedad directa del secuenciador o vienen heredadas. Para diferenciarlas el icono de las funciones heredadas aparece siempre con un tono más claro y una línea en diagonal: .

En el resto de aspectos, todos funciona de la misma forma que el formulario de edición de código, tal y como se explicó en el Capítulo 5.

La pestaña “Parámetros”

La tercera parte de la definición de un secuenciador es la edición de sus parámetros. Se nos muestran inicialmente todos los ya definidos, tanto del secuenciador como de su zona. Podemos ver el aspecto de este formulario en la siguiente figura:

DESIGNER 3.0

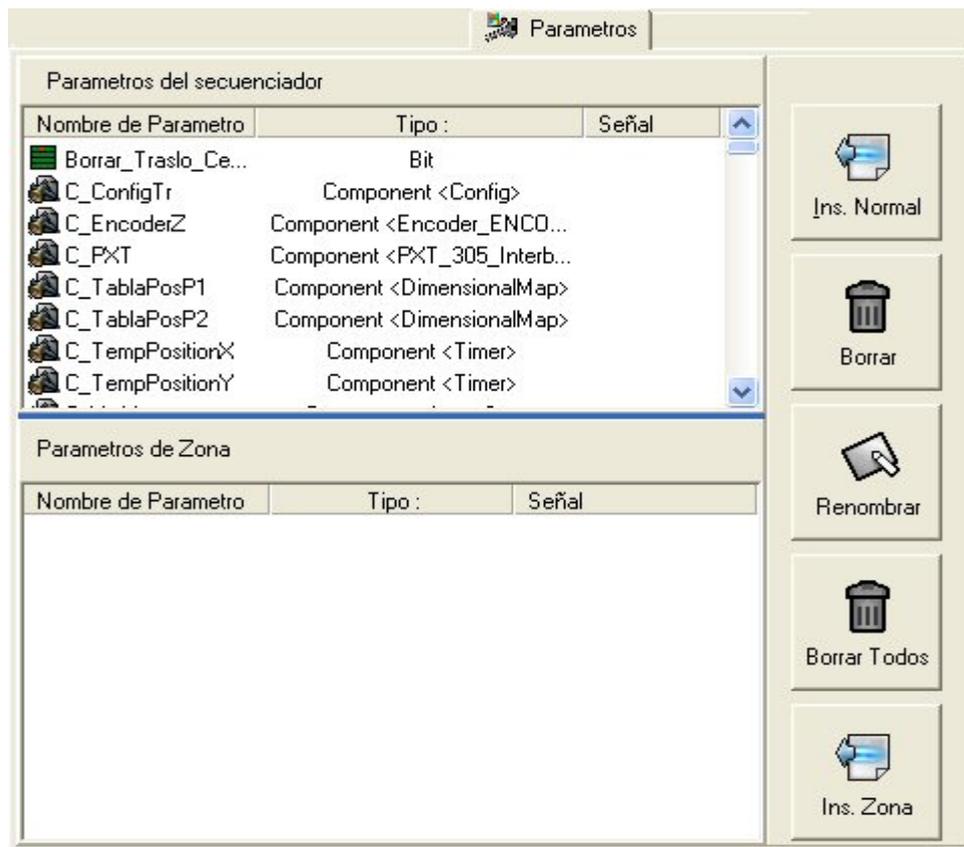


Ilustración 50: parámetros de un secuenciador



Para insertar un nuevo parámetro usaremos, según sea un parámetro de zona o de secuenciador, los botones con este icono. Al hacerlo aparece un diálogo como el de la siguiente figura:

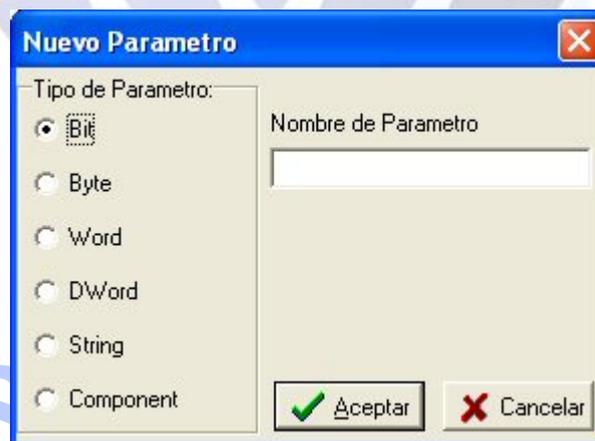


Ilustración 51: Formulario de creación de parámetro

Que nos permite seleccionar el tipo de parámetro que deseamos crear. Si es un parámetro de tipo “Componente”, tendremos que especificar también su interfaz.

Si lo que deseamos es borrar alguno de los parámetros mostrados, usaremos los botones con el símbolo de la papelera. Siempre actuarán sobre el último parámetro seleccionado. Si usamos el botón de “Borrar todos”, eliminaremos todos los parámetros no heredados, tanto los del secuenciador como los de la zona.



También existe la posibilidad, de renombrar un parámetro para cambiarle el nombre por otro más apropiado. Para ello basta con seleccionarlo y pulsar el botón con el icono correspondiente.

Estas dos últimas operaciones también están disponibles a través del menú contextual del botón derecho del ratón cuando se selecciona algún parámetro de las listas. Desde dicho menú también es posible asignar las señales que posteriormente se podrán controlar/ver desde el pupitre virtual.

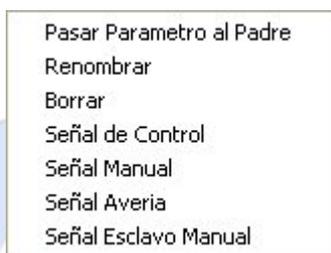


Ilustración 52: Menú contextual de edición de parámetros del secuenciador

Las cuatro últimas opciones son las que permiten asociar señales. Si queremos que las máquinas que implementan un secuenciador sean susceptibles de control desde el pupitre virtual, es necesario al menos que tenga asociada la señal de Manual. Si quedemos desasociar una señal previamente asociada, bastará con que seleccionemos de nuevo la opción de “Señal de control”. Estas asociaciones, al igual que los parámetros en sí, son heredadas por los secuenciadores hijos del actual. Al seleccionar por primera vez esta opción sobre un parámetro de tipo bit (son estos los únicos parámetros que pueden asociarse a señales), se pide el nombre que se desea asignar a esta señal, y será esa la cadena que aparecerá en el pupitre virtual, junto a la marca de la señal. Por supuesto, para que dicha señal aparezca, es necesario que previamente este parámetro haya sido asociado a una variable real mediante la edición del simbólico.

Al igual que en las funciones, existe la posibilidad de pasar un parámetro al secuenciador padre, de forma que sea accesible para los secuenciadores “hermanos” del secuenciador actual, así como de sus descendientes.

También al igual que con el código, los iconos de los parámetros cambian cuando los parámetros son heredados.



parámetro de tipo Bit.



parámetro de tipo Byte.



parámetro de tipo Word.



parámetro de tipo Dword.



parámetro de tipo String.

DESIGNER 3.0

 parámetro de tipo Componente.

 parámetro de tipo Bit heredado.

 parámetro de tipo Byte heredado.

 parámetro de tipo Word heredado.

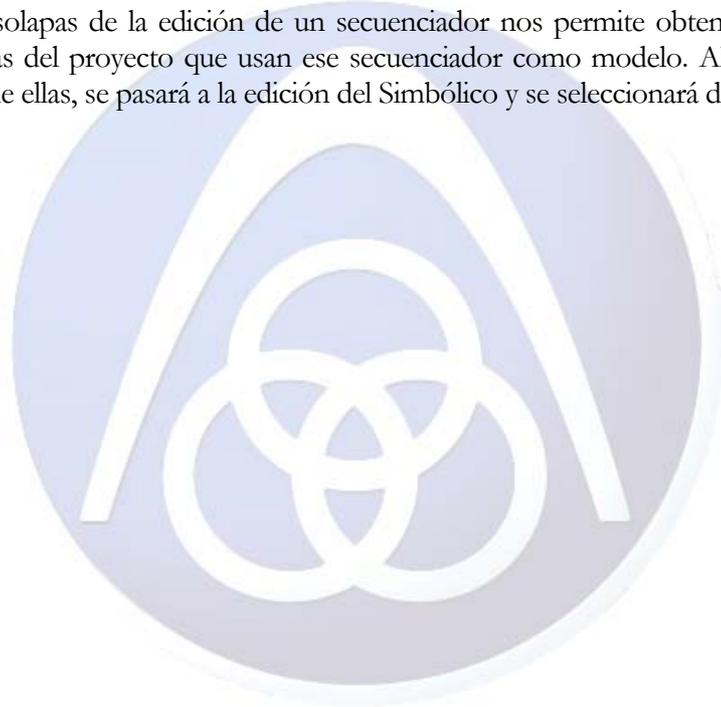
 parámetro de tipo Dword heredado.

 parámetro de tipo String heredado.

 parámetro de tipo Componente heredado.

La pestaña "Instancias"

La última de las solapas de la edición de un secuenciador nos permite obtener un listado rápido de todas las máquinas del proyecto que usan ese secuenciador como modelo. Al realizar un doble click sobre cualquiera de ellas, se pasará a la edición del Simbólico y se seleccionará dicha variable en él.



DESIGNER 3.0

Edición de Simbólico

Una vez definidos los comportamientos del sistema mediante los secuenciadores, es el momento de empezar a usarlos. Esta es la tarea que desempeña el editor de Simbólico, que nos permite definir cuantos elementos de cada tipo queremos manejar.

La edición del simbólico ha sido tradicionalmente una tarea tediosa y complicada, debido a que era una tarea repetitiva y monótona donde era fácil equivocarse. Asimismo, los errores en la definición de un nuevo miembro no eran sencillos de localizar siempre, y la cantidad de información que acarrea cada uno de sus elementos ocasiona que sea difícil controlarlos.

A este plugin se accede pulsando sobre el nodo “Variables” que cuelga del árbol de cada proyecto, lo que provoca la aparición del formulario que se muestra en la Ilustración 53.

En el árbol de la izquierda se verán las máquinas, zonas y variables globales definidas, y pulsando sobre cualquiera de ellas, los valores que las definen aparecerán en los campos de la derecha. Inicialmente, estos valores no pueden ser modificados, para conseguirlo hay que usar el botón derecho del ratón sobre la variable en cuestión y seleccionar la opción “Propiedades”, o bien hacer doble click sobre la variable, lo que hará que sus campos pasen a ser editables. Una vez hechas las modificaciones, hay que pulsar el botón “Actualizar” para que estas se hagan efectivas. Hay que tener en cuenta que, si mientras estamos editando alguna variable seleccionamos otra en el árbol del simbólico, los campos dejarán de estar en modo edición y pasarán a mostrar los campos de la nueva variable, perdiendo los cambios que hubiéramos realizado.

 Cambiar el nombre a las variables, A diferencia de la versión 2.0, no está permitido modificar el nombre de una variable editando directamente el nombre de su nodo en el árbol del simbólico

DESIGNER 3.0

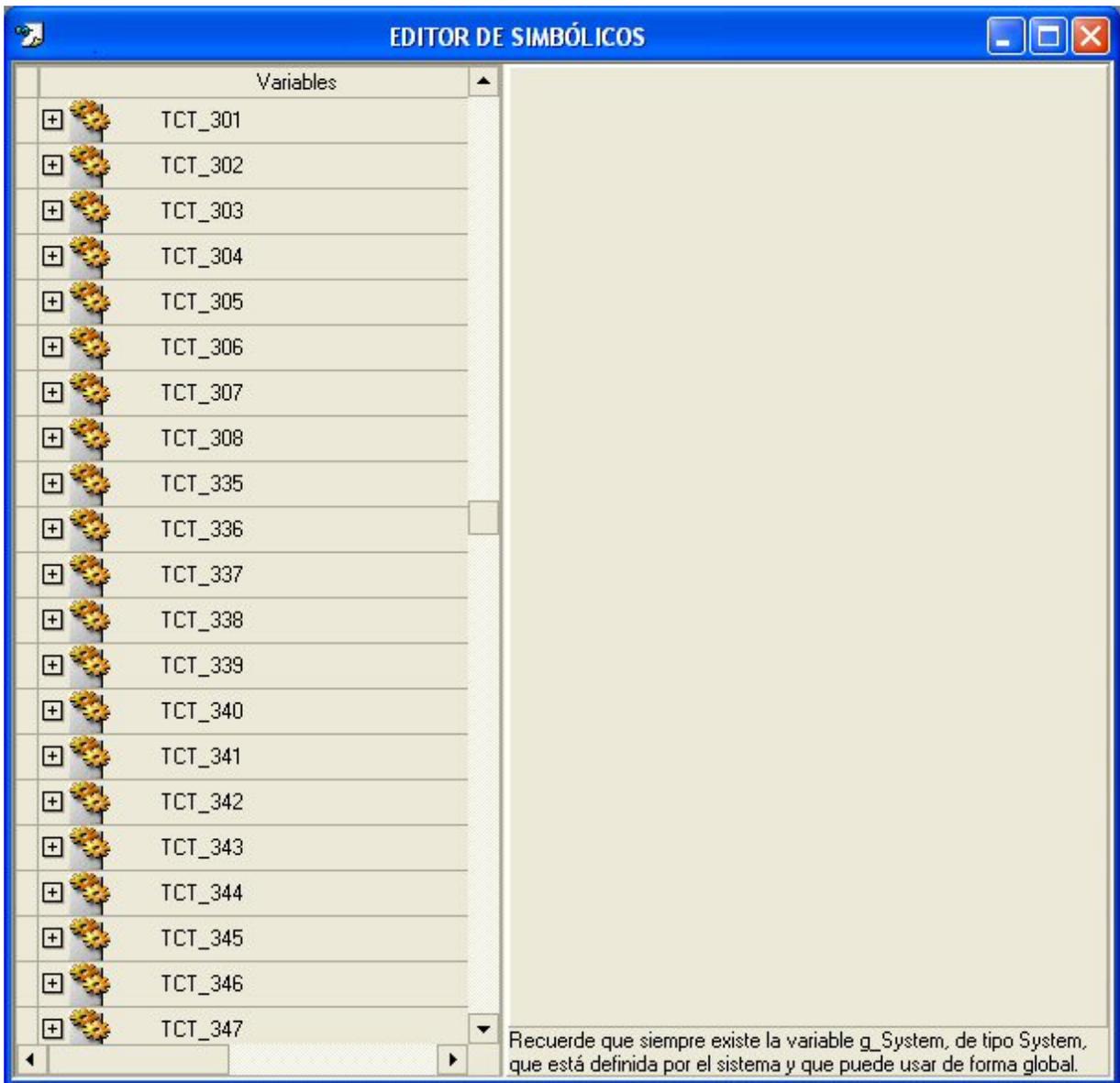


Ilustración 53: El editor de simbólico

Para trabajar con las variables que aparecen en el árbol, usamos, básicamente el menú del botón derecho del ratón que aparece en la siguiente figura:

DESIGNER 3.0

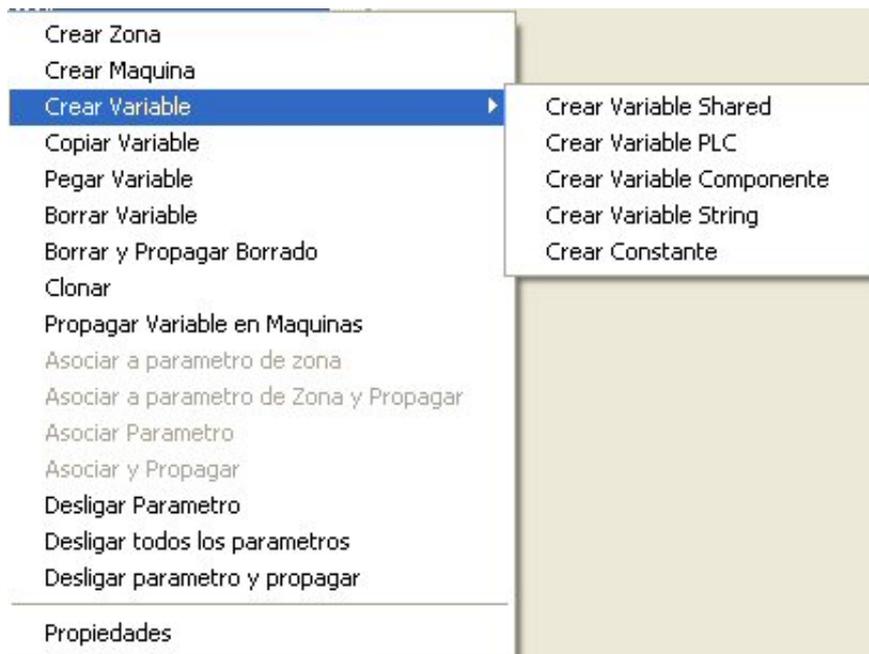


Ilustración 54: Menú del botón derecho.

Las opciones de “Crear” permiten añadir un nuevo elemento al simbólico, del tipo seleccionado.

“Borrar variable” permite eliminar una variable del simbólico. Se consigue el mismo efecto seleccionando la variable y pulsando la tecla **Supr**. Si dicha variable está asociada a un parámetro, se rompe la asociación (pero no se borra el parámetro). Si lo que se borra es una máquina o una zona, esto implica que también se borran todas las variables que contiene.

Si lo que se pretende es eliminar una asociación variable-parámetro, es necesario usar la opción “Desligar parámetro”, que rompe esa asociación sin eliminar la variable. Si se usa la opción “Desligar todos los parámetros” se rompen todas las asociaciones de variables con parámetros para una máquina.

La opción “Desligar y propagar” tiene un significado algo distinto: Se desliga la variable seleccionada del parámetro al que esté asociada, y posteriormente se recorren el resto de máquinas definidas como instancias del mismo secuenciador buscando variables con el mismo nombre y si están asociadas a parámetros, se repite la operación de desligarlas. Esto es muy útil para realizar cambios masivos en el simbólico.

En general, las opciones de “Propagar” implican que el cambio a realizar se propaga a todas las máquinas del simbólico que usan el mismo secuenciador. Por ejemplo, propagar una variable significa que se recorren todas las máquinas del mismo secuenciador que la actual y si no tienen una variable con el nombre de la seleccionada, se crea una automáticamente.

Un caso particular de la propagación de variables es cuando se intenta propagar una variable de tipo “PLC”. Al hacerlo, un diálogo nos pedirá que seleccionemos otra de las variables PLC de la misma máquina en la que estamos. Esta variable sirve como referencia para actualizar el bus al crear la nueva variable en otras máquinas. Es decir, sin esta referencia, la propagación de la variable tendría que copiar todos los valores de la variable original, incluyendo el módulo de bus que usa. Esto sería poco práctico,

dado que luego probablemente tendríamos que ir cambiando el módulo de bus en todas las máquinas donde se ha ido propagando. Para evitarlo, se nos pide que seleccionemos esa otra variable, que sirve únicamente para tomar como referencia en las otras máquinas. Se buscará esta nueva variable en las otras máquinas a propagar y si se encuentra, se cambiará el bus usado en la variable a propagar por el que tiene la variable de referencia en la máquina de destino.

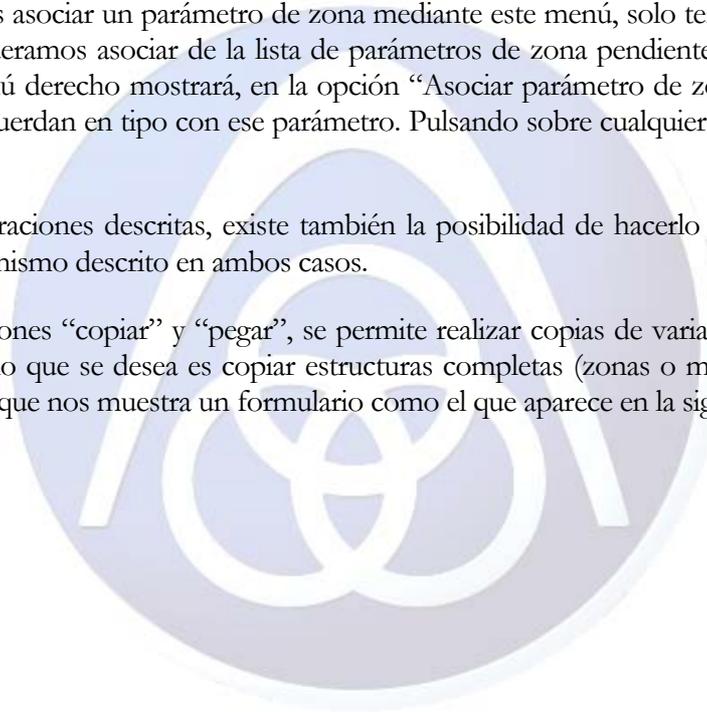
De la misma manera, propagar una asignación “variable-parámetro” significa que se buscan las máquinas del mismo secuenciador que la máquina actual, buscando variables con el mismo nombre y tipo de la actual, y si aun no tienen un parámetro asignado, se realiza esa asignación.

Las acciones de asociación desde este menú son bastante directas: si cuando pulsamos con el botón derecho del ratón sobre una variable esta aun no está asociada a un parámetro, la opción “Asociar a parámetro” mostrará una lista con todos los parámetros del mismo tipo de la variable aun sin asignar en esa máquina. Pulsando sobre cualquiera de esta lista, realizaremos la asociación.

Cuando queramos asociar un parámetro de zona mediante este menú, solo tenemos que seleccionar el parámetro que queramos asociar de la lista de parámetros de zona pendientes de la máquina. En ese momento, el menú derecho mostrará, en la opción “Asociar parámetro de zona”, la lista de variables de zona que concuerdan en tipo con ese parámetro. Pulsando sobre cualquiera de ellas, realizaremos la asociación.

En estas dos operaciones descritas, existe también la posibilidad de hacerlo propagando a la vez. El mecanismo es el mismo descrito en ambos casos.

Mediante las opciones “copiar” y “pegar”, se permite realizar copias de variables “simples” a lo largo del simbólico. Si lo que se desea es copiar estructuras completas (zonas o máquinas), ha de usarse la opción “Clonar”, que nos muestra un formulario como el que aparece en la siguiente ilustración:



DESIGNER 3.0

Ilustración 55: El formulario "clonar"

En esta pantalla vemos que podemos copiar una máquina o una zona. Para realizar la operación de forma efectiva, tras seleccionar lo que queremos copiar, solo tenemos que darle un nuevo nombre a la estructura y en el caso de las máquinas, asignarle también un número no repetido en ninguna otra máquina. La parte central la ocupa la lista de módulos usados en la máquina que vamos a usar como modelo y la correspondencia que tendrán en la nueva estructura. Si lo dejamos tal cual, todas las variables PLC y Componentes clonados usarán la misma referencia que la original. Si los modificamos, se usará esa tabla para encontrar el equivalente. Dado que modificar toda la tabla puede ser algo tedioso, tenemos la opción de usar el botón “Usar offset”, que si esa pulsado, cada vez que realicemos un cambio en la lista de asociaciones, intentará cambiar todas las otras referencias usando el offset en el bus que señale la equivalencia que acabamos de escoger.

Pongamos un ejemplo: si tenemos los siguientes módulos de bus:

Modulo	Tam. entradas	Tam. Salidas	Dir Entradas	Dir Salidas	Tarjeta
BM00	2	2	0	0	T1
BM01	2	0	2	2	T1
BM02	0	2	4	2	T1
BK00	4	4	4	4	T1
BM10	2	2	8	8	T1
BM11	2	0	10	10	T1
BM12	0	2	12	10	T1

Tabla 3: Ejemplo de configuración de bus

Si tenemos una máquina que usa los módulos siguiente y queremos copiarla:

Modulo Origen	Cambiar a...
BM01	BM01
BM02	BM02

Tabla 4: Ejemplo de módulos a clonar

Inicialmente, las columnas de la derecha presentan esos valores. Si tenemos la opción de “Usar offset” pulsada, y cambiamos el valor de BM01 → BM11 entonces, el programa calcula que la diferencia entre BM01 y BM02 que es +2 en entradas y +0 en salidas, luego calculará que el módulo equivalente para BM02 será el que tenga de entradas $BM11+2 = 12$ y de salidas $BM11+0=10$, que corresponde a BM12, y actualizará esa posición con ese valor.

Hay que tener en cuenta que a veces estos cambios pueden no ser evidentes, al estar los módulos repartidos entre tarjetas distintas.

Nodos del árbol del simbólico

Asociado a cada nodo del árbol, hay un icono que proporciona información sobre la variable que representa.



Indica una variable de tipo Zona, que puede contener otras variables. No puede ser asociada a parámetros.



Indica una variable de tipo Máquina. Las Máquinas son instancias de un secuenciador, por lo que se debe escoger a cual representan. Asimismo, requieren de una zona para definirse por completo. Al igual que las de tipo Zona, puede contener otras variables, pero no puede ser asociada ella misma a parámetros, aunque sí las variables que contiene.



Indica una variable de tipo PLC. Las variables de tipo PLC van ligadas a la periferia, y requieren de una referencia a un módulo de bus. Si pertenecen a una Máquina, pueden ligarse con los parámetros del mismo tipo del secuenciador que esta implementa.



Indica una variable de tipo Shared. Las variables de tipo Shared ocupan una zona de memoria pero no van ligadas a la periferia. Si pertenecen a una máquina, pueden ligarse con los parámetros del mismo tipo del secuenciador que esta implementa. Las variables Shared pueden ser a su vez de dos tipos: Read o Write. Las variables Shared Read no pueden ser modificadas por programa, por lo que habrá que tener este hecho en cuenta al usarlas.



Indica una variable de tipo Componente. Representan un objeto nativo en lenguaje Xana, capaz de realizar unas determinadas tareas de forma automática. Típicamente hay componentes que permiten controlar hardware específico o realizar operaciones sólo al alcance del Sistema Operativo donde se ejecuta el Sistema Galileo.



Indica una variable de tipo String. Las variables String son cadenas de texto de longitud variable y compuestas de una o varias líneas, que pueden usarse para pasar como referencia a los

componentes o para implementar **código inline**. Lo mismo que las anteriores, si pertenece a una máquina pueden ser asociadas a los parámetros de su secuenciador. Un detalle importante a tener en cuenta es que aunque la cadena ocupe varias líneas, al usarse en el código XANA se comprimirá a una sola línea.



Indica una constante. Las constantes funcionan del mismo modo que las variables de tipo Shared, la diferencia es que ya tienen un valor asociado que no puede ser cambiado durante la ejecución del programa en el Sistema Galileo. También pueden asociarse a los parámetros de un secuenciador cuando están definidas dentro de una máquina.



Este icono no indica realmente una variable, sino una asociación. Sólo aparece como nodo hijo de una Máquina, cuando la columna “Variables” está vacía, y la de “Parámetro” contiene un nombre, este indica el nombre de un parámetro del secuenciador que aun no ha sido asociado en la máquina. Cuando se realiza la asociación del parámetro con una variable, el icono de la variable sustituye a este icono. La excepción ocurre cuando el parámetro es de tipo “zona” y esta asociación se realiza. En este caso, el icono permanece, indicando que la variable en realidad es una referencia. Además, este icono indica que no se pueden editar los valores de este nodo. (Los parámetros se editan desde su secuenciador y las variables de zona deben editarse desde su nodo correspondiente).

Al abrir el plugin de edición de simbólico, un nuevo elemento de menú aparece en el menú principal del entorno de desarrollo, llamado “Vista”. Los contenidos del menú se pueden ver en la siguiente ilustración:

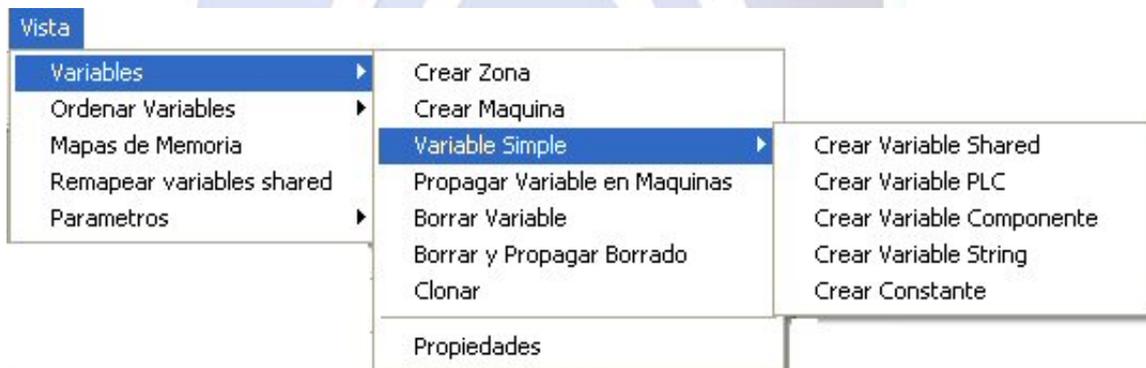


Ilustración 56: Menú principal "Vista"

Como se ve, la sección de “Variables” corresponde en su mayoría con el menú derecho del ratón y sus opciones tienen idénticos efectos.

