

UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948

Facultad de Ingeniería Eléctrica.

Departamento de Automática y Sistemas Computacionales

Trabajo de Diploma

Título: “Estudio y desarrollo de aplicaciones sobre plataforma ISaGRAF”

Autor: Raúl González Morffa

Tutor: Dr. Vicente F. Pantuso Nodarse

Santa Clara

2006

“Año de la Revolución Energética en Cuba”

CON SU ENTRAÑABLE TRANSPARENCIA



Universidad Central "Marta Abreu" de Las Villas
Facultad de Ingeniería Eléctrica
Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

Titulo: Estudio y desarrollo de aplicaciones sobre plataforma
ISaGRAF

Autor: Raúl González Morffa

E-mail: rmorfa@uclv.edu.cu

Tutor: Dr Vicente F. Pantuso Nodarse

Profesor Consultante, Dpto de Automática

Facultad de Ingeniería Eléctrica.UCLV

E-mail:pantuso@uclv.edu.cu

Santa Clara

2006

"Año de la Revolución Energética en Cuba"



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Automática autorizando a que el mismo sea utilizado por la Institución, para los fines que estime conveniente, tanto de forma parcial como tal y que además no podrá ser presentado en eventos, ni publicados sin autorización de la Universidad.

Firma del Autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdo de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del Tutor

Firma del Jefe de Dpto.

Donde se defiende el trabajo.

Firma del Responsable de
Información Científico- Técnica

TAREA TÉCNICA

1. Revisión bibliográfica y análisis del software ISaGRAF.
2. Estudio y descripción de la versión ISaGRAF 5.0 y sus componentes Workbench y HiBeam.
3. Desarrollo y descripción de aplicaciones sobre el ISaGRAF 5.0 a través del Workbench y el HiBeam unidos a una Máquina Virtual (NT Target), para que funcionen como proyectos tipo para la docencia.
4. Redacción del informe final.

Firma del Autor

Firma del Tutor

RESUMEN

En este trabajo de diploma se ha hecho un estudio del software ISaGRAF a través de un análisis de sus componentes Workbench y HiBeam, pertenecientes a la versión ISaGRAF 5.0, que fueron utilizados para el montaje y descripción de aplicaciones. Este programa ofrece grandes ventajas en la operación con los sistemas de control y en la automatización, además de ser muy robusto, lo que lo convierte en una opción importante para ser incluido en la docencia, debido a esto se describen los proyectos realizados siguiendo un procedimiento lógico y detallado. Utilizando el Workbench y el HiBeam unidos a una Máquina Virtual fueron confeccionadas las aplicaciones de modo que fueran ganando en orden de complejidad y mostrando diferentes variantes en el trabajo. De los diversos lenguajes de programación pertenecientes a las normas internacionales IEC 61131 e IEC 61499 se trabajó con el lenguaje de contacto (LD), que está constituido por representaciones gráficas y variables booleanas. Todos los proyectos que se realizaron mostraron resultados satisfactorios en la simulación, acordes con la exigencia y eficiencia del producto ISaGRAF.

ÍNDICE

| | |
|---|----|
| INTRODUCCIÓN | 1 |
| CAPÍTULO I. INTRODUCCIÓN AL ISAGRAF | 3 |
| I.1 Introducción al ISaGRAF | 3 |
| I.2 Algunos componentes y opciones fundamentales del ISaGRAF | 6 |
| I.3 La versión ISaGRAF 5.0 | 8 |
| CAPÍTULO II. ESTUDIO DEL WORKBENCH Y EL HIBEAM | 11 |
| II.1 El Workbench | 11 |
| II.1.1 Apariencia del Workbench | 13 |
| II.1.2 Modificaciones | 15 |
| II.1.3 Trabajo con Proyectos | 15 |
| II.1.4 Control de Acceso a Proyectos | 17 |
| II.1.5 Importar y Exportar elementos del Workbench | 17 |
| II.1.6 Montar elementos del Workbench desde un objetivo | 18 |
| II.1.7 Arquitectura de Vínculos y de Hardware | 18 |
| II.1.8 Diccionario | 20 |
| II.1.9 Cableado de Entrada / Salida | 20 |
| II.1.10 Algunos elementos y definiciones importantes | 22 |
| II.2 EL HiBeam | 23 |
| II.2.1 Screen Builder | 25 |
| II.2.2 Barra de Herramientas (Toolbars) | 26 |
| II.2.3 Crear documentos animados | 26 |
| II.2.4 Visualizar Pantallas | 28 |
| II.2.5 Vincular pantallas | 29 |
| II.2.6 Compilar y cargar documentos | 29 |
| CAPÍTULO III. DESARROLLO DE APLICACIONES | 30 |
| III.1 Pasos comunes en la programación en el Workbench | 30 |
| III.2 Primera Aplicación :El Interruptor | 32 |
| III.2.1 Pasos comunes en la programación en el HiBeam | 34 |
| III.3 Segunda Aplicación: Trabajo con señales analógicas y digitales | 37 |
| III.4 Tercera Aplicación: Control en Red de la Máquina Virtual | 41 |
| CONCLUSIONES | 45 |
| RECOMENDACIONES | 46 |
| REFERENCIAS BIBLIOGRÁFICAS | 47 |
| Anexo 1. Representación de los lenguajes de control | 49 |
| Anexo 2. Esquema de operación del ISaGRAF en ambiente Linux | 50 |
| Anexo 3. Selección del lenguaje de programación en el Workbench | 51 |
| Anexo 4. Ejemplo de la descarga del proyecto en la página Web | 52 |
| Anexo 5. Glosario de Términos | 53 |

INTRODUCCIÓN

En el departamento de Automática y Sistemas Computacionales de la Facultad de Ingeniería Eléctrica de la UCLV se disponen de varios softwares de programación de autómatas, muy útiles para la simulación y control de procesos, que aunque reúnen condiciones para la impartición de conocimientos de estas materias de una forma adecuada en la docencia se han ido convirtiendo en tecnologías antiguas. La aparición de modernas técnicas en esta rama científica ha obligado a los profesionales a una constante superación y asimilación de las mismas, nuestro país no puede ser una excepción y tal como se ha indicado por la máxima dirección de la Revolución y el Partido, la elevación de la eficiencia económica está muy ligada a los avances de la ciencia, la innovación y la introducción de nuevas tecnologías, por lo que son considerados objetivos de una alta prioridad, las Universidades y sus respectivas facultades juegan un alto papel en este sentido.

Actualmente se cuenta en el departamento entre otros con versiones de Prosimax, PG95 e ISaGRAF, este último perteneciente a ediciones pasadas. Con la adquisición de una versión moderna de ISaGRAF se pretende insertar la misma en el modelo docente, por las facilidades y nuevas características que puede ofrecer este software de reciente producción, considerado como un patrón en lo que a automatización se refiere.

El ISaGRAF considerado por muchos como un paquete de software, por la