La opción “Mapas de memoria” permite visualizar los mapas de memoria del simbólico, muy útiles para comprobar los posibles solapes de variables, como explicaremos más adelante.

La opción “Remapear variables Shared” permite compactar la memoria usada para las variables de tipo SHARED definidas en el simbólico. Esto puede ser muy útil si hemos creado y borrado muchas variables, ya que cada definición consume una dirección de memoria del mapa de intercambio, y si se borra no se recupera el valor hasta que se compacte esa memoria. La compactación consiste básicamente en situar las variables en zonas contiguas de memoria, evitando dejar huecos. Puede verse

el efecto que esto provoca mostrando el Mapa de Variables antes y después de una compactación. Este cambio también puede observarse en la barra de tareas principal de la aplicación.

Por ejemplo, la mezcla de proyectos es un caso que suele provocar bastante fragmentación de memoria cuando se importan máquinas y zonas con el mismo nombre que las ya existentes.

También es posible cambiar la ordenación de los nodos en el árbol del simbólico, para que se muestren ordenados alfabéticamente, por tipo o por número de máquina, de forma que sea más sencillo localizar una variable o conjunto de ellas. Los criterios posibles se muestran en la siguiente ilustración:



Ilustración 57: Opciones de ordenación

El criterio con el punto a su izquierda será el criterio seleccionado actualmente. La ordenación alfabética es el criterio por defecto, que se usa siempre que se abre la edición del simbólico.

En cuanto a los parámetros, el acceso a sus opciones desde este menú es un poco diferente que usando el menú del botón derecho, veamos las opciones:

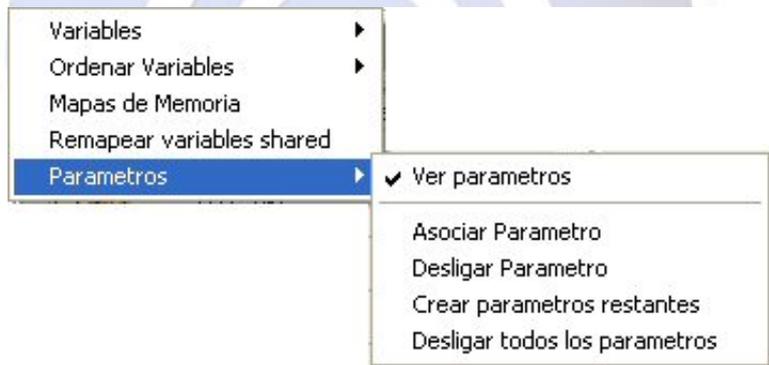


Ilustración 58: Menú de parámetros principal

En este menú no aparecen listados los parámetros restantes, sino que al pulsar sobre la opción de “Asociar Parámetro” aparecerá el siguiente formulario:

DESIGNER 3.0

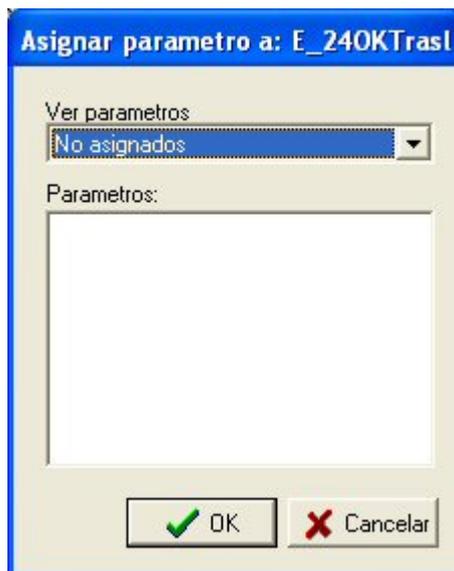


Ilustración 59: formulario de asignación de parámetros

Para la variable que tuviéramos seleccionada, aparecerán los parámetros de su mismo tipo en la lista de parámetros. Inicialmente estarán filtrados, mostrando sólo aquellos que aun no han sido asociados. Podemos modificar este filtro para mostrar todos los posibles parámetros a asignar, pero hemos de tener en cuenta que si finalmente asociamos la variables a un parámetro que ya estaba asociado a otra variable, esto NO implica que el viejo vínculo se rompe y es reemplazado por el nuevo. Cuando hayamos seleccionado el parámetro a asociar, basta con pulsar el botón “OK” para confirmar la asociación.

La opción “desligar parámetro” funciona exactamente igual que desde el menú del botón derecho, rompiendo un enlace parámetro-variable.

“Crear parámetros restantes” es una opción que sólo es accesible desde este menú. Para usarla se debe tener una máquina seleccionada y provocará la creación de variables que se ligarán de forma automática a los parámetros restantes por asignar. Las variables creadas automáticamente tendrán el mismo nombre que el parámetro, su mismo tamaño y su tipo será Shared, String o Componente.

Por último “Desligar todos los parámetros” se puede usar cuando se tiene una variable de tipo Máquina seleccionada y permite romper todos los vínculos de la máquina con los parámetros de su secuenciador.

El mapa de variables en la tarjeta

Este es un formulario que nos muestra como se sitúan las variables que hemos definido en la memoria de periferia (para la variables PLC) o de intercambio (para variables SHARED). Al mostrarlo, aparece un formulario como el mostrado en la Ilustración 60 en la parte derecha del plugin del simbólico.

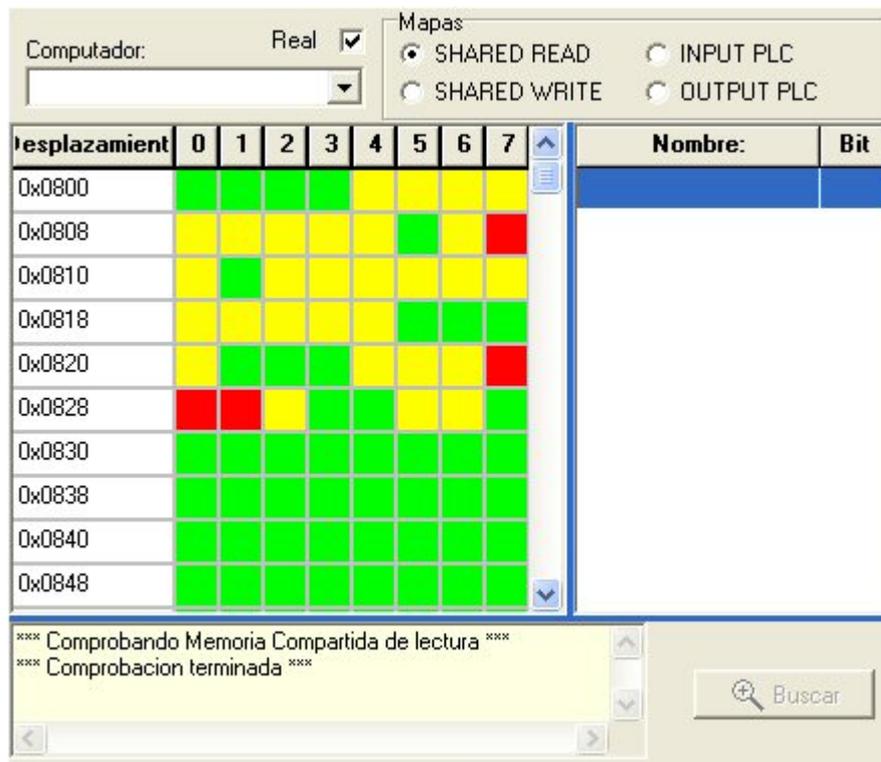


Ilustración 60: Mapa de variables.

Las celdas de memoria aún no ocupada aparecerán de color verde. Las celdas de memoria parcialmente ocupadas (por ejemplo, cuando sólo se ha reservado algún bit), aparecerán en color amarillo, y las que están completamente ocupadas, aparecen de color rojo. Si alguna celda aparece de color negro, ello indica un posible solapamiento entre las variables definidas. En este caso también veremos mensajes anunciando este hecho en el campo de texto de la parte inferior del formulario, donde se nos informará también de la dirección de memoria donde se está produciendo el solapamiento.

Aparte de detectar solapamientos, el mapa de variables da una idea bastante aproximada de cómo está de ocupada la memoria del sistema Galileo, y también permite localizar las variables que se encuentran definidas en una posición de memoria concreta, simplemente seleccionando su celda y pulsando el botón “Buscar”, lo que hará que las variables definidas en esa dirección aparezcan en la lista de la derecha.

Se pueden ver 4 mapas de memoria: El de las variables Shared de Entradas, las variables Shared de salidas, las variables PLC de entradas y las variables PLC de salidas. Para poder ver estos últimos es necesario que seleccionemos un computador de la lista, dado que estos mapas se conforman en base a los módulos de bus de entradas/salidas presentes en los computadores que se seleccionen, y por lo tanto, varían de una computadora a otra.

Relacionado con este último punto está el botón de comprobación con la leyenda “Real” que aparece en la parte superior. Su explicación es la siguiente: El mapa de entradas/salidas PLC de un computador se puede ver desde dos puntos de vista:

- 1) Por los módulos de bus mapeados en sus tarjetas.

- 2) Por las máquinas que se están asignando a ese computador.

El segundo criterio es el que se usa a la hora de compilar, por lo cual es el “real”, y es el que se usa por defecto. El primero es el caso que debería ser obligatorio, puesto que no tiene sentido asignar máquinas con elementos de bus en otro computador en otra computadora distinta. Sin embargo, si por descuido o necesidad esto ocurriera, esto modifica la configuración “real” del bus, dado que el módulo al que apunta siempre se considera local a la computadora donde se ejecuta el programa por parte de Galileo. La comparación entre estos dos tipos de mapas puede darnos pistas de que alguna máquina no está asociada a la computadora que debería (puede ocurrir tras pasar módulos de las tarjetas de una computadora a las tarjetas de otra y reconfigurar el bus).

En el caso de que alguno de los mapas se mostrase totalmente lleno, o alguna de las variables sobrepasase su rango de memoria, se mostraría un error que nos indicaría que con esos valores, el Sistema Galileo no va a poder funcionar de forma adecuada, así que deberemos ajustarlos antes de poder generar un fichero PLC válido.

Campos de Formularios de Edición de variables.

A continuación veremos los distintos campos a rellenar para el caso de cada uno de los posibles tipos de variables existentes.

Una característica común de todos estos formularios es que si alguno de los campos de edición aparece vacío, obtendremos un mensaje de error cuando intentemos aplicar los cambios. Sin embargo, en el futuro alguna modificación del proyecto puede hacer que alguno de esos campos aparezca vacío (por ejemplo, eliminar una estación de bus deja vacíos los campos “BUS” de las variables PLC que lo usaran, pero no borra las variables). Es por ello que a veces es necesario comprobar la consistencia del proyecto mediante el plugin de comprobación de proyecto.

En todos los formularios de creación de variables simples aparece un botón de chequeo que si se activa, especifica que la variable a crear debe crearse como global al proyecto, y no local a la última máquina o zona seleccionada. No es necesario seleccionarlo si no existen máquinas o zonas aun en el proyecto.

Machines

Cuando se crea una variable de tipo “MACHINE” los campos que aparecen en el formulario para rellenar son los que se aprecian en la Ilustración 61.

DESIGNER 3.0

Ilustración 61: Campos de la definición de una variable MACHINE.

Un campo importante es el número de la máquina, que no puede repetirse en ninguna otra máquina del proyecto ni valer 0. Este número de máquina no tiene que ver con el orden en que son manejadas las máquinas por el Sistema Galileo, sino que es un identificador interno usado por el mismo.

Zonas

Para crear una nueva zona solo es necesario darle un nombre, tal y como se ve en la ilustración que sigue:

Ilustración 62: Definición de una Zona

Variables PLC

Las variables PLC son las que más elementos necesitan para definirse, concretamente podemos ver todos los campos en la ilustración siguiente:

DESIGNER 3.0

Ilustración 63: Definición de una variable PLC

El selector Input/Output establecerá la variable dentro del bus de entradas o de salidas. Asimismo, puede tener 4 tamaños distintos: Bit, Byte, Word (2 bytes) o Dword (4 bytes). Solo en el caso de que se seleccione el tamaño bit, será necesario especificar que bit dentro del byte queremos usar.

El tiempo de retraso es, como su propio nombre indica, un periodo de tiempo, medido en milisegundos, que queremos retrasar el cambio de la señal (en subida o en bajada), desde que esta se produce en el bus hasta que el programa de control se aperceba de ese cambio. Si seleccionamos “No”, el programa de control será consciente de ese cambio tan pronto como este se produzca.

Variables Shared

Las variables Shared son mucho más simples de definir que las PLC, aunque básicamente comparten las mismas posibilidades en cuanto a tamaño, y en lugar de Entrada/Salida, se dividen en Lectura/Escritura (las primeras no son modificables desde el programa). La siguiente ilustración nos muestra los campos de su definición:

✦ La dirección de las shared, aunque la dirección en memoria de las variables shared no es algo que el usuario necesite conocer, a veces sirve para confirmar que varias copias del mismo proyecto tienen la misma visión del simbólico (si no es así provocará errores en la comunicación. Para conocer la dirección de una variable Shared, en lugar de buscarla por el mapa del simbólico, podemos simplemente hacer doble click sobre la etiqueta “Nombre” de sus propiedades; Aparecerán dos etiquetas en la parte inferior que nos informarán de la dirección que tiene asignada.

Ilustración 64: Definición de una variable Shared

El resto de campos de las Shared no es necesario especificarlos, ni siquiera el bit, puesto que es asignado internamente por el entorno de desarrollo.

La única diferencia notable es que si la variable que seleccionamos es de tipo “Read”, se nos permite establecer su valor inicial mediante un campo “Valor” que aparece al efecto, al igual que en el caso de las constantes. La diferencia con estas es que las variables Shared Read pueden modificar su valor posteriormente mediante comunicaciones (forzado).

Strings

Las Strings son cadenas de valores y usos muy dispares, para definir las basta simplemente con dales un nombre y su valor, tal y como vemos en el siguiente formulario:

Ilustración 65: Definición de un String

Componentes

Los componentes típicamente requieren de un nombre y el interfaz para ser definidos. No obstante, en función del interfaz que intentemos que usen, pueden requerir que se les asocie también un módulo de bus con el que comunicarse, tal y como se muestra en la siguiente figura:

Ilustración 66: definición de una variable Componente

Al igual que las variables Shared, necesitamos establecer la estación de bus y el byte de origen donde empieza el módulo con el que van a comunicarse. Esto es necesario habitualmente en componentes que realizan tareas de acceso o control de componentes hardware. Si el componente no tiene un interfaz de este tipo, esos campos desaparecen de la edición al pulsar “Crear” y no vuelven a aparecer a menos que cambiemos el interfaz del componente por otro que vuelva a requerirlos.

Constantes

Las constantes, como ya se ha comentado, son similares a las variables shared, pero con un valor prefijado que no puede cambiarse de ninguna manera. Así pues, para definir las hay que rellenar el siguiente formulario:

Ilustración 67: Definición de una constante

Las únicas diferencias destacables con respecto a la definición de una Shared es el campo “Valor” que debe tener asignado un valor numérico dentro de los rangos del tamaño seleccionado:

1 ó 0 para los Bit.

-128 a 127 para los byte

-32767 a 32766 para los Word

-2147483647 a 2147483646 para los DWORD



DESIGNER 3.0

Configuración del Agente de Transportes

El agente de transportes es el eje sobre el que giran las comunicaciones entre la gestión informática realizada por un SGA y la gestión de ordenes realizada por el sistema Galileo.

Galileo lleva aparejada una gestión de ordenes configurable que es configurada y realizada desde el paquete PL/SQL denominado **Agente de Transportes**. Tradicionalmente la configuración de este paquete se realizaba directamente sobre la base de datos, con el inconveniente que ello suponía (no había una herramienta efectiva que facilitase la tarea y la configuración no se almacenaba en conjunción con el proyecto, sino de forma independiente). Para tratar de fortalecer este vínculo entre las dos partes, el Entorno de Desarrollo permite realizar todos los pasos de dicha configuración, a la vez que almacenar el resultado junto con el proyecto, desde el plugin de Configuración del Agente de Transportes.

Paquete PATHFINDER

El paquete PATHFINDER se suministra para ser instalado desde el propio entorno de desarrollo. Es aconsejable hacerlo al comenzar la instalación de la base de datos que proporcionara soporte al Sistema Galileo. Sobre todo a fin de utilizar siempre la última versión del paquete, que se entrega con el entorno de desarrollo.

La instalación que se hace del paquete va encriptada, de forma que no es posible editar el código PL/SQL que contiene.

El paquete en sí contiene procedimientos y “Triggers” escritos en PL/SQL que sirven para insertar transportes, ordenes, estaciones y demás elementos de la parte de gestión en las tablas correspondientes. Si dichas tablas no existiesen, podrían ser creadas desde el entorno de desarrollo también.

Estaciones, Ordenes, Trayectorias y Defectos.

Antes de seguir hablando de las posibilidades de configuración, vamos a explicar unos conceptos básicos:

Estaciones. Las estaciones son puntos en los cuales los elementos de control deben de finalizar la orden que tengan en un momento dado y buscar otra para seguir avanzando.

Ordenes: Las ordenes son lo que su nombre indica: peticiones por parte del sistema de gestión para que una unidad de transporte sea llevada desde una estación de inicio a una estación final.

Trayectorias: Recorridos posibles de las unidades de transporte por la instalación. Toda orden debe seguir una trayectoria para llegar desde la estación inicial a la final, que se decidirá en función de varios criterios.

Defectos: Este es un concepto un poco más genérico y no tan ligado al agente de transportes. En general, se definen los defectos como perturbaciones en el comportamiento normal de algún elemento de la instalación que requieren que el personal de mantenimiento efectúe algún ajuste antes de poder seguir funcionando con normalidad. A fin de ayudar al personal a reaccionar ante las causas del problema, los defectos permiten incluir un texto explicativo y la sugerencia de un curso de acción a tomar para cada error definido.

Configuración del Agente de Transportes

La configuración se realiza a través de unas tablas que definen las trayectorias, estaciones y ordenes a buscar. Estas tablas las usará PathFinder para indicar al sistema Galileo lo que debe hacer en cada momento. Para configurar estas tablas de forma que se integre con el resto de la información del proyecto, se usa el plugin de configuración del agente de transporte, al que se accede mediante la opción **Modificar / Generar configuración para el agente de Transportes** o bien pulsando sobre el nodo “Configuración TA” del árbol del proyectos principal. Al seleccionarla, aparece la figura que se muestra en la Ilustración 68.

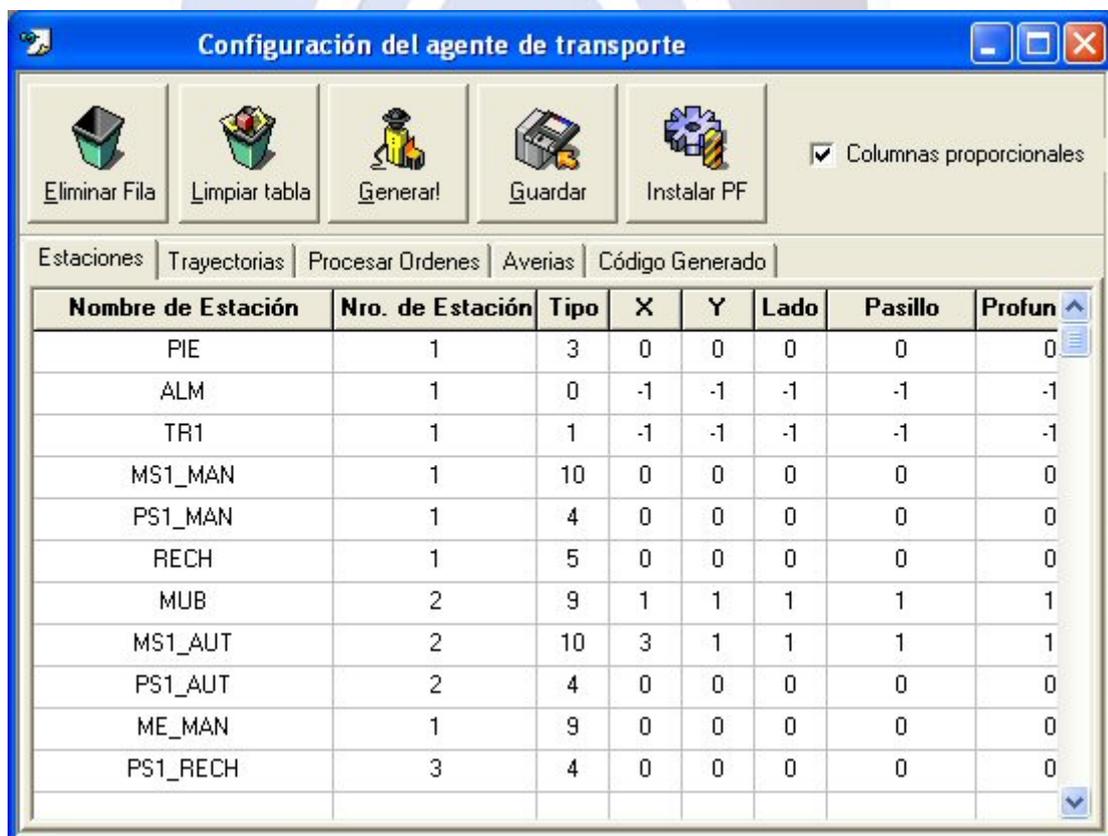


Ilustración 68: Plugin de configuración del Agente de Transportes.

Describamos los botones de la parte superior:



Elimina la fila seleccionada actualmente en la tabla que se está editando.



Limpia la tabla que se está editando en ese momento,



Genera el script una vez que hallamos introducido todos los datos necesarios para configurar el agente. Se nos ofrecerán diversas opciones para generar el script y la posibilidad de ejecutarlo en base de datos directamente.



Salva los cambios introducidos en la configuración a la caché de memoria.



Permite conectarse a la base de datos y cargar el paquete PathFinder en ella, es decir, introducir el código necesario para el funcionamiento del agente de transporte en la base de datos. Para ello, tras pulsarlo y confirmar que efectivamente, queremos cargar el paquete PathFinder en una base de datos, se nos pedirá un usuario y clave para entrar en sesión en la base de datos. Dicho usuario debe tener los permisos necesarios para cargar y compilar paquetes PL/SQL en la Base de Datos. Si introducimos un usuario correcto, se nos preguntará si deseamos crear las tablas necesarias por el Agente en la BD.

IMPORTANTE: Si contestamos que sí en este momento, las tablas que ya existieran con datos de instalaciones anteriores del PathFinder serán borradas, por lo que ante la duda lo mejor es contestar negativamente en este punto. Si las tablas no existiesen, nos daríamos cuenta posteriormente al intentar cargar el código de configuración que generaremos mediante este plugin.

En todo momento, podremos ver como evoluciona el proceso de carga de los paquetes por el avance de una barra de progreso que aparecerá en la parte superior derecha. Cuando dicho proceso se complete, un mensaje nos indicará si todo ha terminado con éxito, y podremos ver un log del proceso en el memo que aparecerá en la parte inferior del plugin.

A la derecha de los botones se puede observar un marcador “Columnas proporcionales”. Si este marcador está pulsado, indica que se quiere que el tamaño de las columnas de la tabla que se está mostrando sean proporcionales, e intenten mostrarse todas en pantalla sin que sea necesario usar la barra de desplazamiento. Si no se pulsa esta opción, el usuario podrá agrandar / disminuir el ancho de cada columna de forma individual en cada tabla (moviendo el ratón sobre la columna de encabezado), para ajustarlo a sus preferencias. Estas preferencias se guardan de forma automática, de forma que serán usadas la próxima vez que se abra este plugin.

Solapa “Estaciones”

Esta pestaña es la primera que aparece cuando se abre el plugin, tal y como se ve en la Ilustración 68. En ella se permite definir los contenidos que se desea que tenga la tabla “Estaciones” del PathFinder.

Durante la edición, debe tenerse en cuenta que los campos “Número estación” y “Tipo” han de ser numéricos obligatoriamente y el campo “Nombre de estación” es de tipo texto. Estos campos son obligatorios: si no se rellena alguno de ellos, la fila se eliminará al salvar la configuración. El resto de

los campos son opcionales, aunque están definidos como numéricos. Si el usuario introduce cualquier cadena no numérica, el valor de dicha columna se fijará a -1, que es un valor comodín.

Los nombres de las estaciones deberían estar relacionados con lo que pretende que representen, puesto que luego se usarán para poder definir la tabla trayectorias.

Solapa "Trayectorias"

En esta pestaña se permite la configuración de la tabla "Trayectorias" del PathFinder. Al pulsarla, se observa la figura que se muestra en la Ilustración 69.

Camino	Función	Camino	Función	Camino	Función	Camino	Función	Camino	Función	MODO	Opción
PIE	nyStore.Iniciar_Transpor	MUB	YSTORE.Fin_ora							M	1
PIE	nyStore.Iniciar_Transpor	RECH	YSTORE.Fin_ora							M	1
MUB	nyStore.Iniciar_Transpor	ALM	YSTORE.Fin_ora							M	1
MUB	nyStore.Iniciar_Transpor	MS1_AUT	YSTORE.Fin_ora	PS1_AUT	YSTORE.Fin_ora					M	1
TR1	nyStore.Iniciar_Transpor	ALM	YSTORE.Fin_ora							M	1
ALM	nyStore.Iniciar_Transpor	ALM	YSTORE.Fin_ora							M	1
TR1	nyStore.Iniciar_Transpor	MS1_AUT	YSTORE.Fin_ora	PS1_AUT	YSTORE.Fin_ora					M	1
ALM	nyStore.Iniciar_Transpor	MS1_AUT	YSTORE.Fin_ora	PS1_AUT	YSTORE.Fin_ora					M	1
PIE	nyStore.Iniciar_Transpor	MS1_MAN	YSTORE.Fin_ora							M	1

Ilustración 69: Solapa de edición de la tabla "Trayectorias".

Con esta tabla se definen "caminos" o "trayectorias" en la instalación. Las columnas MODOS y opción son obligatorias para poder generar el código, y no se toman valores por defecto, a fin de evitar problemas. El resto de las columnas representan pares estación / función, y se debe definir al menos una estación origen y una destino. La estación origen deberá estar OBLIGATORIAMENTE en la primera columna, en caso contrario, esa fila producirá errores durante la generación.

Los "Caminos" se pueden seleccionar del combo de edición que aparecerá al editar una celda concreta, y que permitirá elegir entre todos los que hayamos definido previamente en la tabla "Estaciones".

En cuanto a la función, representa el nombre de una función contenida en el paquete de configuración del PathFinder o en el código de usuario introducido en la base de datos, por ello, las celdas de este tipo admiten cualquier cadena y no se realizan comprobaciones sobre ellas. Así pues, el usuario es el único responsable de garantizar que dichas funciones existen en la base de datos. Si no fuera así, esto provocaría errores al ejecutar el script de configuración generado, o al ejecutarse el procesamiento de ordenes.

Cada usuario de la aplicación tiene la posibilidad de cambiar los colores de las columnas de esta tabla a su gusto, a fin de permitir una visualización más cómoda o adaptada a su monitor, usando el menú que aparece al pulsar el botón derecho del ratón sobre las celdas de la tabla.

Solapa "Procesar Ordenes"

La tercera tabla a configurar es Procesar Ordenes, y al pulsar la solapa correspondiente, aparece la figura que se muestra en la Ilustración 70

Máquina	Función	Tipo
TRE_5	THYSTORE.Palet_En_PIE	PIE
Traslo1	THYSTORE.Busca_Orden_Traslo	Normal
TCF_44	THYSTORE.Busca_Orden_Mesa_Salida	Normal
TRE_5	THYSTORE.Busca_Orden_PIE	Normal

Ilustración 70: Solapa de edición de la tabla "Procesar ordenes".

Todos los campos son obligatorios aquí, y las columnas “Máquina” y “Tipo” llevan asociados combos que permiten seleccionar entre los valores posibles que admiten.

La columna “Máquina” debe contener el nombre de una variable de tipo “Machine” previamente definida en el simbólico de la tarjeta (véase [Edición de Simbólico](#)). Si el nombre introducido no es válido, la fila se eliminará cuando se salve la configuración.

La columna “Función” representa, de nuevo, una función de código PL/SQL contenida en la Base de Datos de destino. Debido a esta causa no se aplica ninguna comprobación al nombre introducido, y es tarea del programador el asegurar que este campo tiene valores correctos, pues de otra manera el agente de transporte no será capaz de llevar a cabo su tarea.

La última columna representa el tipo de orden, que sólo puede ser de dos clases: “Normal” o “PIE”. No se aplica ningún valor por defecto, es necesario que el usuario asigne uno de los valores.

Solapa “Averías”

La última tabla a configurar es la tabla “averías”, y se hace mediante la pestaña correspondiente del plugin, que nos muestra la figura de la Ilustración 71.

Número	Nombre	Descripción	Sugerencia
1001	Emergencia General	Se ha pulsado la seta de seguridad de el armario C0	Una vez comprobada la razón p
1005	Emergencia Zona1	Relé de seguridad correspondiente a la Zona1 caido	Una vez comprobada la razón p
1020	Emergencia Zona 2	Relé de seguridad correspondiente a la Zona2 caido	Una vez comprobada la razón p
1009	Emergencia Pasillo	Relé de seguridad correspondiente al Pasillo caido	Se ha abierto una o ambas Puei
1002	Alimentacion 380V AC	El relé DT0C detecta un fallo en la alimentación cabecera	Ha habido una inversión en las
1007	Alimentacion 380V AC	El relé DT0T detecta un fallo en la alimentación del pasillo	Ha habido una inversión en las
1003	Alimentacion 24V DC	Fallo en la alimentación de 24 V DC que alimenta Cabecer	Comprobar si está saltado el Aut
1021	Alimentacion 24V DC	Fallo en la alimentación del Pupitre 2.1 que permite el man	Comprobar si está saltada la pro
201	Galibo de Altura	El palet dentro de la cuna presenta una altura excesiva	Tomando las medidas de segurii
202	Galibo Lateral Izqd	El palet sobresale por el lado izquierdo de la cuna.	Tomando las medidas de segurii
203	Gálibo Lateral Dcho	Algo sobresale por el lado derecho de la cuna.	Tomando las medidas de segurii

Ilustración 71: Solapa de edición de la tabla "Averías".

Las columnas “Número” y “Nombre” son obligatorias aquí, y la primera sólo admitirá valores numéricos. El resto de columnas admiten campos alfanuméricos de longitud variable, aunque debe

tenerse en cuenta las restricciones aplicadas a la tabla (consúltese la guía de configuración del agente de transporte para más información).

Solapa "Código Generado"

Cuando hallamos completado la información relativa a las tablas, podemos pulsar el botón "Generar" de la parte superior del plugin. En ese momento se nos pedirá un nombre de fichero donde generar el script de configuración del agente, que introduciremos en el diálogo que se observa en la Ilustración 72.

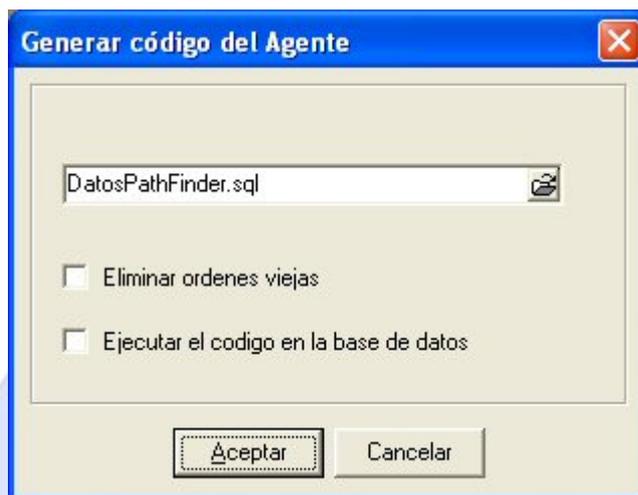


Ilustración 72: Diálogo de generación de configuración.

También tenemos opción, mediante los marcadores correspondientes, de cargar en base de datos el código generado si la generación termina con éxito, y de generar código que elimine las viejas ordenes que aun hubiera en las tablas de la base de datos.

Al pulsar el botón "OK", el plugin genera el código de configuración y lo muestra en la pestaña "Código generado", tal y como se aprecia en la Ilustración 73.

DESIGNER 3.0

```

1  /*
2  FICHERO DE CONFIGURACION DEL AGENTE DE TRANSPORTE
3  Fecha de generación: 10/06/2004 13:20:16
4  Fichero generado automáticamente por Galileo Designer System: NO EDITAR!
5  Copyright (C) 2001-2004 by ThyssenKrupp Ingeniería y Sistemas;
6  Gijón.
7  */
8
9  /* Las siguientes líneas (marcadas entre tags <OLD>) pueden
10 producir un error al intentar ejecutar este script desde ciertos
11 entornos. Si ese es el caso, basta con eliminarlas del script.
12 */
13 /* <OLD> */
14 SET SERVEROUTPUT ON;
15 / /* </OLD> */
16 /* --- ESTACIONES --- */
17
18 DECLARE
19     n Error INTEGER;

```

Ilustración 73: El código PL/SQL generado por la aplicación.

Esto ocurre sólo en el caso de que no se produzcan errores durante la configuración de las tablas. Esto es bastante infrecuente en usuarios avanzados, debido a las restricciones que imponen las otras pestañas durante la edición. Lo que es más frecuente es que el script produzca errores al ejecutarlo en la Base de Datos que contiene el PathFinder.

Para que la opción de ejecutar el código generado (y con ello configurando completamente el agente) en una base de datos sea posible, es necesario que un cliente de Oracle esté instalado en el ordenador donde se ejecuta la aplicación, y que el computador donde está la Base de Datos remota sea accesible a través de la conexión del cliente Oracle.

Cuando llegue el turno de realizar esta tarea, una nueva ventana nos solicitará los parámetros de la conexión: Nombre de usuario (debe tener los permisos adecuados en la BD a fin de ejecutar con éxito el script), su clave de acceso y el nombre de la instancia a la que conectarnos. Si los datos que introducimos son correctos, el plugin se conectará y ejecutará el script en la Base de Datos. El resultado de la ejecución del script se mostrará también en el panel que aparecerá en la parte inferior de la pestaña, con los posibles errores que se produjeron durante dicha ejecución. Si todo fue bien, un mensaje nos informará de este hecho.

AVISO: El realizar esta actualización implica un borrado de las tablas de configuración del agente de transporte en la base de datos. Aunque se rellenen otra vez con los mismo datos, las restricciones con las que se crean estas tablas provocan que el borrado, aunque sea temporal, de un elemento (por ejemplo, las estaciones) haga que desaparezcan filas que los contienen como clave ajena en otras tablas. Esto es particularmente importante en el caso de las tablas de PATHFINDER_TRANSPORTE y PATHFINDER_MSG_TR puesto que provoca la desaparición de las ordenes que hubiera en transporte en el momento de la actualización.

Visualizaciones

Las visualizaciones son la cara visible de todo el proyecto, pues es la parte que finalmente quedara expuesta al uso diario y de la que el futuro usuario de todo el proceso obtendrá información. Por ello hay que ser especialmente cuidadosos a la hora de realizarlas.

No solo por lo anterior las visualizaciones merecen un capítulo aparte: la información que contienen también tiene un significado especial durante el desarrollo, y es necesario organizar dicha información de forma ordenada a fin de localizarla posteriormente en un tiempo razonable.

Cuando seleccionamos un nodo de “Visualización” en el árbol de proyectos, tenemos acceso al editor del formulario de layout definido. Habitualmente, este formulario no es ni más ni menos que una representación esquemática de parte o el total de los elementos de la instalación gobernados por el Sistema Galileo. Si se prefiere, el layout es la versión modificable del formulario de instalación que se usa en la aplicación “Galileo Status Monitor”.

Una vez que el formulario está abierto, se puede pasar a editar los componentes que contiene mediante la opción **Formularios / Modo Diseño** del menú principal, y el resultado es algo similar a lo que se muestra en la Ilustración 74.

A los formularios pueden asignarseles diferentes niveles de acceso a fin de permitir a usuarios con perfiles distintos acceder a unos u otros mediante el uso de sus claves de perfil. Esto se hace a través de la opción de menú “Formularios → Permisos”. Para poder cambiar los permisos a un formulario es necesario tener privilegios de Administrador como usuario de proyecto.

DESIGNER 3.0

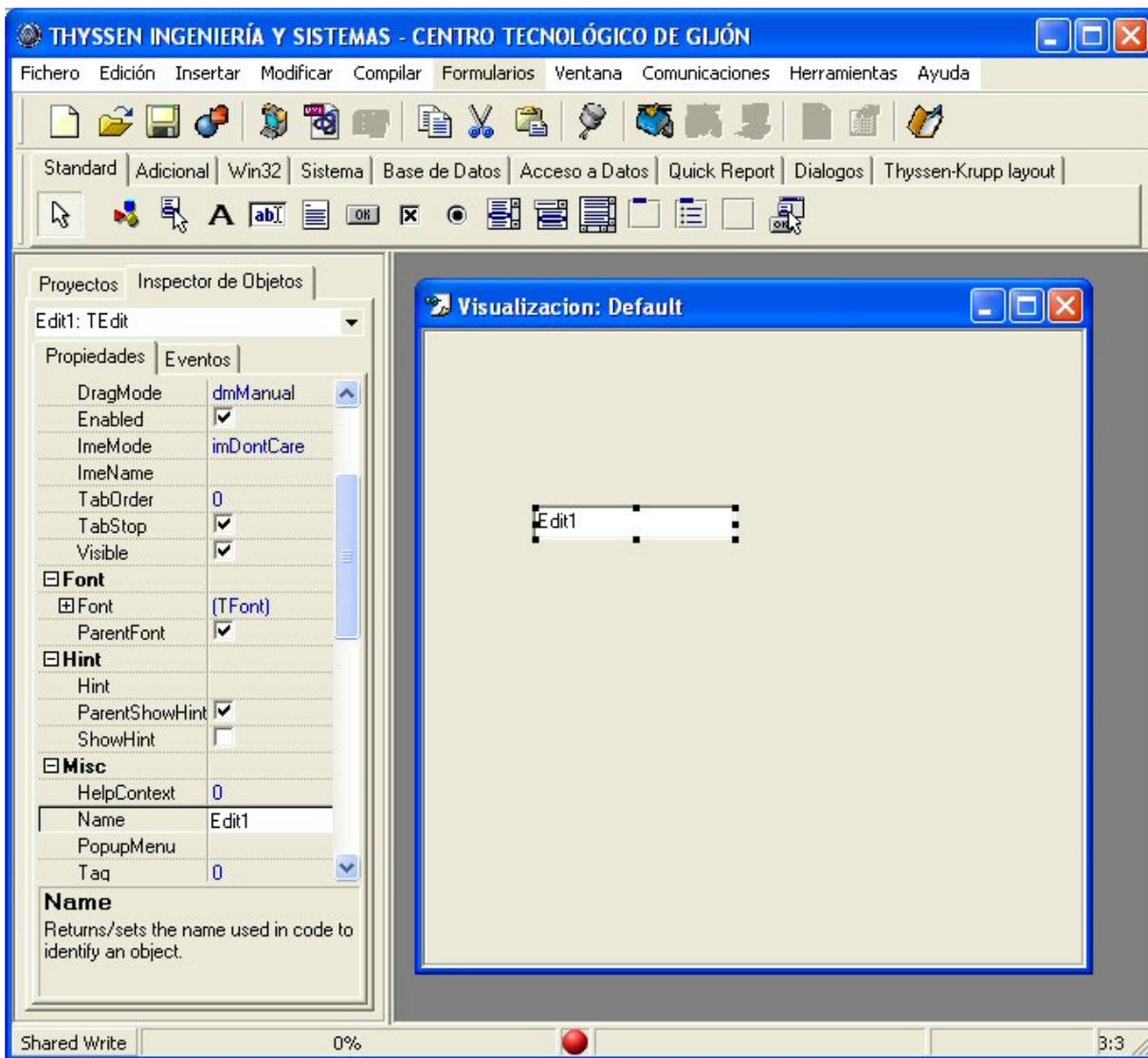


Ilustración 74: Trabajando con la edición del layout.

Modos Runtime / Diseño

Una vez que el formulario aparece en pantalla, podemos usarlo en dos modos:

- **Runtime**, en que veremos el formulario tal y como aparecerá cuando se use desde la aplicación “Galileo Status Monitor”. En este modo no podemos modificar los componentes del formulario, pero podemos usarlos y probarlos, y si tenemos la tarjeta correspondiente en modo “Online”, podremos, mediante la activación de un Timer, obtener datos reales del sistema Galileo en funcionamiento. Los cambios que hagamos al formulario en este modo (por ejemplo: cambiar la solapa activa), no son guardados y al abrir el layout la siguiente vez volverá a estar en el estado inicial.
- **Diseño**, en este modo, la pantalla cambia para mostrarse tal y como aparece en la Ilustración 74, apareciendo una paleta de componentes en la parte superior, que podemos usar para componer el formulario según nuestras necesidades. En este modo, también aparecerá una ventana flotante donde podemos escribir código de script, en el lenguaje JavaScript, que permite programar los eventos de los componentes que hemos situado en el formulario. Los cambios que hagamos en este modo si serán guardados.

Las paletas de componentes.

Los componentes de edición de Layout se agrupan en diferentes paletas según su funcionalidad, pudiendo ser esto fácilmente extensible en versiones futuras, pero de momento comentaremos los iconos que aparecen en cada paleta, y una breve indicación de sus posibilidades de funcionamiento. Esta descripción es muy básica y, salvo en el caso de los componentes de la paleta “Thyssen-Krupp

Layout”, no se referirán las propiedades o métodos inherentes de cada componente particular. Para más información sobre cada caso en particular, se recomienda la lectura de la ayuda incluida en el compilador de Borland C++ Builder, pues son los mismos componentes.

De izquierda a derecha, podremos observar las siguientes paletas de componentes:

- Estándar
- Adicional
- Win32
- Sistema
- Base de Datos
- Acceso a Datos
- Quick Report
- Diálogos
- Thyssen-Krupp layout

El primer icono de cada paleta siempre es el de selección (cursor normal del ratón) y se encuentra seleccionado por defecto mientras no se seleccione ninguno de los otros en la paleta. En ese caso, al pulsar sobre el formulario, se muestran las propiedades del elemento seleccionado. A continuación comentaremos cada paleta individualmente.

LA PALETA “ESTÁNDAR”

Empezando de izquierda a derecha, la primera paleta que se puede contemplar en la llamada “Estándar”, como se muestra en la Ilustración 75



Ilustración 75: La paleta de componentes "Estándar".

Estos componentes son los básicos y normales en una aplicación Windows, y son también los más frecuentemente usados, veamos:

Componente	Descripción
MainMenu	Crea una barra con un menú principal para el formulario. Para acceder a sus eventos o diseñar sus contenidos, basta con hacer un doble clic sobre uno de estos elementos, a fin de abrir una ventana de diseño de Menú
PopupMenu	Crea Menús de tipo “Popup” que aparecen cuando el usuario pulsa el botón derecho del ratón. Se puede usar el mismo mecanismo que en el anterior componente para

	diseñarlos
Label	Muestra texto que el usuario no puede seleccionar o manipular, como títulos o etiquetas de controles
Edit	Muestra un área de edición donde el usuario puede introducir o modificar una línea de texto simple
Memo	Muestra un área de edición donde el usuario puede introducir múltiples líneas de texto
Button	Crea un botón que el usuario puede pulsar a fin de iniciar acciones
CheckBox	Muestra una opción que el usuario puede activar / desactivar. Su uso más común es el de mostrar grupos de opciones que no son mutuamente exclusivas
RadioButton	Como el anterior, pero en este caso las opciones son mutuamente excluyentes
ListBox	Muestra una lista de opciones seleccionables con barra de desplazamiento
ComboBox	Muestra una lista de opciones en una combinación de un TListBox y un TEditBox. Los usuarios pueden escribir texto en el campo de edición o seleccionar un ítem de la lista
ScrollBar	Facilita una manera de cambiar el área visible de una lista o un formulario. Se puede usar también para moverse entre un rango de valores en incrementos
GroupBox	Facilita un contenedor de opciones relativas a un grupo o formulario
RadioGroup	Crea un TGroupBox que contiene TRadioButtons en un formulario
Panel	Crea un panel que puede contener otros componentes del formulario. Se pueden usar, por ejemplo, para crear barras de herramientas o líneas de estado
ActionList	Sirve para crear colecciones de acciones que centralizan la respuesta del formulario ante acciones del usuario

LA PALETA “ADICIONALES”

La siguiente paleta es la de componentes “Adicionales”, es decir componentes cuyo uso es más raro y especializado que los de la paleta “Estándar”, pero aun así muy frecuentemente usados en interfaces de aplicaciones Windows. Su aspecto es el que se muestra en la Ilustración 76



Ilustración 76: La paleta de componentes "Adicional".

Los componentes que contiene, brevemente:

Componente	Descripción
BitBtn	Crea un botón en el que se puede mostrar un bitmap

SpeedButton	Crea un botón que puede mostrar un icono, pero sin texto. Suelen agruparse en ToolBars
MaskEdit	Provee un mecanismo para forzar al usuario a entrar texto en un determinado formato dentro de un TEdit u otro control de texto
StringGrid	Crea una tabla donde se pueden mostrar cadenas con información organizada en filas y columnas
DrawGrid	Crea una tabla donde se pueden mostrar datos en filas y columnas
Image	Muestra una imagen (bitmap, icono o metafile)
Shape	Dibuja figuras geométricas (elipses, círculos, rectángulos, cuadrados o rectángulos redondeados)
Bevel	Crea líneas o cajas de apariencia tridimensional
ScrollBar	Crea un contenedor de tamaño variable que automáticamente muestra barras de desplazamiento cuando es necesario
CheckListBox	Muestra una lista con barras de desplazamiento, similar a una ListBox, excepto en que cada elemento de la lista tiene un CheckBox a su lado
Splitter	Añade un separador al formulario, entre dos controles alineados, permitiendo al usuario cambiar el tamaño que ocupan en tiempo de ejecución, simplemente moviendo el separador
StaticText	Crea una etiqueta de texto no editable, al igual que una Label, con la diferencia de que este componente tiene su propio "handle" a la ventana a la que pertenece
ControlBar	Crea un manager de layouts para los componentes ToolBar

LA PALETA "WIN32"

La siguiente paleta muestra componentes un poco más avanzados, que permiten crear interfaces más vistosas y complejas, tal como se ve en la Ilustración 77



Ilustración 77: La paleta de componentes "Win32".

Componente	Descripción
TabControl	Similar a un divisor en un Notebook, este componente provee un conjunto de solapas mutuamente excluyentes al estilo de un notebook
PageControl	Crea un control de páginas que es usado para definir una caja de diálogo multipágina, aunque usando una única ventana
ImageList	Este elemento es una colección de imágenes de las mismas dimensiones, cada una de las cuales puede luego ser referida por un índice.

RichEdit	Crea un control que contiene texto enriquecido. Por defecto soporta propiedades tales como: tipo, estilo, tamaño y color para sus fuentes. También soporta alineamiento de texto, numeración y drag & drop
TrackBar	Esta es una barra que define una extensión o rango de un ajuste, y un indicador que muestra el valor actual y permite cambiarlo. Se puede establecer su orientación horizontal o verticalmente, definir su longitud, anchura o el indicador de posición o muchos otros parámetros
ProgressBar	Una barra rectangular que se puede “rellenar” de izquierda a derecha, indicando un progreso en una operación o proceso
UpDown	Crea dos botones, uno con una flecha hacia arriba y el otro hacia abajo, que incrementan o decrementan valores al ser pulsados
HotKey	Hace que una “Hot Key” (combinación de teclado) se asocie a un control en particular
Animate	Este control proporciona un medio para mostrar una animación AVI en una parte del formulario, que se estará reproduciendo cuando lo planifiquemos
DateTimePicker	Muestra un control de edición para introducir fechas o tiempos. El usuario puede seleccionar fechas de una calendario mediante el uso de las teclas arriba, abajo o simplemente escribiendo la fecha. Es necesario disponer de una versión actualizada de COMCTL32.DLL, normalmente localizada en el directorio Windows\System o en Windows\System32
MonthCalendar	Muestra un calendario que representa un mes. Puede ser configurado para permitir navegar entre las fechas
TreeView	Permite mostrar y controlar un conjunto de objetos basándose en sus relaciones lógicas de jerarquía. El control incluye botones que permiten expandir / contraer ramas. Se usa principalmente para mostrar dichas relaciones entre el conjunto de objetos ordenados de forma jerárquica
ListView	Permite mostrar una lista en formato de varias columnas. Cada columna mostrando un aspecto del dato que representa la fila
HeaderControl	Muestra una cabecera de texto o números sobre una lista. Se puede dividir el control en dos o más partes para permitir cabeceras en múltiples columnas. Los títulos se pueden alinear de varias maneras, configuradas por separado
StatusBar	Éste es un área donde colocar texto informativo sobre las acciones que lleva a cabo la aplicación.
ToolBar	Maneja ToolButtons y otros controles, colocándolos por filas, y ajustando automáticamente su tamaño y posición
CoolBar	Muestra una colección de controles de tipo TCoolBand, cuyas bandas son móviles y de tamaño variable. El usuario puede posicionar los controles arrastrando la etiqueta que tienen en su parte izquierda
PageScroller	Contiene otros objetos en un área cliente sobre los que se puede hacer desplazamiento horizontal o verticalmente

LA PALETA "SISTEMA"

La siguiente paleta contiene componentes que permiten acceso a componentes propios del sistema operativo. En general no son componentes visuales, véase la Ilustración 78

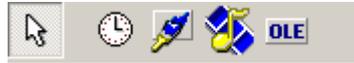


Ilustración 78: La paleta de componentes "Sistema".

Componente	Descripción
Timer	Este componente es un no-visual y lanza un evento de forma periódica. Tanto el evento como el periodo son definidos por el programador
PaintBox	Especifica un área rectangular que provee límites para el repintado de la aplicación
MediaPlayer	Muestra un control con estilo VCR que permite reproducir ficheros multimedia
OleContainer	Crea un área cliente para un Objeto Linkado y Embebido (OLE)

LA PALETA "BASE DE DATOS"

En la paleta "Base de Datos" podemos observar componentes que permiten interactuar con una base de datos Oracle, permitiendo acceso a sus tablas, conexiones, etc., esto confiere una gran potencia al layout que diseñamos, pudiendo incluso llegar a integrar la aplicación de control con la de gestión del almacén. La paleta se puede observar en la Ilustración 79.



Ilustración 79: La paleta de componentes "Base de Datos".

Componente	Descripción
OracleSession	Este componente provee un enlace con una base de datos Oracle. Tras establecer sus propiedades conexión (nombre de usuario, contraseña y base de datos), se puede establecer una conexión con la base de datos llamando al método "LogOn" o estableciendo la propiedad "Connected" a True. Si se desea que sea el usuario final el que especifique los parámetros de conexión, se puede hacer uso del componente OracleLogon
OracleQuery	Se usa este componente para ejecutar sentencias SQL o bloques PL/SQL. Es necesario establecer la propiedad Session a un OracleSession en el cual se ejecutará la consulta. El texto de la consulta puede establecerse en la propiedad SQL y ser ejecutado al invocar al método "Execute". Si la sentencia es de tipo SELECT, se podrá acceder a los campos de un registro mediante el método "Field", y llamar a "Next" para obtener más registros hasta que el retorno de "Eof" indique que ya no hay más registros disponibles. Si ocurriera un error durante la ejecución de la sentencia, se lanzará una excepción "EOraclError"
OracleDataSet	Este componente es útil cuando se quiere hacer uso de otros componentes dependientes de datos, puesto que deriva del estándar DataSet, heredando la mayoría de sus propiedades, métodos y eventos, junto con otros derivados del

	<p>componente TOracleQuery.</p> <p>Para usar un TOracleDataSet, hay que establecer su propiedad "Session" a un componente OracleSession del formulario y establecer la sentencia SELECT que obtendrá el conjunto de datos con el que trabajar</p>
OracleLogon	<p>Este componente provee de un cuadro de diálogo estándar para establecer el usuario, contraseña y cadena de conexión a una base de datos</p>
OraclePackage	<p>Un paquete es una extensión de las bases de datos Oracle que permite el encapsulamiento de funciones, procedimientos, definiciones de tipos, variables y constantes en una única extensión de programa. Esta unidad puede ser luego usada de forma arbitraria desde cualquier punto de un código PL/SQL.</p> <p>Este componente permite el acceso directo a las funciones, procedimientos, etc. Definidas en un paquete. Simplemente hay que establecer la propiedad "Session" y especificar el nombre del paquete en la Base de Datos. Para llamar a procedimientos o funciones de la base de datos, se pueden usar los métodos "CallProcedure" y "CallFunction". Para acceder a variables o constantes, se puede usar los métodos "SetVariable" o "GetVariable"</p>
OracleEvent	<p>Este componente puede ser usado desde una aplicación que necesita reaccionar a señales dbms_alert o mensajes dbms_pipe. Estos son típicamente generados por disparadores en la base de datos que pasan informaciones a otras sesiones de la base de datos. Estos componentes corren en un hilo de ejecución separado en background, sin interferir con el funcionamiento normal del programa. Cuando ocurre un evento, se llama a la función definida en la propiedad "OnEvent", que está sincronizada con el hilo de ejecución principal de la aplicación</p>
OracleScript	<p>Este componente permite definir y ejecutar un script SQL con múltiples sentencias SQL. Esto puede ser muy útil si se necesitara ejecutar muchas sentencias SQL que no pueden ser usadas en bloques PL/SQL, como por ejemplo el caso típico de scripts de instalación</p>
OracleDirectPathLoader	<p>Permite a una aplicación acceder al motor del "Direct Path Load" de un servidor Oracle, que permite al servidor cargar datos desde una fuente externa (un fichero, por ejemplo), a la base de datos a gran velocidad, en lugar de usar sentencias SQL INSERT o UPLOAD</p> <p>Hay varias restricciones para que dicha carga pueda ser llevada a cabo:</p> <ul style="list-style-type: none"> No se permiten disparadores en la tabla, es necesario desactivarlos temporalmente. No se permiten comprobaciones de restricciones o restricciones de claves ajenas en la tabla, aunque si se permiten restricciones de clave primaria No se pueden cargar tablas remotas. No se permiten tipos definidos por el usuario para la tabla. <p>Este interface sólo está disponible en las versiones Net8 8.1 cliente y posteriores.</p> <p>La operación de carga no forma parte de una transacción "normal"</p>
OracleNavigator	<p>Este componente es usado para moverse entre los datos contenidos en un OracleDataSet y permite realizar operaciones sobre dichos datos, tales como insertar nuevos, modificar o borrar datos existentes</p>
DataSource	<p>Este componente actúa como un conducto o paso intermedio entre componentes de</p>

	tipo DataSet y otros componentes dependientes de datos, como un DBGrid
--	--

LA PALETA "ACCESO A DATOS"

Una vez que tenemos componentes que acceden a la base de datos, podemos usar los componentes de esta paleta para mostrarlos de muy variadas formas.



Ilustración 80: La paleta de componentes "Acceso a Datos".

Componente	Descripción
DBGrid	Componente dependiente de datos, en forma de tabla. Permite visualizar y editar los datos en un formato tabular, similar a una hoja de cálculo
DBNavigator	Componente dependiente de datos en forma de una barra de botones que permite moverse por los registros de una tabla de la Base de Datos. también puede poner la tabla en modo de inserción, edición o sólo lectura, colocar nuevos registros o refrescar los que se están mostrando
DBText	Componente dependiente de datos en forma de etiqueta de texto que muestra los valores de un campo de un registro
DBEdit	Componente dependiente de datos en forma de campo de edición (Edit). Muestra los datos de un campo de un registro de la BD
DBMemo	Componente dependiente de datos en forma de memo que muestra o edita textos en un registro de la Base de Datos
DBImage	Componente dependiente de datos con la misma apariencia de un Image que muestra, corta, copia o pega bitmaps desde o hacia un registro de la Base de Datos
DBListBox	Componente dependiente de datos en forma de lista que muestra un lista con desplazadores de valores de una columna en una tabla de la Base de Datos
DBComboBox	Componente dependiente de datos en forma de ComboBox que muestra o edita listas de valores de una columna en una tabla de la Base de Datos
DBCheckBox	Componente dependiente de datos en forma de CheckBox que muestra o edita campos boléanos de un registro de la Base de Datos
DBRadioGroup	Componente dependiente de datos en forma de grupo de RadioButtons que muestran un conjunto de valores de columnas de una tabla de la Base de Datos
DBLookupListBox	Componente dependiente de datos, en forma de ListBox, que deriva su lista de objetos a representar de un campo Lookup definido para un DataSet o un DataSource. En cualquier caso, al usuario se le presenta una lista restringida de valores sobre los que elegir para establecer un campo
DBLookupComboBox	Componente dependiente de datos, en forma de ComboBox, que deriva su lista de objetos a representar de un campo Lookup definido para un DataSet o un DataSource. En cualquier caso, al usuario se le presenta una lista restringida de

	valores sobre los que elegir para establecer un campo
DBCtrlGrid	Este componente muestra múltiples campos de múltiples registros en formato de tabla tabulada. Cada celda en la tabla muestra múltiples campos de un sólo registro
DBRichEdit	Un control de edición multilínea que puede mostrar y editar texto enriquecido de un campo en un DataSet

LA PALETA “QUICK REPORT”

Esta paleta contiene elementos que nos permitirán diseñar informes sobre objetos en base de datos y / o campos estáticos, listos para poder imprimirse o visualizarse por parte del usuario final. Su aspecto puede verse en la Ilustración 81.



Ilustración 81: La paleta QuickReport

Componente	Descripción
QuickRep	El formulario de informe básico a partir del cual se pueden construir los informes personalizados. Es un componente visual que toma la forma de una página impresa en el tamaño de papel seleccionado actualmente. Solo hay que ir soltando componentes sobre ella para componer un informe y enlazarlo a una base de datos si fuese el caso.
QRSubDetail	Enlaza Datasets en el informe. Típicamente se establece una relación maestro / detalle entre una tabla o consulta y este componente.
QRStringsBand	Coloca una banda conteniendo cadenas de texto en el informe.
QRBand	Coloca una banda en el informe y establece la propiedad “BandType” a alguno de los comportamientos predefinidos durante la generación final del informe.
QRChildBand	Si se necesita crear bandas con componentes que se expandan y se desea que otros componentes se desplacen hacia abajo de acuerdo a estos últimos, se usa este componente y se ponen los componentes a mover en el. También es útil si se tienen bandas muy largas que pueden ocupar varias páginas.
QRGroup	Permite crear un grupo de bandas juntas provee control sobre encabezados, pies de página y fines de página.
QRLabel	Imprime texto fijo en el informe. El texto a mostrar se establece en la propiedad “Caption”. Se puede dividir el texto en varias líneas e incluso en varias páginas.
QRDBText	Como la anterior, permite introducir texto en el informe, pero esta vez el texto se obtiene directamente de una Base de Datos, solo hay que establecer las propiedades “DataSource” y “DataField”. Al contrario que otros controles dependientes de Bases de Datos, este componente funciona incluso cuando no están establecidas dichas propiedades, lo cual mejora la velocidad.
QRExpr	Imprime campos de una base de datos, cálculos y texto estático. Solo hay que introducir una expresión válida en la propiedad “Expression”.

QRSysData	Imprime información del sistema, tal como título del informe, número de pagina actual y demás. Hay que seleccionar los datos a imprimir en la propiedad "Data". Cualquier otro texto que se desea preceda a estos datos, debe ir en la propiedad "Text".
QRMemo	Imprime una gran cantidad de texto que no viene de una base de datos. Puede ser texto estático o se puede cambiar durante la generación del informe. Se puede establecer que el campo se expanda verticalmente según sea necesario, e incluso entre varias páginas.
QRExprMemo	Permite programar la generación de expresiones usando expresiones QuickReport.
QRRichText	Permite introducir texto en formato RichText en el formulario.
QRDBRichText	Provee a QuickReport de un envoltorio para acceder a campos DBRichText en el informe.
QRShape	Dibuja formas geométricas simples, como rectángulos, círculos y líneas en el informe.
QRImage	Muestra una imagen en el informe. Soporta todos los formatos de imagen soportados por los componentes "Picture".
QRDBImage	Imprime imágenes almacenadas como campos binarios (BLOB) en una base de datos.
QRCompositeReport	Permite combinar más de un informe junto.
QRPreview	Muestra un formulario que permite al usuario final realizar una previsualización del informe e imprimirlo si así lo desea.
QRTextFilter	Permite exportar los contenidos de un informe a formato texto.
QRHTMLFilter	Permite exportar los contenidos de un informe a formato HTML.

LA PALETA "DIÁLOGOS"

Cuando necesitemos crear diálogos que soliciten información de algún tipo al usuario, usaremos los componentes de la paleta "Diálogos", que se puede contemplar en la Ilustración 82



Ilustración 82: La paleta de componentes "Diálogos".

Componente	Descripción
OpenDialog	Muestra un diálogo de apertura de ficheros, muy común en todas las aplicaciones Windows
SaveDialog	Muestra un diálogo para seleccionar un nombre de fichero para salvar información

OpenPictureDialog	Muestra un diálogo modal para seleccionar y abrir un archivo de imágenes. Es idéntico a un TOpenDialog, excepto que tiene un área donde poder realizar una vista previa de la imagen antes de cargarla
SavePictureDialog	Idéntica al anterior, pero el propósito ahora es salvar una imagen
FontDialog	Muestra una ventana desde la que el usuario puede especificar el tipo, estilo y tamaño de una fuente y seleccionarla
ColorDialog	Este diálogo permite al usuario especificar información sobre un color determinado
PrintDialog	Muestra un cuadro de diálogo que permite al usuario especificar información sobre una impresión, tal como la impresora donde realizarla y el número de copias a realizar
PrinterSetupDialog	Muestra un diálogo de configuración de impresoras que permite al usuario cambiar y parametrizar sus impresoras
FindDialog	Muestra un diálogo de búsqueda de cadenas en un texto dado
ReplaceDialog	Muestra un diálogo que permite realizar las operaciones de buscar y reemplazar cadenas en un texto

LA PALETA “THYSSEN-KRUPP LAYOUT”

Por último, la paleta “Thyssen-Krupp layout” contiene elementos tanto visuales como no visuales, y estos son el verdadero corazón del que partir para crear nuestro layout. Podemos ver estos componentes en la Ilustración 83.



Ilustración 83: La paleta de componentes "Thyssen-Krupp Layout".

A continuación haremos una breve descripción de cada uno de los componentes, para después describir sus propiedades.

Componente	Descripción
LEDLabel	Un led, con una etiqueta adjunta, asociado a una variable booleana, que aparecerá en color rojo o verde según el valor de la variable sea false o true
AnalogMeter	Este componente emula un medidor analógico y puede ser asociado a una variable definida en el simbólico para mostrar su valor numérico
HyperBMP	Este componente es muy similar a los Image, básicamente es un contenedor para una imagen.
IAeverLiquidIndicator	Este componente emula una pantalla de cristal líquido, que va mostrando un valor numérico asociado a una variable cualquiera definida en el simbólico.
LCDScreen	Este componente no está asociado a ninguna variable del simbólico, simplemente sirve para mostrar cadenas o mensajes de forma vistosa debido a la gran variedad

	de efectos que permite
LED	Un simple led, asociado a una variable booleana, que aparecerá en color rojo o verde según el valor de la variable sea false o true
TimerTick	<p>Este componente es el más sencillo y a la vez más importante de un formulario para el Status Monitor. Mediante él se define el “pulso de refresco” que lleva asociado el formulario, es decir, la frecuencia con la que refrescarán los valores que mantiene en una caché interna y que representan los valores actuales de las variables del simbólico en la tarjeta. El refresco de esta caché provoca el refresco de todos los componentes asociados a variables en el simbólico que se encuentran activos en el formulario.</p> <p>Para activar este refresco periódico, basta con poner la propiedad “Enabled” a true, y pasará a modo runtime. Si ocurriese algún problema en la conexión con el sistema remoto, o con la extracción de datos, automáticamente este componente se desactivará, siendo necesario volver a activarlo desde el modo edición si queremos volver a ver los valores que corren en el sistema remoto. Es importante también fijarse en el valor que tiene la propiedad “Interval”, ya que define la frecuencia (en milisegundos) con que se refresca la caché (por defecto se establece a 1 segundo). Valores menores pueden llevar a que el sistema no pueda obtener los datos con la suficiente velocidad para alimentar estas peticiones, mientras que valores muy superiores provocarán la falsa apariencia de que el sistema remoto funciona “a saltos”.</p> <p>¡Debe existir al menos un componente de este tipo en el formulario para que las comunicaciones actuen!.</p>
LinkedElement	Este componente es esencial para construir el grafo de trayectorias posibles en la estación, mediante su uso, el programador debería ser capaz de reflejar en el layout todas las posibles trayectorias sobre las que se quiere tener control.
BoolVarSetter	Este componente tiene un funcionamiento muy similar al CheckBox de la paleta Estándar, con la diferencia que está ligado a una variable de tipo BOOL definida en el simbólico y al cambiar su estado, el usuario fuerza también la variable que se está viendo desde el sistema de control Galileo
RotateLabel	Este componente no está asociado a variables del simbólico. Permite mostrar texto al igual que un Label, con posibilidades ampliadas de rotarlo en vertical o añadir efectos que lo hagan más vistoso
TrasloScroller	Componente que simula un elemento móvil. Tiene capacidad para mostrar hasta 10 señales asociadas a la máquina, así como realizar una simulación del movimiento real llevado a cabo por esta.
dsZoomNavigator	Este componente permite incluir un cuadro de navegación rápido para paneles muy grandes. Aparece como un pequeño panel con un área seleccionable que se puede asociar con un ScrollBox de tamaño mucho mayor que el que permita la pantalla. Al pulsar sobre esta área, nos permitirá desplazarnos de forma rápida por el ScrollBox de forma automática.

Estos componentes comparten muchas propiedades con los ya conocidos y documentados por Borland, así que a continuación nos centraremos en comentar aquellas propiedades exclusivas de este grupo o específicas de alguno de ellos. Haremos especial hincapié en las que son más importantes o permiten más posibilidades a cada componente. El componente BoolVarSetter no aparece listado porque sus únicas características extra son las comunes, que son descritas al principio.

Con la excepción del TimerTick, el LCDScreen, el dsZoomNavigator y el RotateLabel, todos los componentes de esta paleta tienen las siguientes propiedades comunes:

<i>Propiedad</i>	<i>Descripción</i>
Clock	Esta es una propiedad especial que no debería ser modificada en ningún momento por el programador. Se usa en conjunción con el TimerTick para el refresco de los valores que está monitorizando el componente
Secuenciador	Puede ser establecida a un nombre válido de Máquina o de Zona definido en el simbólico. Si no se establece a ningún valor, se sobreentiende que se hace referencia a la zona GLOBAL
Variable	<p>Puede ser establecida a cualquier variable de los tipos BOOL, BYTE, WORD o DWORD definidas en el simbólico. Si no se establece, se entiende que el componente no está ligado a ninguna variable, y por tanto, no monitorizarán sus valores. También es posible, en lugar de una variable, establecer una condición, usando operadores de condicion AND, OR, XOR u operadores matemáticos. Si no se prefijan las variables, se entenderán que pertenecen a la máquina definida con la propiedad “Secuenciador”, mientras que si queremos especificar una variable de otra máquina, deberemos escribirla en su forma completa: “Maquina.Variable” Por ejemplo, si el secuenciador es TRE_01, y en la variable aparece:</p> <p>ActivarEntrada AND ActivarSubir</p> <p>Esto se entiende como que que estamos evaluando</p> <p>TRE_01.ActivarEntrada AND TRE_01.ActivarSubir</p> <p>Sin embargo, podemos poner tambien:</p> <p>ActivarEntrada AND TRE_02.ActivarSalida</p> <p>Que se entenderá como:</p> <p>TRE_01.ActivarEntrada AND TRE_02.ActivarSalida</p> <p>Estas expresiones son susceptibles de parentización.</p>

- Propiedades de LEDLabel

<i>Propiedad</i>	<i>Descripción</i>
InvertLogic	Esta propiedad sirve para invertir la lógica de representación del componente, es decir, que si se establece a true esta propiedad, el led se encenderá cuando la variable que monitoriza se encuentre a false, y se apagará cuando la variable contenga el valor de true
Lit	Esta propiedad sirve para forzar el encendido del led. Cuando se establece a true, el led aparece encendido, sin importar el valor de la variable que monitoriza

- Propiedades del AnalogMeter

<i>Propiedad</i>	<i>Descripción</i>
------------------	--------------------

AngularRange	Define la forma del medidor analógico. El indicador será un sector angular cuyo arco serán los grados marcados como valor de esta propiedad, hasta 360, que corresponderían a una circunferencia completa
EnableZoneEvents	El indicador permite definir unas zonas mínimas o máximas por debajo o encima de las cuales se pueden disparar eventos que alerten de que la variable se encuentra en dichos umbrales. Estableciendo esta propiedad a true, podremos hacer uso de dichos eventos

- Propiedades del HyperBMP

<i>Propiedad</i>	<i>Descripción</i>
Image	Corresponde a la imagen que queremos representar, que se puede cargar desde un fichero de imágenes (BMP, ICO o Metafile)
Height	Altura total del área donde se dibujará la imagen. Esta altura no tiene porque coincidir con el tamaño de la imagen cargada en la propiedad Image
Width	Anchura total del área donde se dibujará la imagen. Esta altura no tiene porque coincidir con el tamaño de la imagen cargada en la propiedad Image
Stretch	Si esta propiedad se establece a true, la imagen cargada en la propiedad Image se intentará ajustar al tamaño definido por las propiedades Height y Width, con lo que realizaremos un zoom de dicha imagen
InvertLogic	El HyperBMP puede llevar asociada una variable definida en el simbólico de tipo BOOL, y si esta variable es true, la imagen se mostrará, en cambio, si es false, la imagen no se mostrará. Si se establece esta propiedad a true, la lógica de visualización se invertirá, al igual que se hacía con los LED

- Propiedades de IAEverLiquidIndicator

<i>Propiedad</i>	<i>Descripción</i>
IndicatorAngle	Si se desea que el indicador este situado en un ángulo inclinado, se puede aumentar el valor de esta propiedad (0 → Horizontal, 900 → Vertical)
IndicatorGap	Indica el espacio (en píxeles) que queda entre el borde del indicador y los indicadores de valor
IndicatorKind	Según el tipo, podemos usar un indicador 7 segmentos (por defecto), 9 segmentos, o como una matriz de puntos
Indicatorstring	Esta es la cadena que se está mostrando en el indicador. Si el tipo del indicador es matriz de puntos, se muestra cualquier cadena, pero con los tipos de 7 y 9 segmentos, sólo se pueden mostrar números y los caracteres a, b, c, d, e, f (para números en hexadecimal).
IndicatorStyle	Mediante esta propiedad podemos hacer más gruesas o delgadas las líneas que forman los números o mensajes

SignNumber	Indica en que posición va el signo. Como el signo siempre se coloca en la posición más a la izquierda del indicador, esta propiedad también indica cuantos dígitos son visibles en el indicador
-------------------	---

- Propiedades de LCDScreen

<i>Propiedad</i>	<i>Descripción</i>
DotMatrix	Mediante esta propiedad podemos elegir entre diferentes tipos de pantallas de matriz de puntos, lo que aumenta las posibilidades de visualización en pantalla
Intensity	Esta propiedad aumenta la intensidad de brillo de los puntos en la pantalla LCD. A mayor valor, más brillantes aparecerán. El rango varía entre 0 y 255
Lines	Este es el mensaje que aparecerá en la pantalla LCD, puede estar compuesto por varias líneas, sobre las que se puede ir desplazando de forma periódica
ScrollActive	Al marcar esta variable a true, la pantalla LCD comienza a realizar el desplazamiento definido por las propiedades correspondientes
ScrollHorz	Indica la cantidad de desplazamiento que se realiza en horizontal (en píxeles) de cada vez. Un valor positivo implica un desplazamiento de izquierda a derecha, y uno negativo de derecha a izquierda
ScrollSpeed	Indica la velocidad del desplazamiento. A mayor valor, más rápidamente se realizará el desplazamiento
ScrollVert	Indica la cantidad de desplazamiento que se realiza en vertical (en líneas) de cada vez. Un valor positivo implica un desplazamiento de abajo hacia arriba, y uno negativo de arriba hacia abajo

- Propiedades de LED

<i>Propiedad</i>	<i>Descripción</i>
InvertLogic	Idéntico al LEDLabel
Lit	Idéntico al LEDLabel

- Propiedades de LinkedElement

<i>Propiedad</i>	<i>Descripción</i>
Hint	Esta propiedad es el texto que aparecerá flotando sobre el nodo cuando situemos el cursor del ratón sobre dicho nodo

Entrada1	Un nodo típicamente estará conectado a otros para formar el grafo o camino, y mediante estas 7 propiedades podremos especificar hasta 7 nodos (Componentes LinkedElement también, ya existentes en el formulario), que serán la entrada a este elemento. Esta es una información muy importante, pues es en la que se basa el generador de código para expandir los elementos this, ant1, .. ant7 que coloquemos en nuestro código
Entrada2	
Entrada3	
Entrada4	
Entrada5	
Entrada6	
Entrada7	
Image	Podemos seleccionar un Componente HyperBMP o Image para colocarlo aquí, y esa será la imagen que queramos que represente ese nodo
InvertLogic	Idéntico al HyperBMP
Peso	Esta propiedad está pensada para poder seleccionar diferentes rutas en tiempo real en un futuro, en función del peso que asignemos a cada nodo del grafo
Salida1	Un nodo típicamente estará conectado a otros para formar el grafo o camino, y mediante estas 7 propiedades podremos especificar hasta 7 nodos (Componentes LinkedElement también, ya existentes en el formulario), que serán la salida a este elemento. Esta es una información muy importante, pues es en la que se basa el generador de código para expandir los elementos this, pos1, .. pos7 que coloquemos en nuestro código
Salida2	
Salida3	
Salida4	
Salida5	
Salida6	
Salida7	
Stretch	Tiene el mismo comportamiento que en el HyperBMP
Tipo	Los nodos pueden ser de varios tipos según su posición en el grafo: Común, Especial, Inicio o Fin. Esta propiedad, al igual que la de Peso, está diseñada para futuras implementaciones, por lo que su uso aún no tiene ningún efecto.

- Propiedades de RotateLabel

<i>Propiedad</i>	<i>Descripción</i>
Escapement	Indica el ángulo en el que queremos colocar el texto (0 → horizontal, 90 → vertical)
TextStyle	Permite darle algún efecto extra al texto: elevado, incrustado o sin efecto

Scripting

Si el uso de componentes confiere grandes ventajas a la visualización, la posibilidad de usar “Scripts” para programarlos eleva su potencia hasta el punto de que la visualización del layout tiene las mismas posibilidades que el uso de un compilador comercial.

El lenguaje de programación usado es JavaScript, y desde él se puede hacer uso de los formularios, objetos y componentes que tenemos a nuestra disposición en la paleta de componentes, siguiendo las reglas de dicho lenguaje. No es el objetivo de este manual el explicar los entresijos de JavaScript como lenguaje de programación, por lo que en caso de que el lector tenga dudas en este aspecto, recomendamos la consulta de cualquier tutorial o manual sobre dicho lenguaje.

La ventana con nuestro código de script aparece en cuanto trabajamos con un formulario de layout y lo pongamos en modo diseño. En ese momento aparece la ventana de la Ilustración 84, pudiendo escribir nuestras funciones en ella. También, cuando hacemos doble clic sobre alguna de las propiedades de un componente (en la ventana de propiedades de componentes), se crea en dicha ventana una cabecera para la función que responderá a dicho evento, y se nos muestra, a fin de que rellenemos el cuerpo de la función.

Para borrar una función de respuesta a un evento, lo más cómodo es borrar el código de su cuerpo (sin comentarios siquiera), puesto que al salvar el formulario, se eliminan las funciones cuyo cuerpo se encuentre vacío, y se desligan de los eventos a los que estuvieran asignadas, al igual que pasa desde el IDE de los compiladores de Borland.

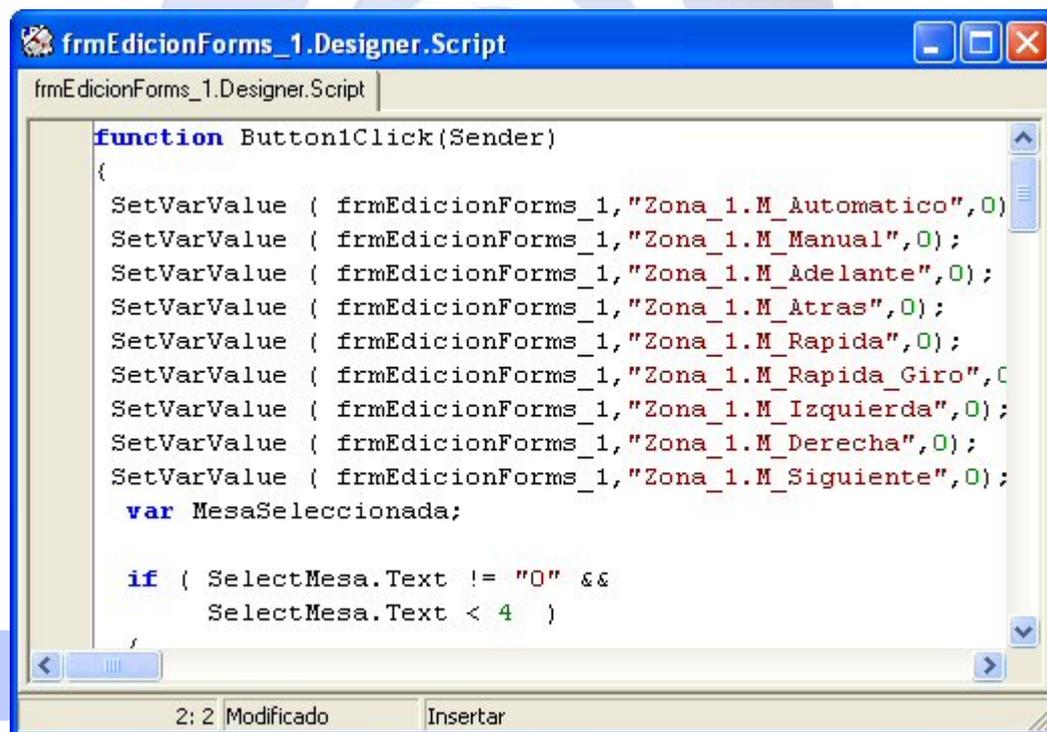


Ilustración 84: La ventana de edición de script.

Además de las funciones que incorpora el lenguaje JavaScript, también es posible usar una serie de funciones añadidas desde el entorno, como son:

- `void ChangeModeDesigner(TForm formulario);`
- `void ShowReportFailures(TForm formulario);`
- `int GetVarValue(TForm formulario, string variable);`
- `void SetVarValue(TForm formulario, string variable, int valor);`

El uso de las dos primeras es explicado en los apartados [El Galileo Status Monitor](#) y [Tratamiento de Históricos](#). Las dos últimas funciones pueden usarse para obtener y modificar los valores de las variables de programa que esté corriendo en un momento dado en el sistema Galileo. Las variables han de ser de tipo numérico, no se pueden obtener o modificar los valores de variables de tipo Machine, Componentes o Strings.



DESIGNER 3.0

Comunicaciones

Las comunicaciones son el corazón del sistema. Permiten ver como funcionan las partes implementadas, lo cual es la realimentación necesaria para poder desarrollar un proyecto de forma efectiva.

Una de las partes fundamentales del desarrollo de un sistema de control es la depuración del funcionamiento una vez programados los secuenciadores apropiados. Para lograr esto, es necesario que exista una comunicación entre el entorno de desarrollo y los computadores donde se está ejecutando el sistema de control. Esta comunicación, aparte de servir para depuración, como veremos en otro capítulo, nos permite el acceso a varios formularios que realizan operaciones que de otra manera no se podrían realizar y muestran información “en directo” de lo que está ocurriendo. Para iniciar la comunicación de un proyecto, es necesario usar la opción **Comunicaciones / Modo Online** del menú principal, o bien pulsar el botón correspondiente de la barra de herramientas. Del mismo modo, cuando no queramos ya seguir obteniendo esa información o no necesitemos acceder a dichas opciones, podemos abandonar la comunicación mediante el menú **Comunicaciones / Modo Offline** del menú principal.

Las comunicaciones se establecen de forma puntual (P2P con las computadoras que hayamos definido en el proyecto). Ha de tenerse en cuenta que dichas computadoras deben estar accesibles por TCP/IP desde la red donde se esta ejecutando el entorno de desarrollo, y que los nombres de computadora son sensibles al uso de mayúsculas y minúsculas, por lo que debemos tener cuidado a la hora de nombrarlas.

Al ponernos en el Modo Online, podemos obtener los datos de las máquinas en ejecución, así como usar el editor de tracking, las plantillas de estatus de variables y la transferencia tras la compilación de un proyecto. Además de todo esto, también se nos muestran accesibles una serie de nuevas posibilidades:

Ver red Galileo

Esta ventana se encarga de mostrarnos un panel de control desde el que podemos realizar todas las tareas “extra” que podemos realizar con las computadoras remotas. La ventana está dividida en 3 partes, tal y como se ve en la Ilustración 85:

La parte inferior contiene un área de mensajes donde se nos irá informando como evolucionan o terminan los diversos comandos que invoquemos.

La parte central posibilita la transferencia y manejo de ficheros del computador remoto. Aparte del fichero de control de la instalación, también podemos querer transferir otros ficheros a la computadora

remota (ficheros de configuración, INIs o librerías necesarias para el buen funcionamiento de los componentes).

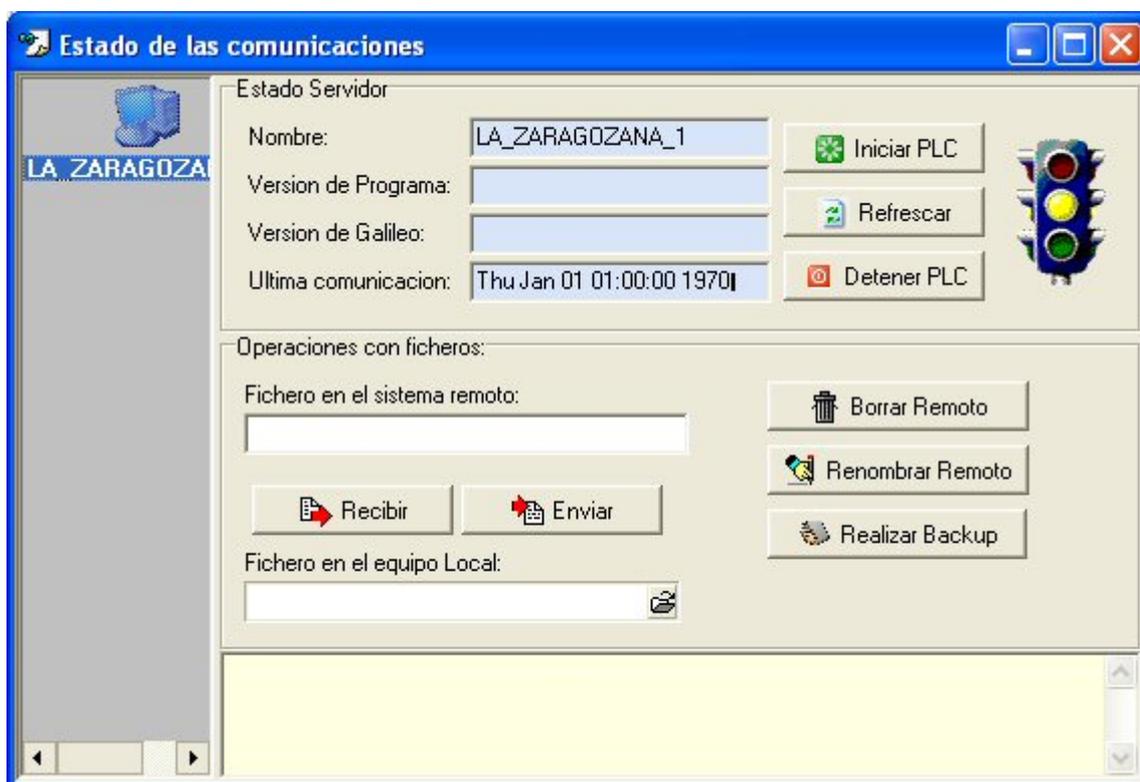


Ilustración 85: Ventana de operaciones sobre la red Galileo.

Para ello, deberemos seleccionar la computadora de la lista de la izquierda, y luego introducir el nombre del fichero en la computadora remota (path incluido) y el nombre del fichero local y usar los botones correspondientes.

- ** Recibir**: El fichero remoto será transferido a la computadora donde estamos trabajando y salvado con el nombre especificado como "Fichero Local". Si se produce un error durante la transferencia, un mensaje de error nos informará de tal evento en la parte inferior. Cuando la transferencia termine también se nos informará de este evento.
- ** Enviar**: El fichero de la computadora remota, cuyo nombre hemos especificado en el campo de texto, será transferido a nuestro ordenador, y salvado en el directorio seleccionado. Si se produce algún error, recibiremos un aviso sobre tal hecho.
- ** Borrar Remoto**: Se envía una petición al sistema remoto para que borre de su sistema de ficheros el archivo cuyo nombre especificamos en el campo de texto. Recibiremos un mensaje de error si el sistema no pudo realizar la operación.
- ***Renombrar Remoto***: aparecerá un nuevo diálogo solicitándonos el nuevo nombre del archivo en la estación remota, tras lo cual se enviará el comando pertinente a dicha estación para que intente renombrar ese archivo al nuevo nombre. Si no se consigue, aparecerá un mensaje de error indicando tal hecho.

-  **Realizar Backup.** Esto persigue la realización directa de una copia de seguridad de archivos del sistema remoto. Su utilidad es inmediata para la gente de mantenimiento, que quiere tener a salvo una copia de todos los ficheros de configuración necesarios para restablecer el sistema remoto en caso de fallo de la computadora. No tenemos que poner nada en los campos de fichero local o remoto. Al pulsar este botón aparecerá un diálogo que nos preguntará donde queremos dejar un fichero llamado “Backup.zip”. Dicho fichero contendrá:
 - El fichero .plc que esté corriendo en ese servidor en ese momento.
 - El fichero de configuración de bus “Galileo Control System.ini”
 - Todos los ficheros que aparezcan referenciados como cadenas de configuración en el programa de control (típicamente los .ini de configuraciones).

En la parte superior del plugin, en el cuadro “Estado Servidor”, tenemos también alguna información y algunas posibilidades de interactuar con la computadora seleccionada

Detener PLC: Esta opción envía un comando de control al Servicio de Control Galileo indicándole que detenga la ejecución del programa de control actual. El bus conectado a sus tarjetas será desactivado, y las salidas físicas se pararán. También se descargará el programa de control activo en ese momento de la memoria. Si el programa de control ya estaba parado, esta opción no tiene ningún efecto significativo si Galileo ya está parado.

Iniciar PLC: La acción opuesta a la anterior. Si el programa de control de la computadora remota está parado, se procede a iniciarlo, levantando el bus controlado por sus tarjetas e iniciando el ciclo de control de los transportadores que se gobiernan desde esa computadora. Si el programa de control estaba iniciado, primeramente se procede a pararlo (como en el caso anterior) y luego se continúa en el modo mencionado.

Refrescar. Refresca la información de los elementos “Versión del programa”, “Versión de Galileo” y “última comunicación”. Estos campos indican lo siguiente:

- **Versión del Programa:** Versión del programa cargado en memoria por el sistema de control. Esta versión es distinta para cada programa compilado por el entorno de desarrollo y puede servir para identificar una de ellas.
- **Versión de Galileo:** Se refiere a la versión del Sistema de control usada. No cambia a menos que instalemos una nueva versión de Galileo como servidor de control.
- **Última Comunicación:** se refiere a la fecha y hora de la última comunicación que se produjo entre el entorno de desarrollo y el programa de control

Además de estos elementos, también hay que destacar el semáforo de la derecha, que cambia de estado, mostrando sus diferentes tonos en función de lo que las comunicaciones digan del computador seleccionado.

Cuando el semáforo está en verde, significa que el computador remoto esta funcionando y ejecutando un programa de control que se muestra en las etiquetas correspondientes.

Si el semáforo esta en rojo, significa que no hay comunicación con el computador remoto (imposible acceder a él).

El semáforo en amarillo significa que no es posible obtener datos del computador remoto (posiblemente esté desconectado o Galileo no puede responder a las peticiones que se le hacen).

Ver averías

Desde el menú **Comunicaciones**, una vez se esté en modo Online, se puede activar o desactivar la opción **Ver Averías**.

Esto provoca la aparición de una ventana flotante como la que se puede ver en la Ilustración 86 en cuanto se active una avería.

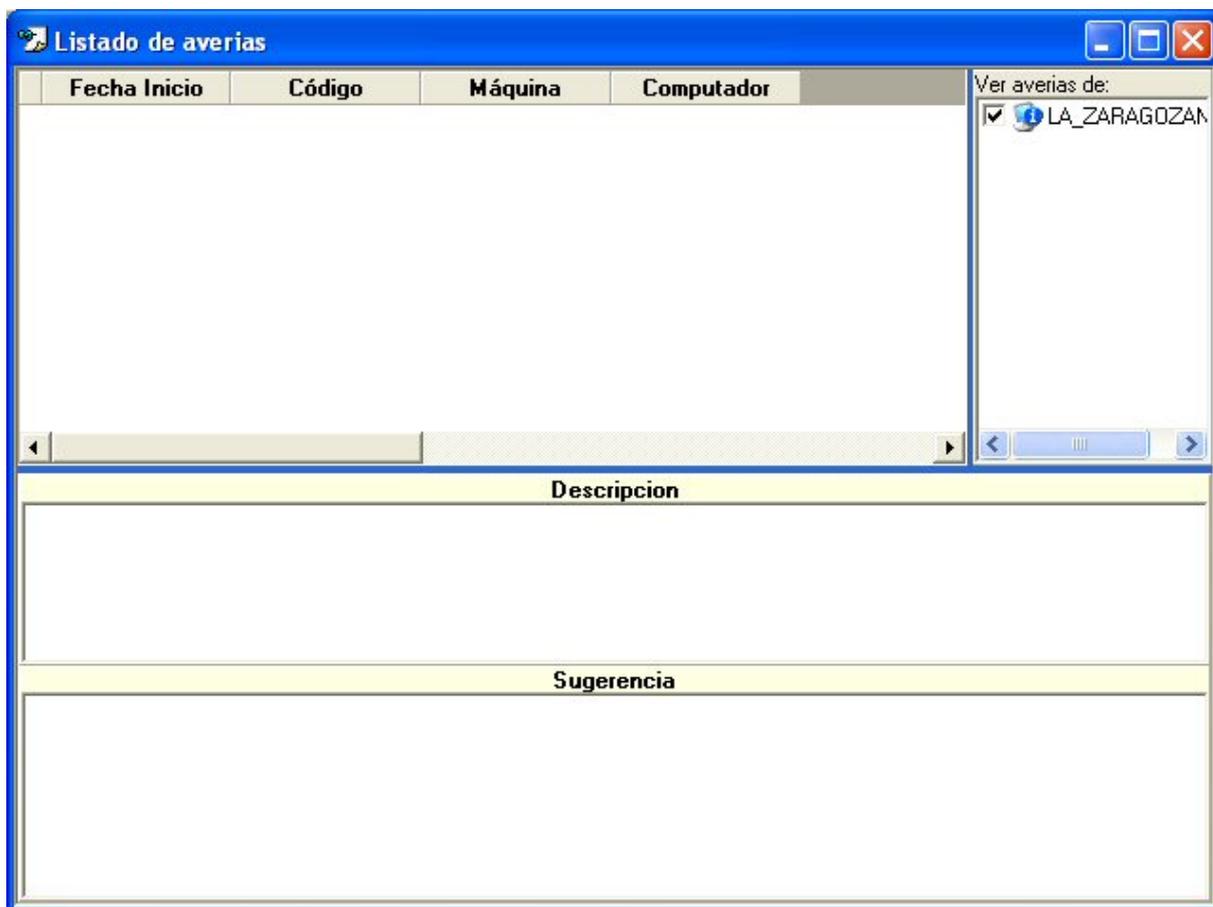


Ilustración 86: Ventana de aviso de averías.

Esta ventana se actualiza periódicamente cuando se produzca un cambio en la información que contiene mientras no se cierre o se pase a modo Offline. Si el equipo dispone de una tarjeta de sonido y altavoces, se podrá escuchar una sirena de alarma cuando alguna avería se active.

En el caso de que muchas averías se activen de forma simultanea, puede resultar difícil ver lo que nos interesa, por este motivo se puede aplicar un filtro a las averías activas, pulsando los marcadores asociados a las computadoras que se encuentran en la parte derecha de la ventana. Si se marcan, se mostraran las averías de dicha computadora. También es posible ordenar la lista de averías por sus

diversos campos, para ello sólo es necesario pulsar la cabecera de la columna por la que queremos que se ordene la tabla de averías activas.

El Pupitre Virtual

Consiste en un formulario independiente que muestra las señales básicas de control de una máquina. Dichas señales deberán haber sido definidas previamente mediante los parámetros de los secuenciadores, y al asociar los parámetros a las variables de las máquinas, se posibilitará el control de las mismas mediante el Pupitre Virtual. Al acceder a él, se mostrará un formulario como el de la siguiente ilustración:

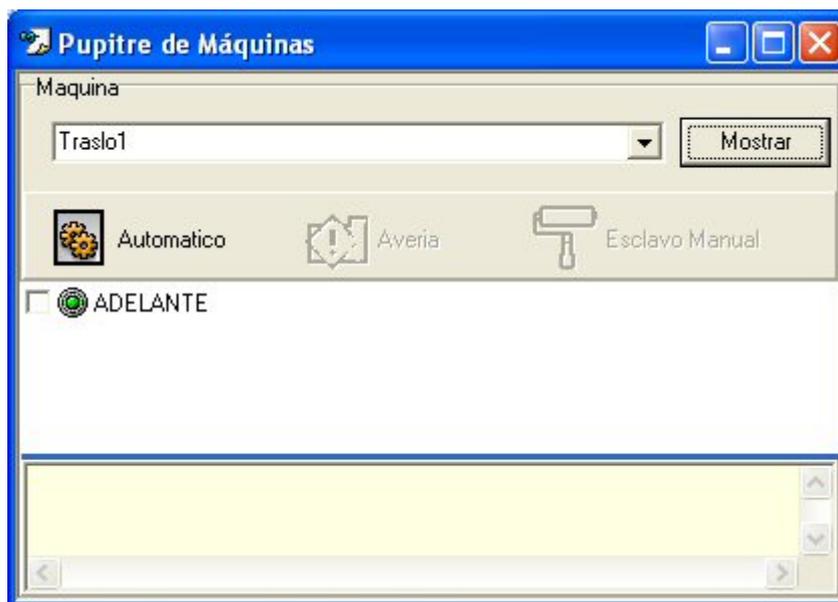


Ilustración 87: El pupitre Virtual

Para empezar a trabajar con una máquina, solo tenemos que seleccionarla de la lista y pulsar el botón “Mostrar”. En la lista solo aparecen máquinas que tengan asociadas las variables de “Manual” al parámetro correspondiente.

Como se observa, las señales de control principal (Manual, Averia y Esclavo Manual) aparecen como botones en una barra aparte. Si aparecen sombreadas, esto significa que dicha señal no está activa. La excepción es el botón de “Manual/Automático”, que muestra el valor contrario al valor actual de la variable “Manual”. Es decir, si la variable de “Manual” vale falso, se muestra el botón “Manual”, para indicar que al pulsarlo se forzará el valor de esa variable a cierto y se pasará (supuestamente) a un modo “Manual” del secuenciador. Si la variable tiene el valor “cierto”, el que se muestra, como en la ilustración, es el botón de “Automático”, indicando que al pulsarlo se pasará al modo “Automático” programado en el secuenciador.

Si alguno de los botones no aparece, significa que no se ha definido la señal correspondiente.

En la parte central aparecen todas las señales definidas para el secuenciador de la máquina seleccionada. Puede haber más, pero solo se muestran los que tengan una variable de tipo bit asociada al parámetro con señal definida.

El el panel inferior, se mostraran los avisos o errores que se produzcan como consecuencia de manipular las variables mediante el pupitre. Para forzar alguna de las variables basta con pulsar sobre la marca que aparece a su izquierda indicando su estado. Al hacerlo por primera vez, se nos pedira que nos identifiquemos mediante un usuario que disponga de permisos de usuario con perfil “básico”.



DESIGNER 3.0

Depuración

Una vez completada la programación “teórica”, es momento de depurar el programa, obteniendo la realimentación adecuada para poder solucionar los problemas que surjan al probar los diversos elementos a controlar.

La depuración de un programa es un proceso lento y casi siempre complejo, más aun si cabe si hablamos de depurar algo que funciona en tiempo real. En el capítulo anterior ya avanzamos que la depuración es posible desde el momento que podemos usar el modo “online”, ahora vamos a centrarnos en las opciones que tenemos en este modo para acometer esta tarea.

Estado de variables

Una vez que tenemos un sistema funcionando, suele ser necesario entrar en fase de depuración, lo cual podemos llevar a cabo desde la edición del layout, colocando visualizadores de las variables correspondientes. Esto puede ser impracticable si ya tenemos un layout bien definido y no queremos modificarlo (por el trabajo que esto implica, ya que luego habrá que deshacer los cambios), en lugar de esto, si lo que queremos es simplemente ver los valores de las variables y, llegado el caso, modificarlas o forzar sus valores, podemos hacerlo mediante la opción del árbol principal PLANTILLAS_STATUS, el menú **Comunicaciones / Status**, o bien mediante el botón correspondiente de la barra de herramientas. Aunque podemos acceder a este formulario sin estar en modo “Online”, no podremos usarlo si no estamos en ese modo. Además, pasar de modo “Online” a modo “Offline” provoca el cierre de este formulario. Cuando se muestra, aparecerá el formulario de la Ilustración 88.

DESIGNER 3.0

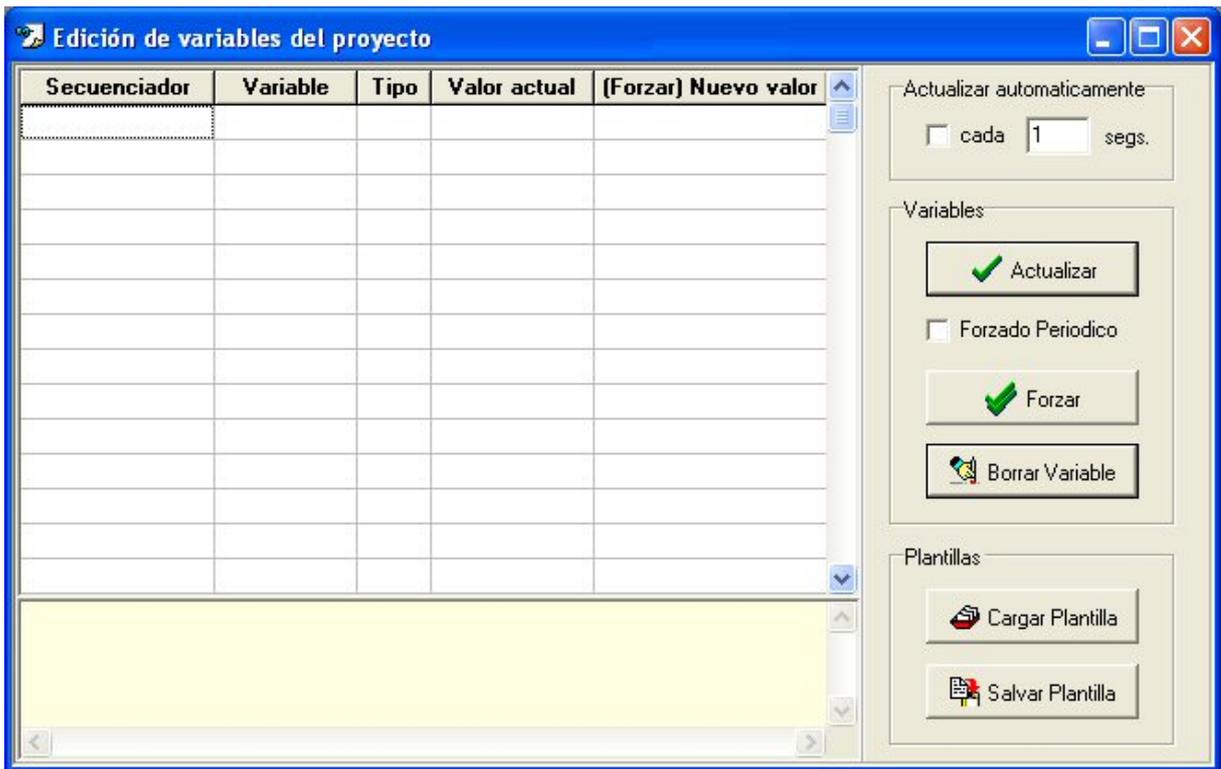


Ilustración 88: El formulario de visualización de variables.

En la tabla podemos seleccionar variables (especificando su componente “Machine” o “Zona” y su nombre de variable. En la columna “Tipo” podremos seleccionar el formato en el que queremos visualizar los valores, pudiendo elegir entre los siguientes modos:

- KM → El valor de la variable se mostrará en formato binario, por ejemplo: una variable de tipo byte con valor 4 aparecerá como “00000100”
- KH → El valor de la variable aparecerá en formato hexadecimal, por ejemplo, una variable de formato byte y valor 15 aparecerá como “F”
- KF → El valor de la variable se muestra en formato decimal. Este es el formato por defecto, se selecciona automáticamente tras seleccionar una variable.
- KC → El valor de la variable se mostrará en formato carácter. Por ejemplo, la variable de tipo byte cuyo valor sea 88 aparecerá como “X”. Si la variable tiene formato Word o Dword debe tenerse en cuenta que al alineación de sus bytes puede hacer que no se muestre ningún carácter, al ser el primero de la cadena el “0” por tener el byte de más peso ese valor (una cadena que comienza por 0 se considera vacía).
- KY → El valor de la variable se mostrará como el valor de sus bytes separados por espacios, por ejemplo, la variable de tipo word cuyo valor sea 12000 se mostrará como “046 014”

El formato de estas variables puede cambiarse en cualquier momento por cualquier otro de los posibles, sin afectar ello al valor de la variable que se está mostrando.

El valor de las variables seleccionadas se refresca por defecto a cada segundo. Podemos cambiar la frecuencia del refresco mediante el cuadro de texto de la parte superior, o desactivarlo completamente al desmarcar el checkbox situado en la parte superior derecha de la pantalla.

Para eliminar una variable de la ventana (para no ver su valor), basta con seleccionarla en la tabla y pulsar a continuación el botón “Borrar variable”, lo que hará que desaparezca esa línea de la tabla (no afectará a la variable del programa).

Por último, para facilitar la selección rápida de muchas variables, si pulsamos con el botón derecho del ratón encima de las 2 primeras columnas de la tabla, aparecerá un menú contextual similar al que se muestra en la Ilustración 89.

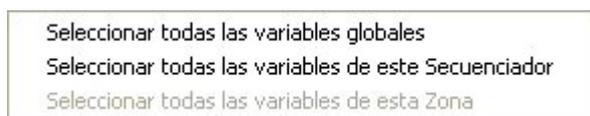


Ilustración 89: Menú contextual de la visualización de valores.

Si seleccionamos la primera opción, la tabla será rellenada de forma automática con todas las variables globales definidas en el simbólico, mostradas en formato “KF”.

Si el botón derecho del ratón lo pulsamos sobre una casilla en la que previamente hemos seleccionado el nombre de una “Machine” definida en el simbólico, la opción “Seleccionar todas las variables de este secuenciador” aparecerá activa en este menú, permitiéndonos, en ese caso, seleccionar de golpe todas las variables contenidas en esa “Machine”, mostradas en formato “KF”. Un funcionamiento análogo puede deducirse si lo que seleccionamos en la casilla es una zona.

Por último, si pulsamos el botón derecho del ratón cuando este se encuentra situado encima de la tercera columna de la tabla, esto provocará la aparición de otro menú contextual que nos permitirá elegir entre todos los tipos de formato de valores, y al seleccionar alguno de ellos, el cambio se aplicará a TODOS las variables que aparecen en la tabla en ese momento.

Las variables de tipo BOOL sólo se mostrarán como valores “0” o “1”, independientemente del formato seleccionado.

Forzar cambios en los valores de las variables

Además de ver los valores de las variables, a veces puede ser útil forzar un cambio en dichos valores, a fin de emular condiciones y ver como reacciona nuestro programa de control. Para ello, una vez seleccionada la variable, basta con que coloquemos un valor en la celda de la columna “Forzar (Nuevo Valor)” de la tabla. Este forzado sólo se mantendrá durante 1 único ciclo del programa de control, es decir, que si forzamos una variable de periferia, durante 1 ciclo para el programa de control esa variable tendrá el valor que le hayamos asignado, pero al siguiente ciclo de control, si la variable es de periferia, al refrescar los valores, la variable volverá a tener su valor real. Todas las variables cuyo campo “Forzar” contenga un valor, serán forzadas a la vez cuando pulsemos el botón.

✦ Forzar durante más ciclos, es posible que 300 ciclos sea un tiempo insuficiente para realizar nuestra depuración. Podemos cambiar el número de ciclos de forzado pulsando la siguiente secuencia: Pulsar el botón “Forzar” y a continuación pulsar la tecla F2. Aparecerá un pequeño control que nos permitirá establecer el número de ciclos de forzado.

Si deseamos forzar esa variable durante un número de ciclos mayor y de forma periódica, sólo tenemos que pulsar el checkbox “Forzado periódico”, lo cual provoca que se fuercen las variables periódicamente

por un periodo de 300 ciclos de programa.

Hay que hacer notar que debemos introducir el valor de forzado en el mismo formato en el que estamos mostrando el valor de la variable.

Plantillas

Pulsando el botón “Cargar plantilla” tenemos la opción de cargar una plantilla de variables, es decir, una vista o conjunto de variables que queremos ver junto con su formato de visualización. Al pulsar el botón, aparece el formulario que se observa en la Ilustración 90



Ilustración 90: Cargando una plantilla de variables.

Desde aquí podemos borrar alguna de las plantillas almacenadas en el proyecto (seleccionándola y pulsando el botón “Borrar”) o cargar una o varias de estas plantillas, para ver rápidamente los valores de un conjunto de variables. Si seleccionamos varias plantillas (mediante la tecla **Shift**), en la tabla aparecerán las variables contenidas en todas las plantillas seleccionadas (unión de plantillas).

Si tenemos una vista de variables podemos salvarla para visualizarla posteriormente, pulsando el botón “Salvar Plantilla”, lo que provocará la aparición del formulario de la Ilustración 91.

DESIGNER 3.0



Ilustración 91: Salvando una plantilla de variables.

Para salvar la plantilla, lo único que hay que hacer es darle un nombre descriptivo (o seleccionar uno de los ya existentes), y después pulsar el botón “Guardar”.

Grafcet en línea

Cuando estamos en modo “Online” y abrimos la edición de un secuenciador, vemos que una nueva solapa aparece en la parte inferior de la edición de los grafos.

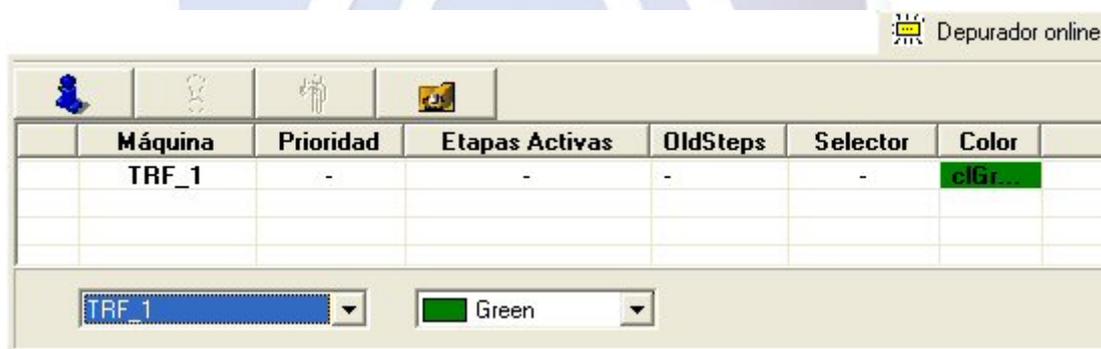


Ilustración 92: Pestaña de opciones "Online" de la edición de grafos

Desde esta ventana flotante, se puede seleccionar una instancia y un color con que se desea que aparezcan resaltadas las etapas activas de un secuenciador. Al pulsar el botón  se empezará a monitorizar esa máquina y a preguntar por sus estados, lo cual se reflejará al dibujar las etapas del mismo en pantalla (las activas aparecerán del color seleccionado).

La información que se nos muestra es la siguiente:

- **Prioridad:** Coincide con la prioridad actualmente activa. Si el valor es distinto de 0, significa que estamos ante un modo de funcionamiento por interrupción.
- **Etapas activas:** es la lista de etapas actualmente en ejecución. El número de etapa corresponde a la etapa dentro del modo de funcionamiento activo.

- OldSteps: Estas son las etapas de retorno (aquellas a las que la máquina volverá cuando se salga del modo de prioridad). Aparecerán marcadas con una flecha ← apuntándolas, para indicar que son el punto de retorno de la ANY.
- Selector: Si no se encuentra en un modo de prioridad, este número nos indicará el selector (o modo de funcionamiento) que se está usando actualmente en la ejecución. Esto también nos apuntará hacia el grafo concreto en ejecución.

Cuando se selecciona una instancia concreta de un secuenciador, el formulario principal de este plugin intentará centrar la primera etapa activa para que sea visible en el centro del área de dibujo, haciendo más rápida la localización del estado actual.

Si mientras tenemos seleccionado un elemento de esta lista pulsamos el botón , eliminaremos el elemento de la lista, y dejará de mostrarse información sobre su estado.

El tercero es el icono de “reset” . Tras seleccionar una instancia de la lista y pulsarlo, se nos dará la opción de resetear la máquina a su etapa inicial o a cualquier otra de sus etapas (a diferencia de formulario de edición de tracking, podemos resetearlo a cualquier etapa definida en cualquier modo de funcionamiento, aunque no tengan definido un alias).

El último botón  provoca, tras seleccionar una instancia y pulsarlo, la aparición del formulario de edición de tracking, que se discutirá más concretamente en el siguiente apartado.

El Editor de Tracking

Este formulario es el único que no puede seleccionarse de forma directa desde el árbol de proyectos, sino que se accede a él a través de otros plugins, concretamente desde una visualización (haciendo doble click sobre un objeto TlinkedElement) o desde la edición de Secuenciadores (cuando la tarjeta está “online” y se ha seleccionado una instancia desde la solapa “online” de los grafcet). Si se pulsa el botón apropiado, se pueden consultar los tracking que contiene la instancia que se ha seleccionado, tal y como se muestra en la Ilustración 93.

DESIGNER 3.0

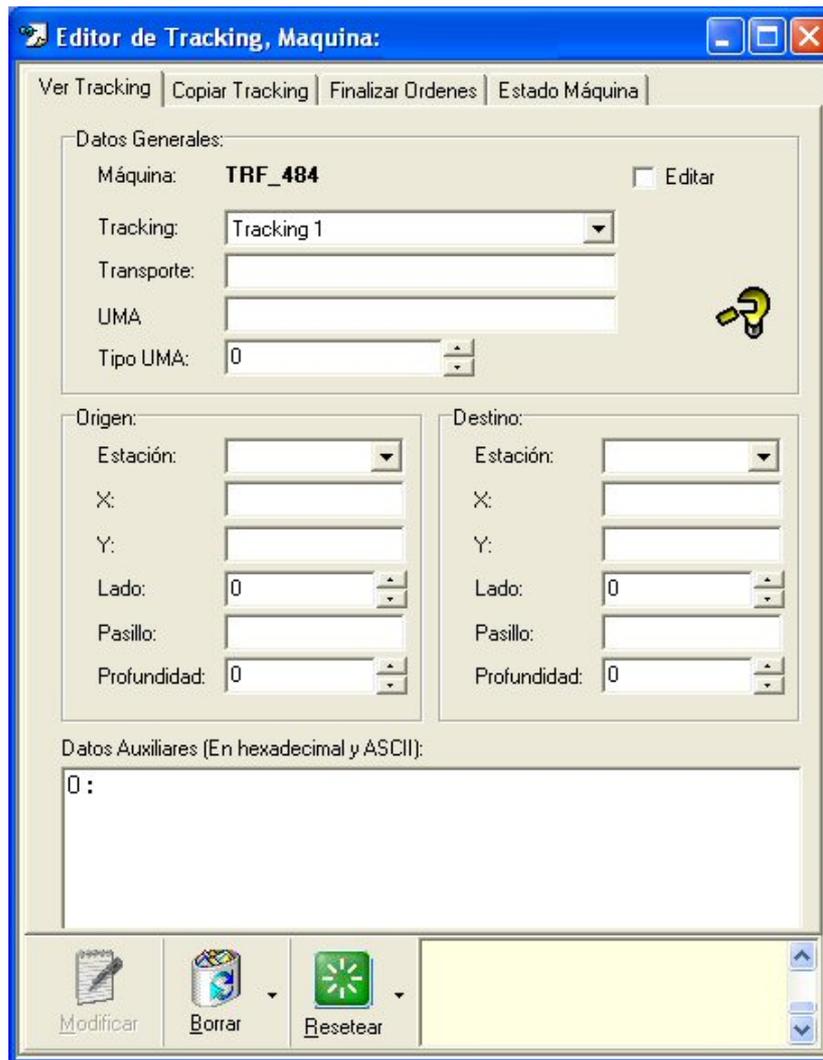


Ilustración 93: La solapa "Tracking" con información sobre una máquina.

Además, desde este formulario se puede limpiar el tracking seleccionado en cada momento (de los 10 posibles que se pueden ver de la máquina), es decir, colocar todos sus valores a 0. También es posible editar un valor concreto, pulsando la casilla “Editar”, modificando el valor deseado y pulsando el botón “Modificar” tras la modificación. Mientras se están editando los valores, no se refrescan los valores mostrados. Si tras una modificación no queremos establecer el valor, basta con deseleccionar la casilla “Editar” o salir del formulario.

El botón “Borrar” permite borrar los datos asociados al tracking que se está mostrando en el momento de su pulsación. Si se pulsa el botón “Borrar Todos”, se borrarán los 10 trackings asociados a la máquina.

El botón “Resetear” permite reinicializar la máquina mostrada, al igual que desde el formulario de edición de secuenciadores, con alguna diferencia:

- Si no se ha seleccionado nada en el combo “Etapa a la que resetear”, la máquina se inicializará a la etapa definida como “inicial”.

- Si se despliega el combo “Etapa a la que resetear”, se mostrarán todos los “Alias” definidos en las etapas del secuenciador, Si se selecciona uno de ellos, entonces la máquina se inicializará en la etapa a la que hace referencia dicho alias.

Las otras solapas del editor de tracking sirven para:

“**Copiar tracking**” (Ilustración 94): Mediante esta solapa copiaremos el tracking actualmente seleccionado en la primera solapa (“Ver tracking”) a otra máquina / tracking. También cabe la posibilidad de intercambiar los trackings si se marca previamente la casilla “Intercambiar”.



Ilustración 94:Copia / intercambio de un tracking

“**Finalizar Ordenes**” (Ilustración 95): Mediante esta solapa se permite que una orden actualmente en transporte pueda ser finalizada por el programador u operario de la instalación. Para ello basta con acceder a esta solapa e indicar los datos de la orden que se desea finalizar, es decir: Número de transporte, UMA que está transportando, código de error con el que se desea que se finalice la orden, datos auxiliares que pudieran ser necesarios para efectuar dicha finalización, y estación donde se quiere que dicha orden sea finalizada. La importancia de todos estos datos varía en función de la situación concreta en la que se pretende anular la orden y de la instalación en si. Debe ser algo que el programador debería definir como procedimiento a efectuar por el personal de mantenimiento de la instalación, basándose en el funcionamiento correcto de dicha instalación (es decir, intentando evitar que la finalización de una orden con parámetros incorrectos provoque una descoordinación de la parte de gestión informática). Este formulario en sí no realiza ningún tipo de filtro sobre ordenes que pueden ser finalizadas o no, siendo a priori posible finalizar cualquier tipo de orden en transporte, con las consecuencias que esto puede tener. Es por ello importante que el programador de la instalación documente cuando se puede hacer uso de esta opción en el manual de mantenimiento de la instalación, y los parámetros adecuados en cada caso. Un buen sistema de documentación puede incluir un ejemplo ficticio de situación en la que sea necesario usar esta opción para solucionar un problema en la instalación. Veamos una posible plantilla para la documentación:

Ante la rotura de un palet durante su transporte, y ante la imposibilidad de que dicho palet siga moviéndose por las mesas de cabecera, este palet debe ser retirado manualmente y a continuación es necesario usar la opción de finalizar ordenes del Editor de Tracking para:
Finalizar el transporte con un código de error XXX (como valor auxiliar se usara YYY).
Borrar el tracking de ese transporte de la mesa donde se produjo el fallo (o de su posterior o anterior, según la situación.)”

The screenshot shows a software window with four tabs: 'Ver Tracking', 'Copiar Tracking', 'Finalizar Ordenes', and 'Estado Máquina'. The 'Finalizar Ordenes' tab is active. It contains a form with the following fields:

- Transporte: [Text input field]
- UMA: [Text input field]
- Código de Error: [Text input field]
- Auxiliar: [Text input field]
- Estación: [Dropdown menu]

At the bottom of the form are two buttons: 'Copiar Act.' (with a document icon) and 'Finalizar' (with a checkmark icon).

Ilustración 95: Finalizando una orden desde el editor de tracking.

Los valores XXX e YYY deberán ser dependientes de lo que la gestión informática del almacén considere apropiado para estos casos.

El botón “Copiar Act.” Funciona como un atajo rápido para rellenar algunos de los campos del formulario, y lo que hace es copiar los valores (número de transporte y UMA) de la pestaña de “Ver Tracking” a los campos de este formulario, ahorrándose tener que teclearlos.

“**Estado de Máquina**” (Ilustración 96): Existen ciertos casos en los que nos interesa (bien para trazar su estado o para depurar su funcionamiento), conocer el estado de los flags internos de funcionamiento de una máquina, así como otros detalles internos de la misma que nos pueden informar sobre un malfuncionamiento en la misma o en el programa de control. Estos detalles podemos obtenerlos en esta solapa.

DESIGNER 3.0



Ilustración 96: Estado de los flags de la máquina.

En la parte superior podemos obtener el número de máquina. A continuación, dos campos nos muestran en forma de texto el estado actual de la máquina y una explicación sobre el último error que sufrió dicha máquina si se encuentra en estado de error.

En esta última solapa aparece otro nuevo elemento de información donde se pueden distinguir varias páginas:

Zona Recepción de orden (Ilustración 96): Esta pantalla solo contiene datos útiles si la máquina puede recibir ordenes (es decir, es un inicio de trayectoria). Veremos en todo momento los flags mostrados nos darán los valores referidos a la última orden recibida por la máquina:

- X Actual: Coordenada X actual de la UMA en transporte.
- Y Actual: Coordenada Y actual de la UMA en transporte.
- Lado: Lado actual en que se encuentra la UMA en transporte.
- Pasillo: Pasillo actual en que se encuentra la UMA en transporte.
- Estatus: Palabra de estado que define el estado en que se encuentra la UMA en transporte.
- Flag Petición Orden: Indica si la máquina esta libre (solicitando una orden), u ocupada (ya tiene una orden o no puede pedir ordenes en ese momento).

- Flag Orden Enviada: Indica si la máquina ha recibido una orden y se encuentra gestionándola.

Zona de fin de Orden (Ilustración 97): De forma análoga a la anterior, esta página solo contiene datos útiles cuando la máquina en cuestión puede finalizar ordenes (es un fin de trayectoria). En dicho caso, se nos muestran los flags correspondientes a la última orden finalizada en esa máquina:

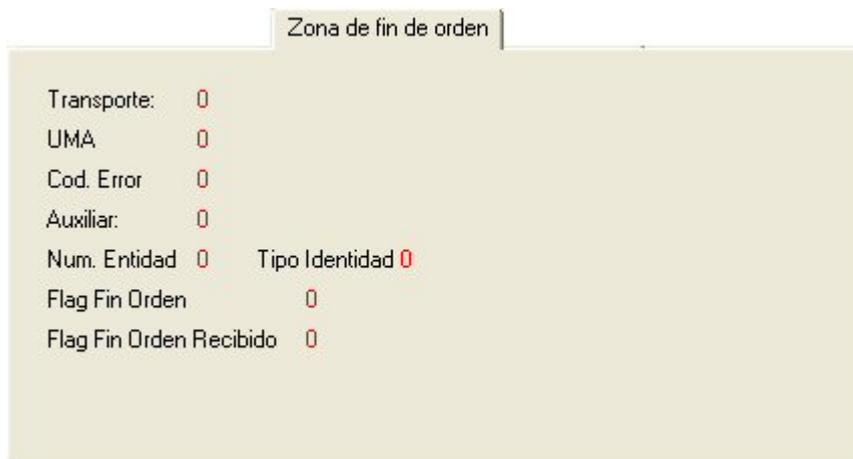


Ilustración 97: Flags de fin de orden.

- Transporte: Número de transporte que finaliza.
- UMA: Código de identificación de la unidad transportada.
- Cod. Error: Código de error con el que la máquina finaliza el transporte.
- Auxiliar: esta es una palabra auxiliar que se puede establecer con el fin de orden.
- Num. Entidad: Número de la entidad (estación lógica) donde finaliza la orden.
- Tipo Entidad: Tipo de entidad (estación lógica) donde finaliza la orden.
- Flag Fin Orden: Nos indicará si la máquina está intentando señalar un fin de orden.
- Flag Fin Orden Recibido: Este flag indicará que el sistema de gestión se ha apercibido del fin de orden que la máquina ha enviado.

Zona de PIE (Ilustración 98): Análogamente a los anteriores, esta página solo tiene sentido si la máquina fue definida como una estación de PIE. En otro caso, estos datos no tienen mayor relevancia, pero si es el caso, aquí se nos mostrarán los datos del último PIE procesado por la máquina:

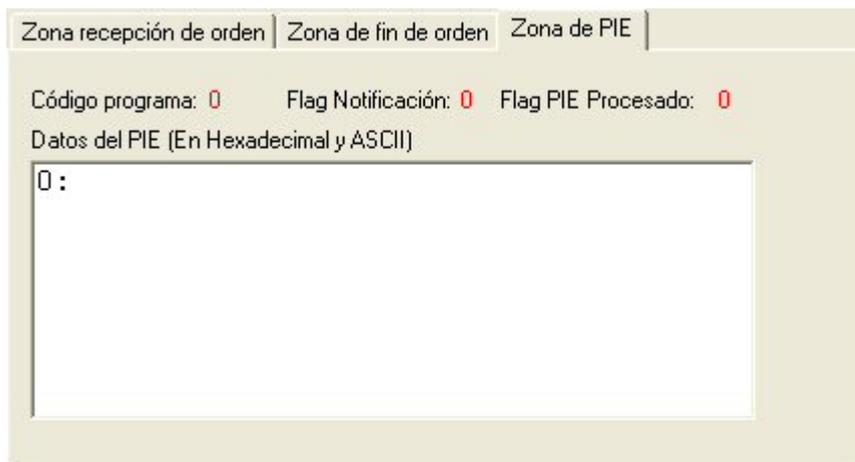


Ilustración 98: Flags de zona de PIE.

- Código programa: Esta palabra es usada a manera de convención para señalar el tipo de PIE que se quiere comunicar entre el sistema de gestión del almacén y el sistema de control del mismo. El valor de este flag es convenio entre las dos partes, y debe documentarse en los manuales de mantenimiento de la instalación.
- Flag Notificación: indica que la máquina esta señalando una entrada al sistema de Gestión del almacén
- Flag de PIE procesado: indica si la entrada marcada por la máquina ya ha sido reconocida por el sistema de gestión del almacén.
- Datos PIE: muestra en hexadecimal y ASCII los valores de los 50 bytes que componen la información del PIE (los primeros 25 pueden ser usados por el programador, los últimos 25 pueden ser usados por el SGA).

Al igual que la solapa inicial del editor de tracking, los datos de esta solapa se refrescan automáticamente cuando se están mostrando, pero sin embargo, no podemos modificarlos manualmente.

Las solapas ocultas

Existen un par de solapas más en el editor de tracking que en principio no son mostradas y permanecen ocultas hasta que se accede a ellas de la siguiente forma:

Desde la primera solapa, pulsamos doble click sobre el icono en forma de herramienta. Aparecerá en ese momento el diálogo de permisos de usuario, que nos pedirá un perfil “Developer” para poder acceder a estas opciones. Si introducimos un usuario correcto y su clave, veremos que dos nuevas solapas se añaden al editor de tracking, y un nuevo icono aparece en la actual. Si pulsamos otra vez con doble click sobre el mismo icono, el editor de tracking volverá a su estado anterior y las solapas se ocultarán de nuevo.



Veamos ahora la funcionalidad de todas esas etapas e iconos nuevos.



Al hacer doble click sobre este icono el tracking actual se pondrá en modo edición y sus campos se rellenarán con datos más o menos aleatorios. Será necesario rellenar manualmente los campos de estación de origen y destino, por ejemplo. Su utilidad es meramente la de generar datos de tracking distintos de cada vez para ayudar al desarrollador a mover unidades de transporte está en fase de pruebas, no debería usarse nunca una vez que el sistema está ya en producción.

En cuanto a las solapas, tenemos en primer lugar la solapa de Grafo en Ejecución, que podemos observar en la Ilustración 99

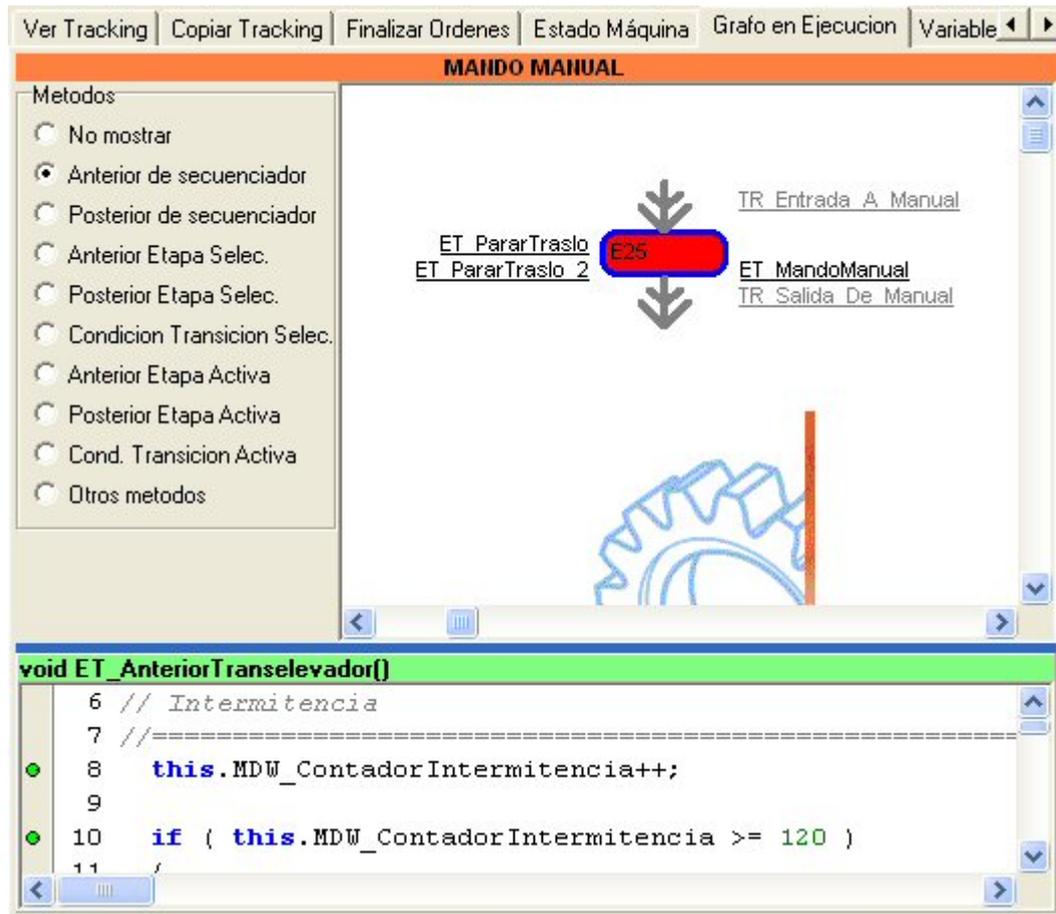


Ilustración 99: Grafo en ejecución.

En esta solapa se puede ver el modo de funcionamiento actualmente activado y centrado en las etapas activas en ese momento. En el panel de la izquierda se puede elegir el código en ejecución a mostrar.

Aparte de una serie de funciones predefinidas, también podemos elegir la función que queramos para mostrar las líneas en ejecución (aparecerán mostradas en la parte inferior). Las líneas que se están ejecutando se distinguen porque tienen un símbolo en forma de círculo verde en el margen izquierdo. Por supuesto, la exactitud de esta información depende de que el proyecto que tenemos cargado sea el mismo que está funcionando. Si por ejemplo, hemos editado esa función con posterioridad a la compilación, las líneas pueden estar desplazadas, notándose la adición o eliminación de otras líneas.

Si el estado de ejecución cambia, el grafo cambiará para mostrar el nuevo estado. En la etiqueta de la parte superior siempre aparecerá el nombre del modo activo.

El grafo que se muestra en esta solapa no es modificable, por lo que no podremos usar el puntero de ratón para mover sus etapas o ninguna otra operación que solo es posible desde la edición de Secuenciadores.

La última solapa sirve para conocer los valores de las variables asociadas a la máquina actual y sus posteriores y anteriores. Su aspecto puede observarse en la Ilustración 100

Variable	Valor
Traslo9.F_ConfigTraslo	c:\galileo\configTr9.ini
Traslo9.M_EnPosicionManualZ	0
Traslo9.M_TrasloCentradoX	0
Traslo9.M_CentrajeAlto	0
Traslo9.M_MovimientoManualProhibido	0
Traslo9.MW_ActivarSemiautoCuna	0
Traslo9.M_ErrorGalibo	0
Traslo9.M_EnPosicionY	0
Traslo9.M_EnPosicionX	0
Traslo9.M_CompruebaSobrelevacion	0
Traslo9.MDW_EstadoTraslo	77
Traslo9.M_ControlTimeout	0
Traslo9.MDW_SalvarPresetZ	0
Traslo9.M_FlancoCentraje	0
Traslo9.M_Centraje	0
Traslo9.MDW_Mov_CotaTrasloManualY	0
Traslo9.M_B_TranManual	0
Traslo9.MDW_Tipo_Entidad_Destino	0

Ilustración 100: Variables de máquina en ejecución

Como vemos, se muestran los valores de todas las variables, incluso las cadenas de texto. La única excepción son los componentes, que no tienen valor real. Si pulsamos los botones de la parte superior, podremos ir viendo los valores que toman las máquinas anterior1 (-1), anterior2 (-2)... y las posterior1 (1), posterior2 (2), etc. Si alguna de estas máquinas no está definida, la tabla se mostrará vacía. El botón con forma de rayo nos realizará un filtro de las variables de la máquina, de forma que sólo las de tamaño bit (booleanas) sean mostradas en la lista.

Los valores que se muestran en esta tabla no son modificables, se refrescan periódicamente, pero si queremos forzar algún valor tenemos que acudir al formulario de depuración de variables. Sin embargo, como un caso puntual, si se permite hacer “toggle” (cambio) del valor de las variables booleanas, mediante un menú del botón derecho del ratón. Al usar esta opción, se intentará un forzado

de la variable booleana seleccionada al valor opuesto al que tiene actualmente (es decir, pasarla a falso si vale cierto, y viceversa). Si la variable no es de tipo booleano, o es una constante, esta acción no tendrá ningún efecto.

Asimismo, para poder realizar ese forzado, se debe contar con los permisos adecuados como usuario del proyecto.



DESIGNER 3.0

Repositorio de Secuenciadores

Antes de ponerse manos a la obra, hay que consultar si lo que necesitamos no está hecho ya...

Hemos comentado ya que la potencia de uso de los secuenciadores está en su reutilización. El uso de secuenciadores previamente probados y que funcionan de forma genérica puede ahorrar horas de programación y depuración, a la par que disminuir el tiempo de aprendizaje para el personal de mantenimiento, al reaprovechar también manuales, documentación de casos de trabajo, etc..

El Repositorio Remoto

Por este motivo, existe un repositorio de Secuenciadores desde el que se pueden cargar una serie de secuenciadores estándar listos para su uso. Este repositorio está organizado en forma de versiones y revisiones, que permiten actualizar un proyecto de forma automática a nuevas revisiones teniendo la seguridad que el cambio será totalmente compatible. Dicho repositorio puede ser cargado, bien para editarlo o para actualizar un proyecto dado, desde la dirección www.tkmanutencion.com.

Para acceder a él, desde el entorno de programación basta con usar la opción “Fichero → Acceso al Repositorio”, que nos llevará a la aparición de un formulario como el de la Ilustración 101.

Para acceder al repositorio necesitamos un nombre de usuario y password con permisos en el repositorio. Estos usuarios son creados desde el departamento de soporte, luego deberá de hablar con uno de ellos si requiere de este tipo de acceso. Los usuarios pueden tener permiso de lectura-escritura (lo que les permite editar los elementos del repositorio) o solo de lectura (lo que les permitiría solamente actualizar un proyecto para que incluyera los elementos del repositorio, pero no modificar estos en modo alguno). Estos permisos se muestran en el campo “Permisos” una vez que se ha accedido al repositorio tras pulsar el botón “Actualizar vista”.

DESIGNER 3.0

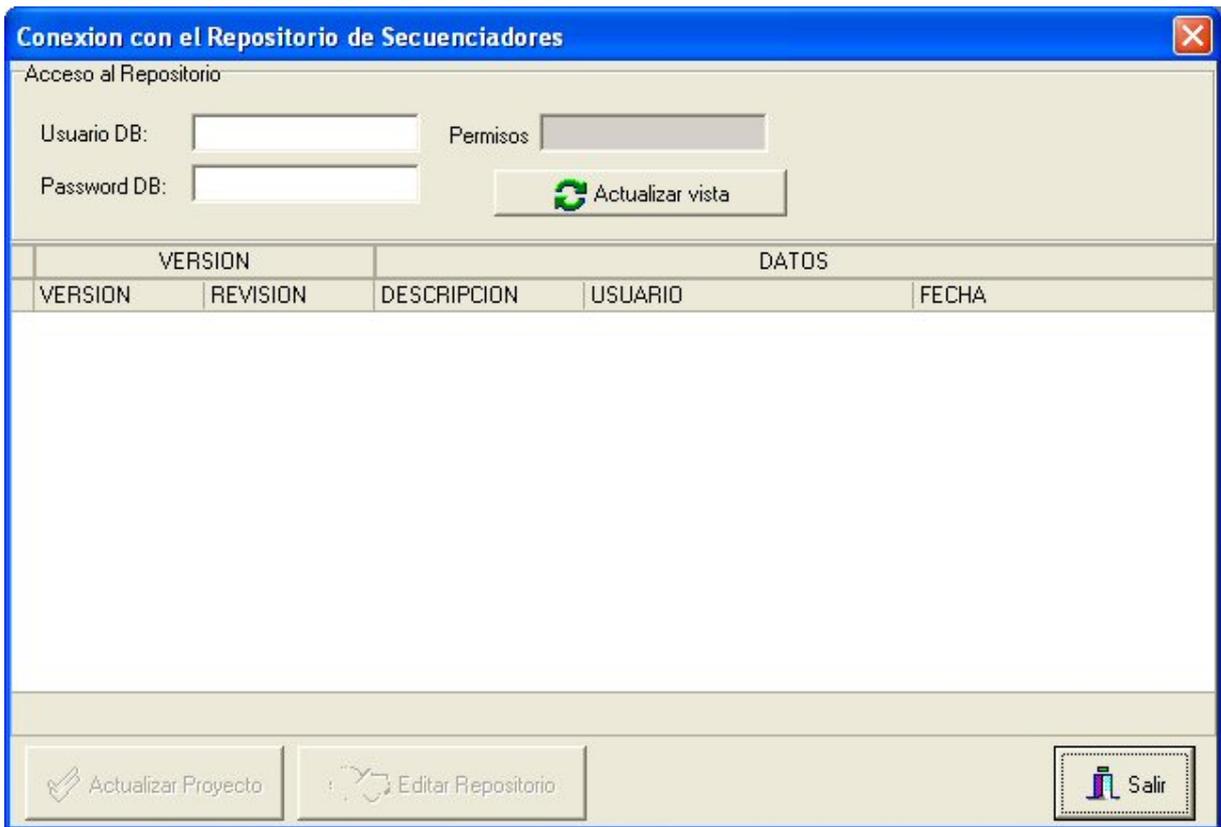


Ilustración 101: Formulario de acceso al repositorio de secuenciadores

Una vez que tengamos acceso al repositorio, veremos una lista de todas las versiones/revisiones disponibles en él. Si ya estamos editando un proyecto, podemos actualizarlo (siempre que la version / revision que usemos en el proyecto sea menor que la hemos seleccionado). Si elegimos actualizar a un numero de version superior al actual del proyecto, recibiremos un aviso de que los cambios de version implican cambios de funcionalidad que puede provocar que nuestro proyecto tenga que ser reconfigurado para poder funcionar. En cambio, si sólo actualizamos a una revisión superior del repositorio, no recibiremos más avisos y la actualización será llevada a cabo sin más dilación. En ambos casos, al acabar recibiremos otro mensaje indicandonos los cambios llevados a cabo en nuestro proyecto.

Si tenemos permisos de escritura en el repositorio, podemos seleccionar una version/revision para su edición. Al hacerlo, el repositorio se cargará como si fuera un proyecto normal, con un nombre compuesto de los valores de versión y revision seleccionada. A partir de ahí podemos trabajar con sus secuenciadores de la misma manera que lo haríamos con un proyecto normal. La diferencia estriba en que cuando queramos salvar el proyecto, en lugar de salvarlo a disco, nos aparecera el formulario que presentamos en la Ilustración 102

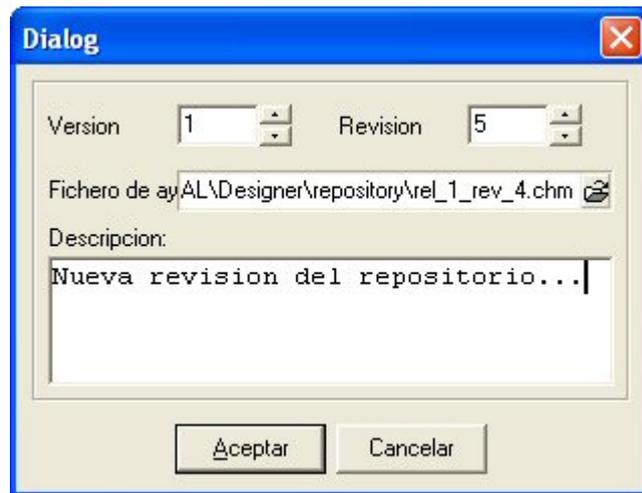
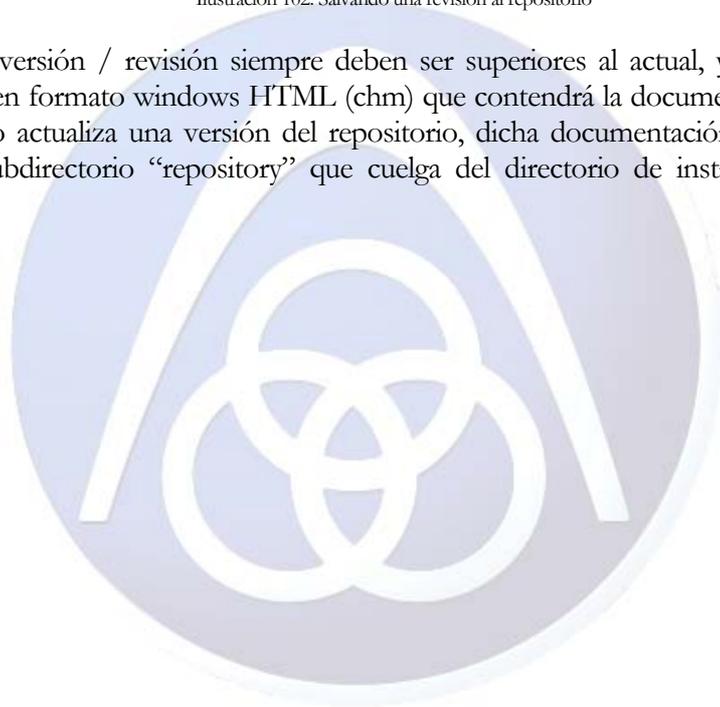


Ilustración 102: Salvando una revisión al repositorio

Los números de versión / revisión siempre deben ser superiores al actual, y se debe de incluir un fichero de ayuda en formato windows HTML (chm) que contendrá la documentación del repositorio. Cuando se abre o actualiza una versión del repositorio, dicha documentación (si existe) también se descarga en el subdirectorío “repository” que cuelga del directorio de instalación del entorno de desarrollo.



DESIGNER 3.0

Herramientas

Las herramientas incluidas con el entorno de desarrollo son aplicaciones independientes que permiten realizar una serie de tareas no tan frecuentes, pero necesarias en algún momento.

Aunque no se han producido muchos cambios desde la versión 2.x, algunas herramientas se han incorporado a esta nueva versión, en este capítulo nos dedicaremos a repasar las viejas herramientas y explicar las nuevas incorporaciones.

Generación de Informes de proyecto

Los informes son una herramienta muy aconsejable para poder estudiar / documentar un proyecto o simplemente para explicar sus características a terceros. El plugin de generación de informes cumple una doble funcionalidad, por una parte permite generar una documentación técnica exhaustiva y completa de los detalles del proyecto y por otra, también es posible generar informes de partes concretas de los proyectos, que pueden ser muy útiles a la hora de depurar su funcionamiento. Para lanzar el plugin de generación de informes, sólo es necesario tener un proyecto activo, y usar la opción **Fichero / Generar Informe** del menú principal. Ello causa la aparición del formulario de la Ilustración 103, que está dividido en dos solapas que comentaremos por separado.

DESIGNER 3.0



Ilustración 103: El plugin de generación de informes.

Solapa "Generación"

Para la generación de un Informe hay que definir donde se quiere generar y la hoja de estilo (plantilla) sobre la que se quiere crear. Estos datos serán recordados para cada usuario de la aplicación y aparecerán por defecto la próxima vez que se cargue el plugin. Si no se determinasen estos datos, se generaría en el directorio actual de la aplicación y sin ninguna hoja de estilo. Existen unas hojas de estilo básicas, localizadas en la carpeta html dentro de la propia carpeta donde se instala el entorno de desarrollo y cuyos nombres son *template.css* y *profesional.css*. El propio usuario puede personalizar la generación del informe elaborando sus propias hojas de estilo. Para más información sobre las hojas de estilo (Cascating Style Sheets), se recomienda cualquier manual o tutorial de los que se encuentran en Internet sobre el tema.

Una vez dada la orden de "Generar", el plugin creará un fichero de extensión html en el path indicado previamente y cuyo nombre será el del proyecto. En función de las opciones que señalemos a la hora de generar el informe, este contendrá más o menos datos. Las opciones posibles, tal y como se ve en la Ilustración 103 son las siguientes:

- **Imprimir Gráficos:** Seleccionando esta opción, en el fichero Nombre_Proyecto.html aparecerá el esquema de todos los secuenciadores definidos, así como un fichero de extensión .gif para cada grafo de cada Secuenciador y cuyo nombre estará compuesto por el nombre del

proyecto, seguido del nombre del servidor, el nombre del secuenciador que contiene y el nombre del modo de funcionamiento dentro del secuenciador.

- **Imprimir Variables:** seleccionando esta opción, en el fichero Nombre_Proyecto.html aparecerá un listado de todas las variables, organizadas en tablas.
- **Imprimir Código:** seleccionando esta opción, en el fichero Nombre_Proyecto.html aparecerá un listado de todas las funciones definidas en el proyecto, con su correspondiente código.

Solapa “Visualización”

Una vez que se pulsa el botón “Generar Informe” en la pestaña “Generación” del plugin, la página html que se ha generado se cargará en la pestaña “Visualización” del plugin, siendo posible verla como si se estuviera cargando desde el Navegador Internet Explorer de Windows. Si se han generado los elementos correspondientes, será posible navegar (mediante los hipervínculos) entre las diversas informaciones que contenga la página.

Edición de mapas

La edición de máquinas apareció por primera vez en las últimas versiones 2.x. Consiste en un formulario que permite crear y editar mapas de varias dimensiones, de forma mucho más controlada que editando el fichero a mano. Para acceder a este editor, solo hay que usar la opción **Herramientas / Editar Mapa 2D/3D** del menú principal, momento en el que aparecerá el formulario de la Ilustración 104:



Ilustración 104: Plugin de edición de mapas.

En la imagen el plugin se encuentra editando un ConfigMap (mapa de 1 dimensión). Se pueden introducir los campos “Valor” y “Comentario”, y al salvar el fichero este se generará en el formato apropiado para que sea entendible por un componente “ConfigMap”.

El método de trabajo con este formulario siempre es el mismo:

Si lo que deseamos es crear un nuevo mapa, pulsamos alguno de los botones “Nuevo” de los que disponemos, ya sea de 1, 2 o 3 dimensiones el mapa que queramos crear. Si queremos partir de un mapa ya existente, seleccionaremos su fichero en el combo “fichero” y a continuación pulsaremos el botón “Cargar”, para mostrar sus valores en la tabla.

A continuación insertaremos, borraremos o modificaremos los valores que contendrá el mapa editando directamente los valores en la tabla. También disponemos de una opción de “Relleno automático”, que en función de las dimensiones del mapa, nos mostrará alguno de los diálogos siguientes:



Ilustración 105: Algoritmo de relleno de ConfigMap



Ilustración 106: Algoritmo de relleno de mapa de 2 dimensiones

DESIGNER 3.0

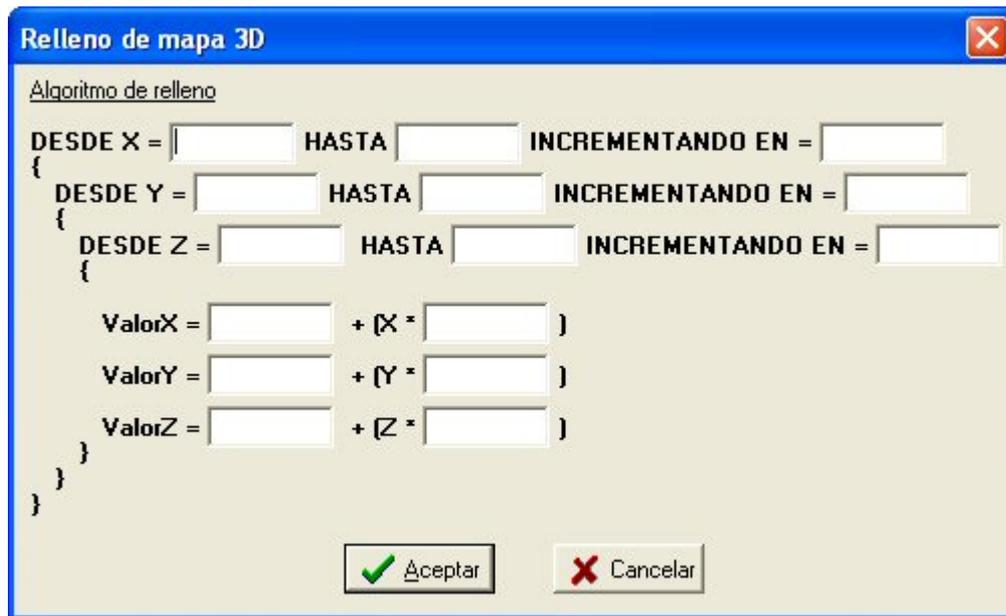


Ilustración 107: Algoritmo de relleno de mapa de 3 dimensiones

Como puede apreciarse, los algoritmos de relleno son bastante simples. (Nótese que un relleno automático borrará los valores existentes actualmente en la tabla).

En el caso de que estemos modificando mapas de más de una dimensión, tenemos siempre la opción de modificar sus “offsets” pulsando el botón correspondiente, lo que provoca la aparición de uno de los diálogos siguientes:



Ilustración 108: Offset de mapa 2D



Ilustración 109: Offset de mapa 3D

También, sólo en el caso de mapas de más de una dimensión, tenemos la posibilidad de acceder a una representación del mapa en una cuadrícula gráfica, en la solapa asociada que aparece al editar el mapa. Por ejemplo, en el caso de los mapas 2D, se nos mostrará algo como la Ilustración 110

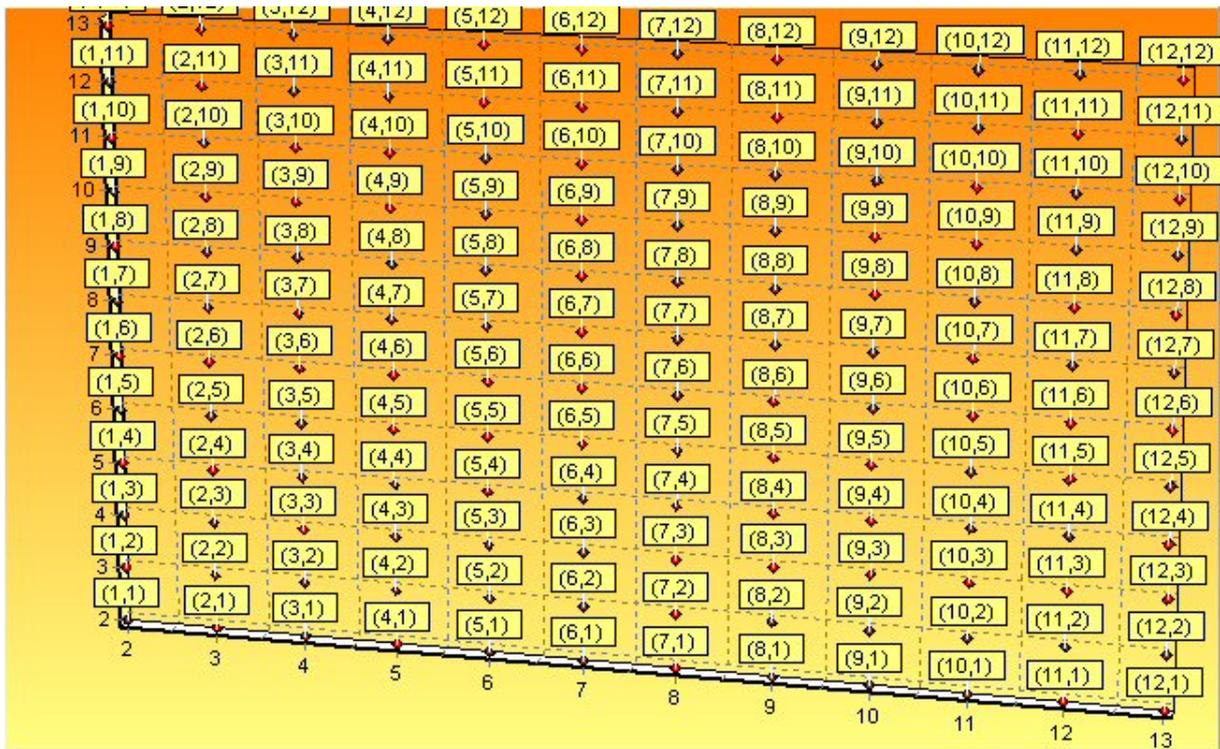


Ilustración 110: representación grafica de un mapa 2D

Como vemos, los valores del mapa aparecen representados en los ejes X, Y, y las coordenadas concretas se muestran en la etiqueta que aparece sobre cada punto. Podemos realizar un acercamiento a un área concreta de este gráfico pulsando con el puntero del ratón sobre cualquier punto del gráfico y arrastrando el puntero hasta una posición hacia abajo y hacia la derecha de la actual. Esto definirá una región sobre la que se realizará el zoom a fin de agrandar ese área. Si hacemos el desplazamiento del puntero en cualquier otra dirección o sentido, volveremos al mapa original.

Si en cambio lo que tenemos es un mapa en 3D, el gráfico que lo representará será un poco más complicado, veámoslo en la Ilustración 111:

DESIGNER 3.0

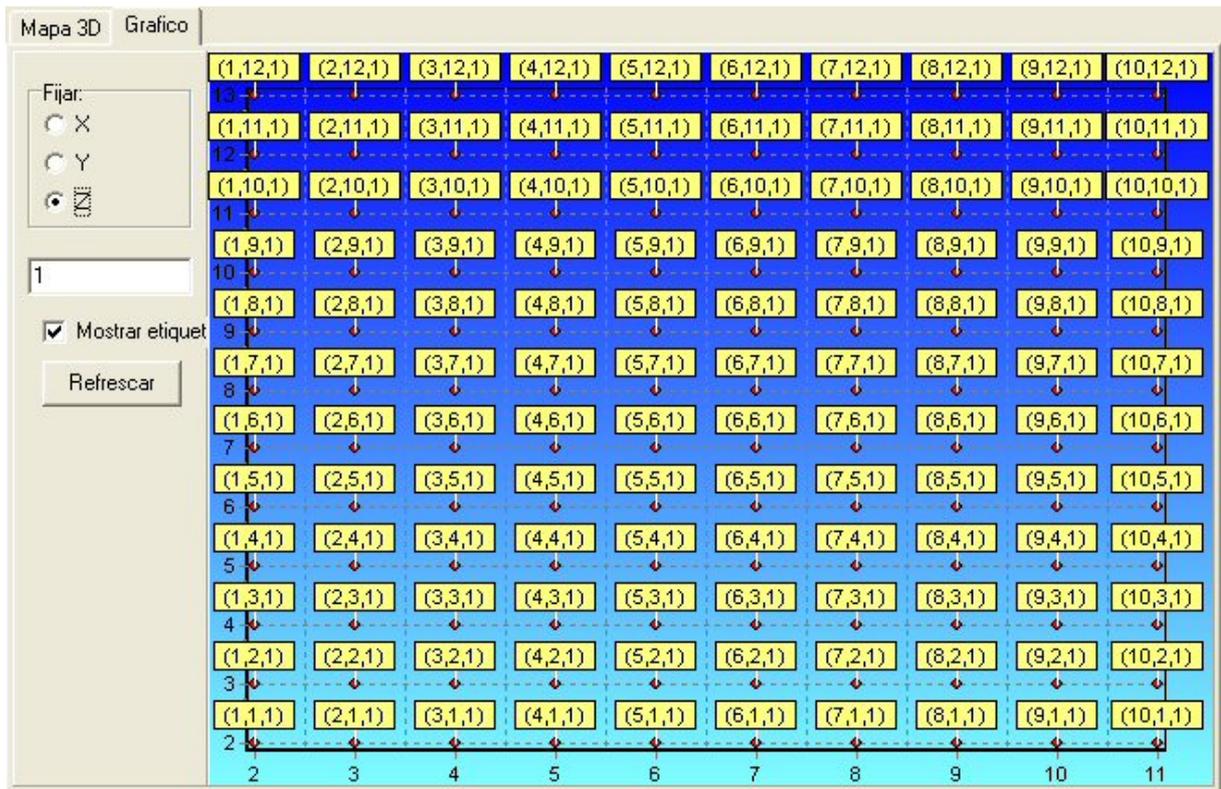


Ilustración 111: representación de mapa 3D

Como representar las 3 dimensiones es excesivamente complejo, se opta por fijar una de las coordenadas a un valor concreto y la representación del mapa consistirá en los valores de las otras coordenadas actuando como eje X, Y. Sobre los puntos concretos aparece de nuevo la etiqueta con el valor de las coordenadas (X,Y,Z) que lo definen. En este caso podemos elegir si queremos mostrar esta etiqueta o no pulsando el botón “Mostrar Etiqueta”. Si cambiamos el eje a fijar o el valor de la coordenada fija, deberemos pulsar el botón “Refrescar” para actualizar la vista del plano.

Como en el caso del grafico 2D, podemos hacer zoom sobre una zona concreta del plano, de la misma forma antes descrita.

Esta manera de fijar los planos nos permite tener una visión en rodajas del gráfico tridimensional, seccionándolo en cada uno de sus planos, lo que nos puede permitir ver a simple vista valores anómalos si nuestro plano sigue una distribución ordenada.

Cuando hallamos realizado los cambios deseados, pulsaremos sobre el botón “Aplicar” y tras ello, para salvar a fichero estos cambios, usaremos el botón “Salvar”. Para hacerlo es necesario que el campo “Fichero” contenga un nombre de fichero válido donde salvar los contenidos.

Los ficheros de datos no tienen una extensión específica, pero generalmente se les suele aplicar la extensión .ini

Interfaz de objeto nativo

Cuando se quiera escribir la invocación de un método nativo en el lenguaje XANA, es muy útil tener una referencia a las funciones contenidas en la librería de componentes nativos XANA instalada en

nuestro sistema. Dichas referencias están descritas en la documentación sobre componentes, pero como puede no ser accesible, la aplicación proporciona este plugin que muestra toda la información que exporta un interfaz contenido en la librería de tipos que tengamos instalada.

Para acceder al formulario que muestra esta información, basta usar el menú **Herramientas / Mostrar Interfaz**, que nos mostrará el formulario que se aprecia en la Ilustración 112.

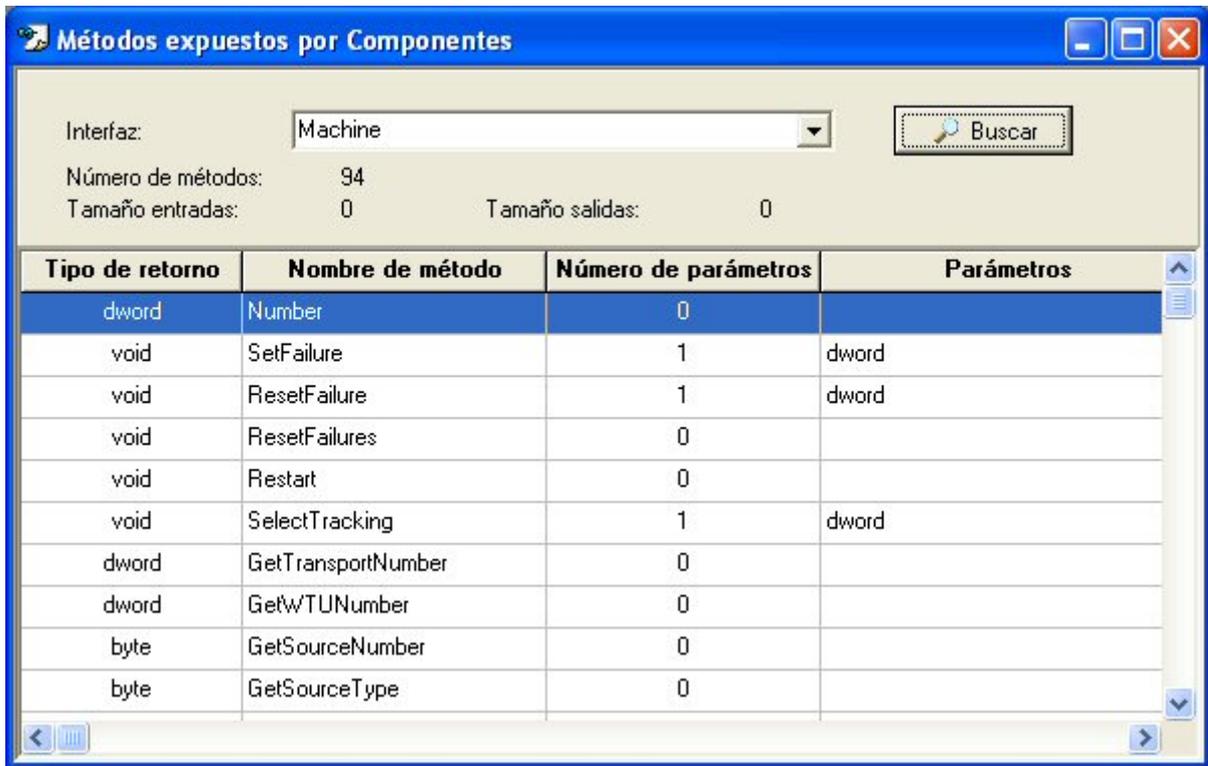


Ilustración 112: El plugin mostrando detalles del objeto "Machine"

En él se puede apreciar el tipo de retorno, nombre y los parámetros que requiere la utilización de cada uno de los métodos nativos que contiene el interfaz seleccionado. También podemos ver el número de métodos que exporta el componente y el tamaño, en bytes, que requiere de entradas y salidas de periferia para poder ser usado.

Si el nombre que solicitamos no era válido o no estaba instalado en nuestra computadora, el formulario no mostrará ninguna información, sólo un mensaje de error en la parte superior.

Podemos cambiar el nombre que aparece en el campo "Interfaz" y, pulsando el botón de la derecha, buscar los métodos de otro interfaz. El combo contiene una lista de los nombres de componentes más comunes. Se puede encontrar un listado de todos los componentes disponibles en la documentación del Sistema Galileo.

Para que este formulario funcione de forma correcta, es necesario que el sistema Galileo esté instalado en la misma computadora donde esta corriendo el entorno de desarrollo (aunque no este activo o se encuentre desactivado por no tener licencia).

Conversión de proyectos 2.x a 3.0

Se realiza mediante la herramienta conv223.exe, incluida en la instalación del Entorno de Desarrollo. Si al instalar el entorno elegimos crear un icono en el escritorio con acceso a esta aplicación, bastará con que arrastremos el fichero .THY con el proyecto que deseemos convertir a la versión 3.x.

Si la conversión se realiza con éxito, se creará un fichero con el mismo nombre que el original, pero con la extensión .TKR y en la misma carpeta que el fichero original. Durante el proceso de conversión pueden producirse diversos avisos a los que puede ser necesario atender. Al acabar la aplicación, es necesario pulsar cualquier tecla para que esta termine (esto nos permite leer los avisos).

Si no disponemos del acceso directo, podemos usar directamente la línea de comandos. La sintaxis de invocación es la siguiente:

```
conv223 <fichero.thy>
```

Registrar el tipo de archivo .TKR

Los ficheros donde se guarda la definición de los proyectos tienen por defecto la extensión .TKR. Si el usuario lo desea, es posible registrar este tipo de fichero para poder abrirlos directamente desde el administrador de archivos con un simple doble clic sobre el fichero deseado. Para ello, basta con seleccionar la opción **Herramientas / Registrar tipo de fichero** desde el menú principal.

El Galileo Status Monitor

Una vez que se completa la instalación de la aplicación, se suele dejar al cliente la parte de visualización, a fin de este pueda tener algún tipo de control sobre la instalación. Para ello fue diseñado el sistema "Status Monitor", que no es más que una versión reducida de la aplicación "Designer". Los pasos para dejar el Status Monitor funcionando son:

1. Obtener una licencia del producto "Status Monitor", que se debe instalar en el directorio donde esté instalado el "Designer".
2. Usar la opción **Herramientas / Instalar Status Monitor** del menú principal del Designer. Si esta opción fallase por algún motivo, siempre sería posible renombrar el ejecutable "Designer.exe" como "status.exe" directamente desde el administrador de archivos.
3. Crear un acceso directo para ejecutar este programa, y añadirle como parámetro la localización del fichero de proyecto con la última versión del programa de control corriendo. Este paso es realizado automáticamente por la opción de menú comentada en el punto 2, si cuando la usamos tenemos un proyecto activo en edición.

A partir de ese momento, cuando se ejecute ese acceso directo, la aplicación "Status Monitor" mostrará el formulario definido por defecto para la visualización (el primero creado), se pondrá en modo "online" e intentará pedir información sobre el progreso de los elementos que aparecen en la visualización.

Si el proyecto contiene más de una posible visualización, podremos elegir las que queramos ver simultáneamente mediante el menú principal de la aplicación.

Es posible que en algún momento no interese cambiar entre el status y el designer de forma rápida, generalmente durante las pruebas iniciales de una instalación. Es posible realizar este cambio mediante la invocación de la siguiente función registrada en el sistema de script:

```
ChangeModeDesigner( TForm formulario );
```

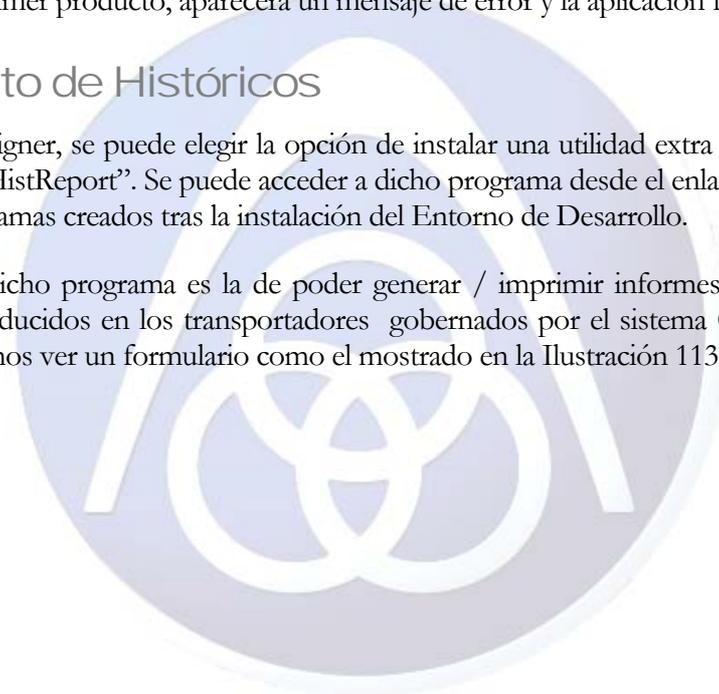
El parámetro requerido es un formulario, que bien puede ser “Application.MainForm”. Cuando invoquemos esta función desde el Status (Si lo hacemos desde el Designer no tendrá ninguna efecto), podremos ver que el entorno cambia y se transforma en una versión limitada del entorno Designer. Las limitaciones que tiene este modo se refieren principalmente que no tendremos acceso a ningún menú que permita salvar los posibles cambios que realicemos desde este modo. Cuando queramos volver otra vez al modo normal del Status Monitor, bastará con que usemos la opción “**Herramientas/Cambiar a Status Monitor**” que ahora aparecerá activa en el menú principal.

NOTA: Para realizar el cambio entre Status / Designer, es necesario tener las licencias de ambos productos. Si intentamos pasar a modo Designer desde el Status Monitor, y no tenemos instalada una licencia para el primer producto, aparecerá un mensaje de error y la aplicación finalizará su ejecución.

Tratamiento de Históricos

Al instalar el Designer, se puede elegir la opción de instalar una utilidad extra en el mismo grupo, que es el programa “HistReport”. Se puede acceder a dicho programa desde el enlace directo que se crea en el grupo de programas creados tras la instalación del Entorno de Desarrollo.

La finalidad de dicho programa es la de poder generar / imprimir informes sobre los históricos de movimientos producidos en los transportadores gobernados por el sistema Galileo. Tras arrancar la aplicación, podemos ver un formulario como el mostrado en la Ilustración 113:



DESIGNER 3.0

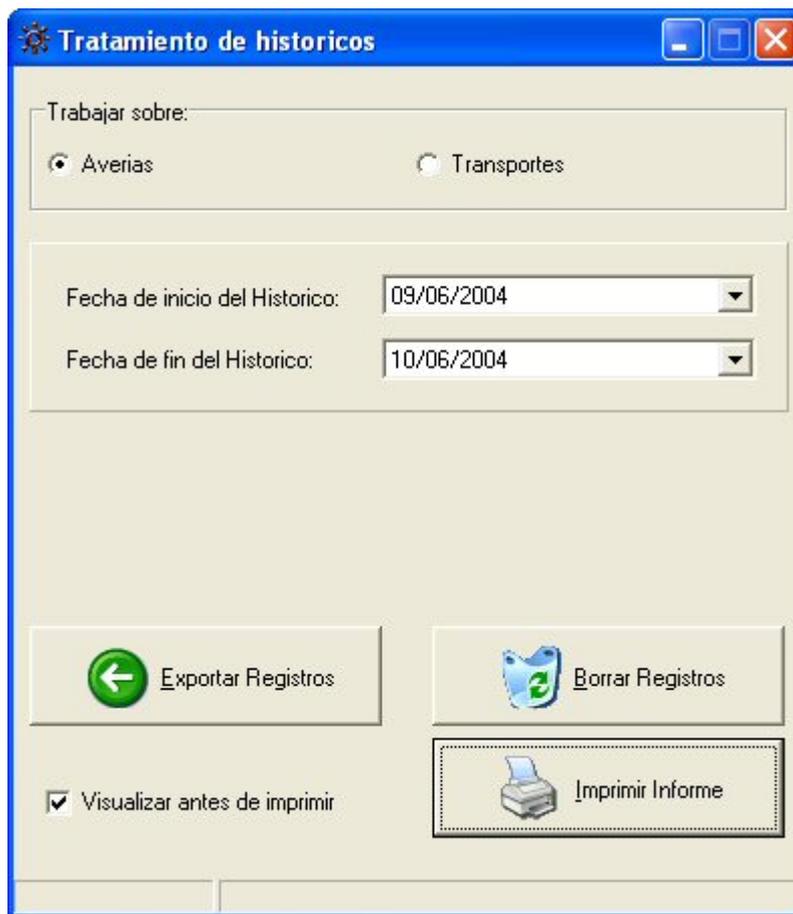


Ilustración 113: Informe de averías y transportes

Se pueden generar informes sobre los transportes ocurridos entre dos fechas, pudiéndose filtrar la consulta por número de palet o por número de transporte. Estas cadenas se tratan como literales, y se busca todo lo que coincida con ellos como comodín, es decir, si nosotros filtramos por número de palet y usamos la cadena “5” como filtro, todos los transportes cuyo número de palet contenga el número 5 aparecerán en el listado de las fechas seleccionadas. Si la cadena que usamos es “56”, aparecerán todos los transportes que contengan un número con las cifras “56” consecutivas, por ejemplo “125678”.

También se pueden generar informes sobre las averías producidas entre dos fechas, aunque en este caso, no se pueden aplicar más filtros.

En ambos casos, se pueden:

- a) Visualizar e imprimir el informe que resulta de la consulta que realizamos.
- b) Borrar de la BD los elementos que concuerden con nuestra consulta (para mantenimiento, o limpieza de tablas).
- c) Exportar los resultados de la consulta a una BD Access, cuya ubicación especificaremos, a fin de, por ejemplo, poder llevárnosla a otro sitio para un posterior análisis.

Como añadido extra, y a fin de facilitar un posible uso de esta herramienta por parte del cliente, esta aplicación también puede ser invocada desde el Script de los formularios de visualización. Para ello basta con invocar a la función:

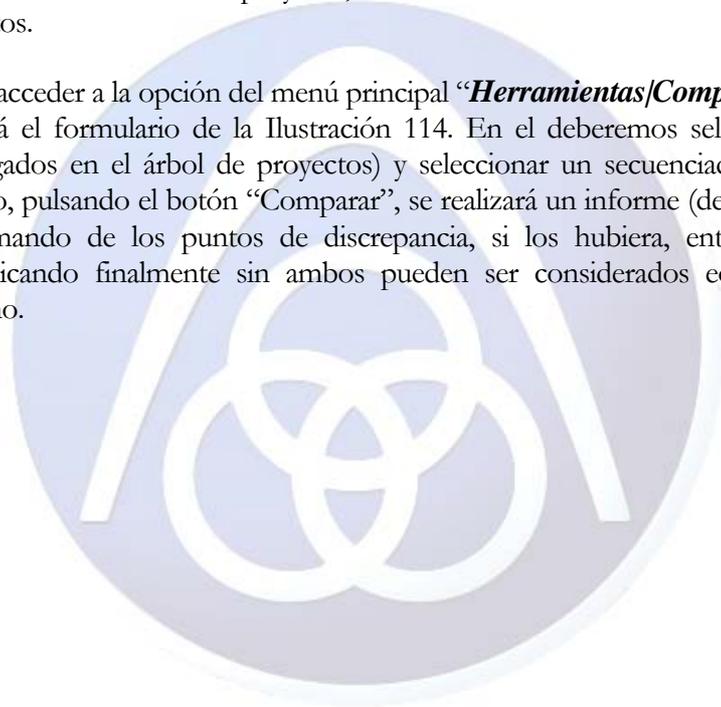
```
ShowReportFailures( TForm Formulario );
```

El Formulario necesario bien puede ser `Application.MainForm`. Cuando se invoca esta función, el entorno (Status Monitor o Designer) se minimiza, y esta aplicación (si está instalada), aparecerá en primer plano. No se podrá volver a acceder al entorno de programación / visualización hasta que no se cierre esta aplicación.

Comparación de Secuenciadores

Como una herramienta extra se suministra este plugin cuya funcionalidad consiste en comparar dos secuenciadores del mismo o distintos proyecto, a fin de encontrar diferencias a nivel de funcionalidad entre ambos objetos.

Para usarla, basta acceder a la opción del menú principal “**Herramientas/Comparar Secuenciadores**” y se nos mostrará el formulario de la Ilustración 114. En el deberemos seleccionar dos proyectos (previamente cargados en el árbol de proyectos) y seleccionar un secuenciador en cada una de las ventanas. Tras ello, pulsando el botón “Comparar”, se realizará un informe (de texto) en la ventana de la derecha informando de los puntos de discrepancia, si los hubiera, entre ambos modelos de secuenciador, indicando finalmente si ambos pueden ser considerados equivalentes (a nivel de funcionalidad) o no.



DESIGNER 3.0

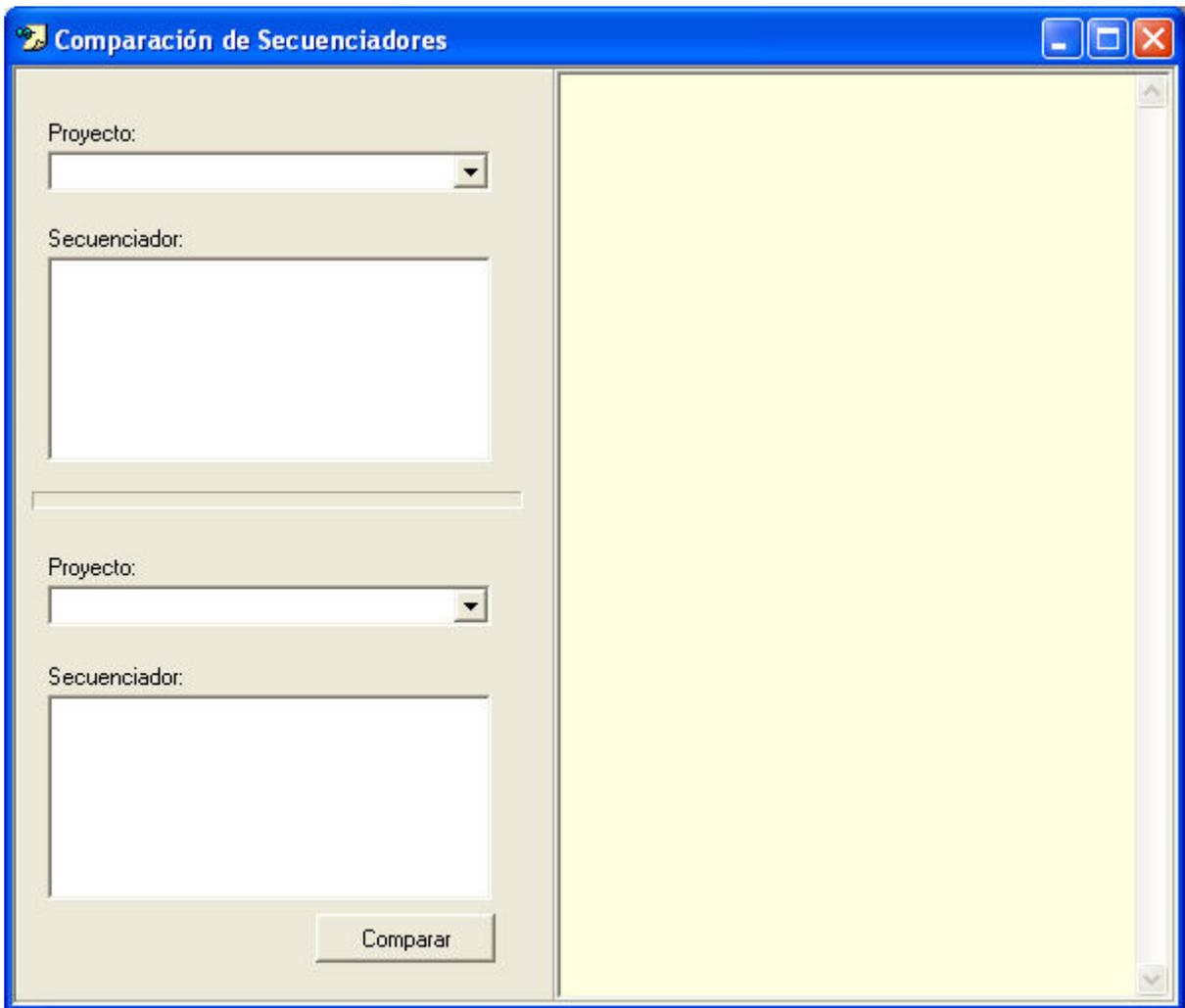


Ilustración 114: Plugin de comparación de secuenciadores.

Comprobación del proyecto.

Este plugin no se encuentra accesible desde el menú principal, ni tampoco desde los botones de la barra de herramientas. Sólo puede ser mostrado usando el menú contextual del botón derecho del ratón sobre el nodo de un proyecto cargado en el árbol de proyectos, y seleccionar la opción “Comprobar proyecto”.

Al hacerlo, aparece un formulario como el mostrado en la Ilustración 115.

DESIGNER 3.0

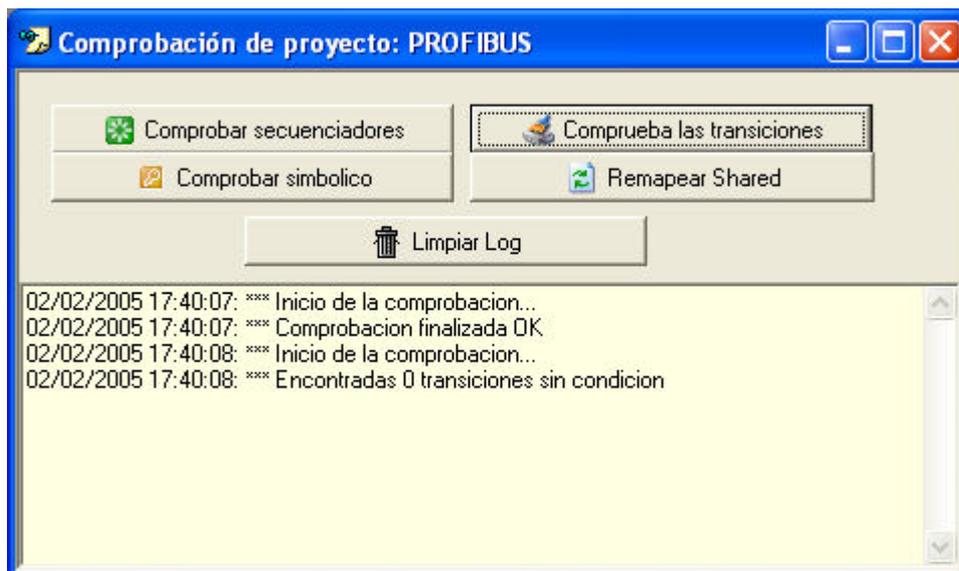


Ilustración 115: Plugin de comprobación de proyecto.

Desde este formulario se pueden realizar varias comprobaciones y algunas acciones correctivas:

Comprobar Secuenciadores: Se comprueba que todos los grafos sean correctos a nivel de las relaciones de herencia (que no aparezcan herencias cruzadas). Si durante la comprobación se encuentran fallos en la definición de secuenciadores, estas se repararán de forma automática.

Comprobar Simbólico: Se buscan elementos en la definición de las variables que no estén bien definidos, por ejemplo, variables PLC sin elemento de bus asociado, o parámetros obsoletos asignados a variables.

Comprobar Transiciones: Se comprueba que las transiciones de todos los secuenciadores del proyecto, buscando que todas sean coherentes. No se pueden permitir transiciones AND 1-N de 1 sólo elemento, por ejemplo, o transiciones sin condición establecida. Si se encuentran transiciones con fallos de ese tipo, aparecen botones en el formulario que se pueden marcar para intentar solucionar esos problemas en una comprobación posterior, volviendo a ejecutar la comprobación.

Remapear Shared: Si el mapa de variables shared aparece muy fragmentado o excesivamente grande, debido a creaciones y borrados arbitrarios de variables durante el desarrollo del proyecto, puede ganarse algo de espacio remapeando las variables shared de forma contigua. Esta opción también puede realizarse desde el plugin de edición del simbólico. Debe tenerse en cuenta que tras un remapeo de variables shared, el proyecto debería ser recompilado y enviado al servicio Galileo en todas las computadoras del proyecto, a fin de que todos los servicios Galileo vean el nuevo mapa de variables.

DESIGNER 3.0

Opciones del Entorno

El entorno de desarrollo dispone de un buen número de opciones que permiten configurarlo de forma que el usuario se sienta más cómodo con el o que llegado el caso, le permitan obtener un mejor rendimiento del mismo.

EL entorno de desarrollo es configurable en una serie de opciones que le permiten solucionar problemas puntuales o para adaptarse mejor a los gustos del usuario. Para acceder a estas posibilidades de configuración basta con usar la opción **Herramientas / Opciones de configuración** que existe en el menú principal de la aplicación, lo que nos dará acceso a las siguientes pantallas:

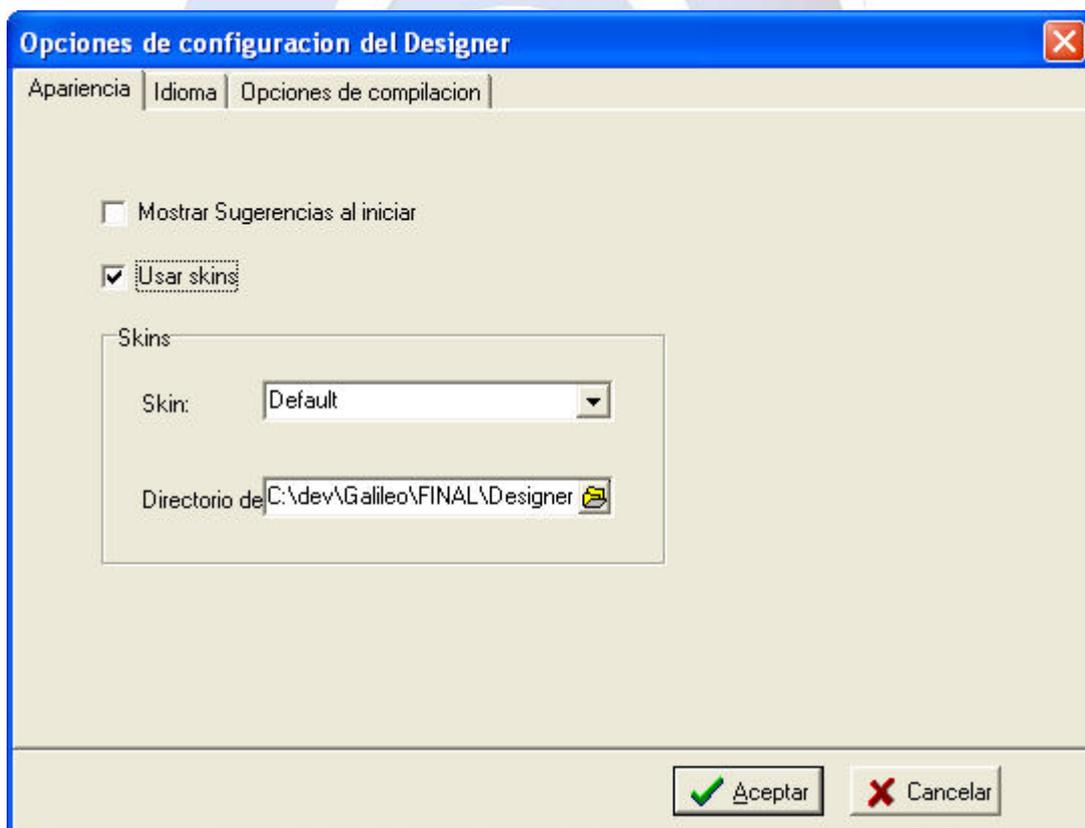


Ilustración 116: Opciones de "skins"

Si “Mostrar sugerencias al iniciar” se encuentra activo, un pequeño cuadro de diálogo se nos mostrara cada vez que iniciemos el entorno de desarrollo, ofreciéndonos algún pequeño consejo sobre como sacarle más rendimiento al entorno. Podemos desactivarlo desde esta pantalla o desde el propio cuadro de diálogo, tal y como se ve si usamos la opción “**Ayuda / Sugerencias...**” del menú principal.

Desde la pestaña “Idioma” podremos ver el código de idioma usado por el sistema en ese momento. No se recomienda cambiar dicho código a no ser necesario, puesto que si se usa uno del que no se dispongan los recursos de idioma necesarios, obtendremos un error de sistema al usar el entorno y este se volverá inutilizable.

La tercera solapa nos permite acceder a algunas opciones de la compilación de proyectos.

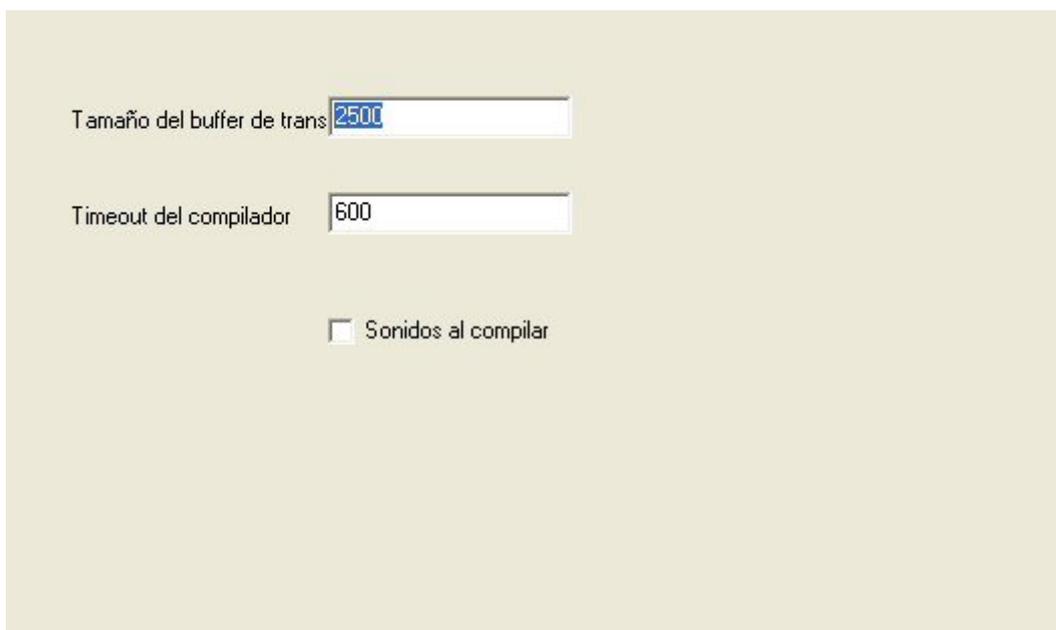


Ilustración 117: Opciones de configuración de la compilación

El tamaño del buffer de transferencia es un tamaño (en Kilobytes) que se usa para pasar datos entre el log de compilación y la ventana de compilación del entorno. En el caso de que la compilación produjese muchos avisos, el aumentar este tamaño puede hacer que la compilación termine un poco antes, a costa de usar una mayor cantidad de memoria.

En cuanto al Timeout de compilación, se refiere al tiempo máximo (en segundos) que el entorno espera antes de terminar el proceso de compilación de forma fulminante. Esto es necesario a fin de no dejar procesos pendientes que bloquearían el sistema. En el caso de compilaciones de proyectos especialmente grandes (o computadoras suficientemente lentas), puede ser necesario aumentar este tiempo a fin de que el proceso termine, aunque ello le lleve muchísimo tiempo.

Por último, la opción de “Sonidos al compilar” activa o desactiva los sonidos que se reproducen al terminar alguna fase de la compilación. Estos sonidos son diferentes según el proceso termine con éxito o con error.

Problemas Típicos

Durante el uso del entorno de desarrollo, una serie de problemas pueden ir surgiendo. En la mayoría de los casos son bien conocidos y el objetivo de este apéndice es dar una solución sencilla a los mismos sin necesidad de esperar a las respuestas del departamento de soporte.

Las comunicaciones son el elemento que más problemas puede causar. La mayoría de las veces es un fallo en las comunicaciones lo que nos impide avanzar u obtener la información que necesitamos para proseguir el desarrollo del proyecto.

Causas habituales de fallos de comunicaciones:

Nombres de computadora inválidos (con espacios en blanco, por ejemplo)

Computadoras que no se pueden resolver por el sistema: Dependiendo del modelo de red donde esté funcionando el sistema, pueden ocurrir problemas con la configuración de red que hagan que un nombre de computadora no sea accesible. Una manera rápida de comprobar si este es el caso es realizar un comando ping a la computadora destino. Si la dirección se resuelve de forma satisfactoria y la computadora está accesible, esta parte estaría correcta. Si no es así, y no podemos resolver el nombre de computadora, podemos usar el el fichero "HOST", situado en :

```
C:\Windows\system32\drivers\etc\host
```

Y añadirle una línea del tipo:

```
Nombre_computadora      xxx.yyy.zzz.ttt
```

Donde xxx.yyy.zzz.ttt sería la dirección IP que tiene la computadora. Tras realizar este cambio, deberemos comprobar con el comando ping de nuevo. Si sigue sin ser accesible, todo apunta a que existen otros problemas de acceso que deben ser consultados con el administrador de la red correspondiente.

Hay que resaltar que la inclusion de esta información en el fichero HOST vuelve a esa direccion estática, lo que quiere decir que cada vez que el computador destino cambie de dirección, tendremos que actualizar manualmente este fichero, o perderemos la comunicación de nuevo.

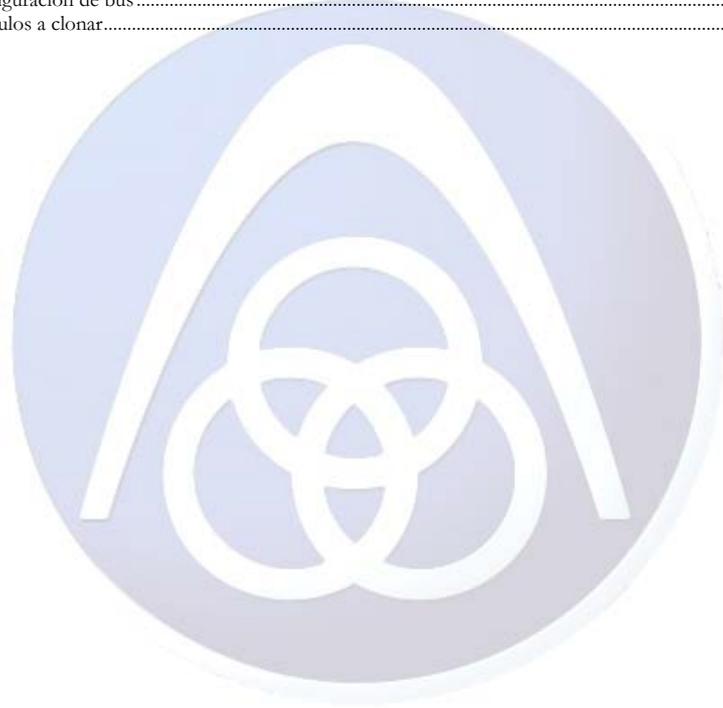
Índice de Ilustraciones

Ilustración 1: Diálogo de aceptación de licencia.....	7
Ilustración 2: Diálogo "readme".....	8
Ilustración 3: Diálogo solicitando un directorio donde instalar la aplicación.....	9
Ilustración 4: Selección de componentes a instalar.....	10
Ilustración 5: Diálogo de tareas adicionales.....	11
Ilustración 6: Diálogo con información de instalación.....	12
Ilustración 7: Pantalla principal de la aplicación.....	13
Ilustración 8: Ventana "Acerca de...".....	19
Ilustración 9: Diálogo "Sugerencia del día".....	20
Ilustración 10: La barra de herramientas principal.....	20
Ilustración 11: Información necesaria para la creación de un nuevo proyecto.....	22
Ilustración 12: Objetos por defecto de un proyecto.....	25
Ilustración 13: Diálogo de creación de una nueva computadora.....	26
Ilustración 14: Diálogo de creación de una nueva tarjeta.....	26
Ilustración 15: Propiedades de una computadora.....	33
Ilustración 16: Máquinas asociadas a la computadora.....	34
Ilustración 17: Plugin de configuración de las propiedades de una tarjeta.....	35
Ilustración 18: Configuración de entradas y salidas de una tarjeta.....	36
Ilustración 19: El plugin de edición de bus.....	37
Ilustración 20: Formulario de compilación.....	38
Ilustración 21: Opciones de compilación de un secuenciador.....	39
Ilustración 22: Opciones de compilación de un proyecto.....	40
Ilustración 23: Formulario de edición de perfiles.....	42
Ilustración 24: Diálogo de creación de un nuevo secuenciador.....	43
Ilustración 25: Plugin de edición de código.....	44
Ilustración 26: Creación de una nueva función.....	45
Ilustración 27: Minibrowser de funciones.....	46
Ilustración 28: Aviso de edición de código heredado.....	47
Ilustración 29: Formulario de mezcla de proyectos.....	48
Ilustración 30: Mezclando visualizaciones.....	51
Ilustración 31: Mezclando plantillas de depurado.....	51
Ilustración 32: Mezclando computadoras.....	52
Ilustración 33: Mezclando esclavos de bus.....	53
Ilustración 34: Mezclando máquinas y zonas.....	54
Ilustración 35: Mezclando la configuración del agente de transportes.....	55
Ilustración 36: Representación de una etapa.....	58
Ilustración 37: Representación de una transición.....	58
Ilustración 38: Ciclo de ejecución de una etapa y una transición.....	59
Ilustración 39: Plugin de edición de secuenciadores.....	64
Ilustración 40: La paleta de herramientas de edición del grafcet.....	65
Ilustración 41: Diálogo de creación de nuevo grafo.....	65
Ilustración 42: Propiedades de un grafo.....	67
Ilustración 43: Propiedades a especificar para una etapa.....	67
Ilustración 44: Propiedades a especificar para una transición.....	68
Ilustración 45: Divergencia AND.....	69
Ilustración 46: Divergencia OR.....	69
Ilustración 47: Convergencia de transiciones OR.....	70
Ilustración 48: Convergencia de transiciones AND.....	70

Ilustración 49: Solapa de propiedades genéricas de un secuenciador	71
Ilustración 50: parámetros de un secuenciador.....	73
Ilustración 51: Formulario de creación de parámetro.....	73
Ilustración 52: Menú contextual de edición de parámetros del secuenciador.....	74
Ilustración 53: El editor de simbólico	77
Ilustración 54: Menú del botón derecho.....	78
Ilustración 55: El formulario "clonar".....	80
Ilustración 56: Menú principal "Vista".....	82
Ilustración 57: Opciones de ordenación	83
Ilustración 58: Menú de parámetros principal.....	83
Ilustración 59: formulario de asignación de parámetros	84
Ilustración 60: Mapa de variables.....	85
Ilustración 61: Campos de la definición de una variable MACHINE.....	87
Ilustración 62: Definición de una Zona	87
Ilustración 63: Definición de una variable PLC.....	88
Ilustración 64: Definición de una variable Shared.....	89
Ilustración 65: Definición de un String.....	89
Ilustración 66: definición de una variable Componente.....	90
Ilustración 67: Definición de una constante.....	90
Ilustración 68: Plugin de configuración del Agente de Transportes.....	93
Ilustración 69: Solapa de edición de la tabla "Trayectorias".....	95
Ilustración 70: Solapa de edición de la tabla "Procesar ordenes".....	96
Ilustración 71: Solapa de edición de la tabla "Averías".....	96
Ilustración 72: Diálogo de generación de configuración.....	97
Ilustración 73: El código PL/SQL generado por la aplicación.....	98
Ilustración 74: Trabajando con la edición del layout.....	100
Ilustración 75: La paleta de componentes "Estándar".....	101
Ilustración 76: La paleta de componentes "Adicional".....	102
Ilustración 77: La paleta de componentes "Win32".....	103
Ilustración 78: La paleta de componentes "Sistema".....	105
Ilustración 79: La paleta de componentes "Base de Datos".....	105
Ilustración 80: La paleta de componentes "Acceso a Datos".....	107
Ilustración 81: La paleta QuickReport.....	108
Ilustración 82: La paleta de componentes "Diálogos".....	109
Ilustración 83: La paleta de componentes "Thyssen-Krupp Layout".....	110
Ilustración 84: La ventana de edición de script.....	116
Ilustración 85: Ventana de operaciones sobre la red Galileo.....	119
Ilustración 86: Ventana de aviso de averías.....	121
Ilustración 87: El pupitre Virtual.....	122
Ilustración 88: El formulario de visualización de variables.....	125
Ilustración 89: Menú contextual de la visualización de valores.....	126
Ilustración 90: Cargando una plantilla de variables.....	127
Ilustración 91: Salvando una plantilla de variables.....	128
Ilustración 92: Pestaña de opciones "Online" de la edición de grafos.....	128
Ilustración 93: La solapa "Tracking" con información sobre una máquina.....	130
Ilustración 94: Copia / intercambio de un tracking.....	131
Ilustración 95: Finalizando una orden desde el editor de tracking.....	132
Ilustración 96: Estado de los flags de la máquina.....	133
Ilustración 97: Flags de fin de orden.....	134
Ilustración 98: Flags de zona de PIE.....	135
Ilustración 99: Grafo en ejecución.....	136
Ilustración 100: Variables de máquina en ejecución.....	137
Ilustración 101: Formulario de acceso al repositorio de secuenciadores.....	140
Ilustración 102: Salvando una revisión al repositorio.....	141
Ilustración 103: El plugin de generación de informes.....	143
Ilustración 104: Plugin de edición de mapas.....	144
Ilustración 105: Algoritmo de relleno de ConfigMap.....	145
Ilustración 106: Algoritmo de relleno de mapa de 2 dimensiones.....	145
Ilustración 107: Algoritmo de relleno de mapa de 3 dimensiones.....	146
Ilustración 108: Offset de mapa 2D.....	146
Ilustración 109: Offset de mapa 3D.....	146
Ilustración 110: representación grafica de un mapa 2D.....	147
Ilustración 111: representación de mapa 3D.....	148
Ilustración 112: El plugin mostrando detalles del objeto "Machine".....	149
Ilustración 113: Informe de averías y transportes.....	152
Ilustración 114: Plugin de comparación de secuenciadores.....	154
Ilustración 115: Plugin de comprobación de proyecto.....	155
Ilustración 116: Opciones de "skins".....	156
Ilustración 117: Opciones de configuración de la compilación.....	157

Índice de Tablas

Tabla 1: Permisos de usuarios de proyecto.....	30
Tabla 2: Tipos de Tarjeta de Control.....	35
Tabla 3: Ejemplo de configuración de bus.....	80
Tabla 4: Ejemplo de módulos a clonar.....	81



DESIGNER 3.0

Índice

A

Agente de Transportes, 15, 16, 93
árbol de proyectos, 13, 14, 15, 16, 17, 23, 26, 43
Averías, 17

C

Compilar, 16, 21
Comunicaciones, 17, 21
Conversion, 150

D

driver, 12

I

Informes, 15, 18

L

licencia, 6, 12
Licencia, 7

M

MACHINE, 86
Minibrowser, 46

O

Oracle, 6

P

Parámetros, 62

R

Repositorio, 14, 20, 139

S

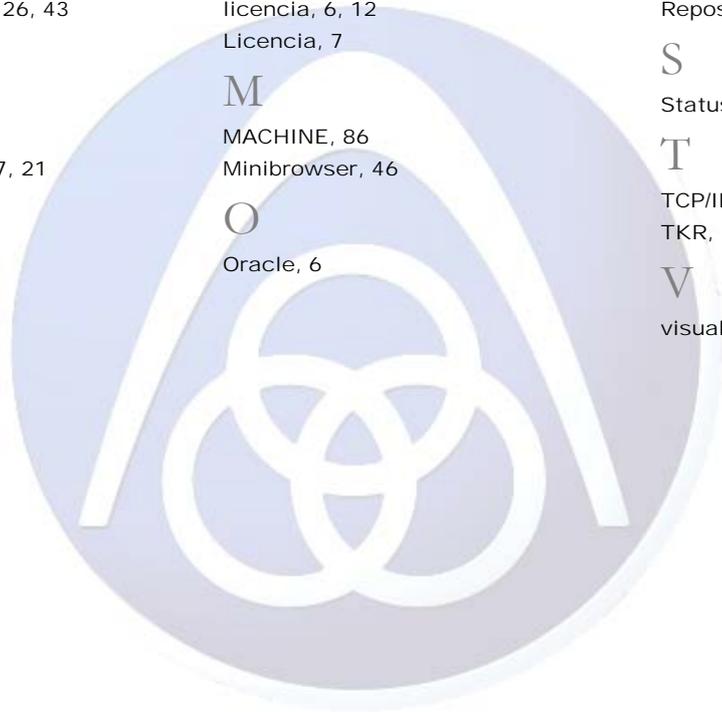
Status Monitor, 6, 18

T

TCP/IP, 3
TKR, 14, 18

V

visualización, 16



DESIGNER 3.0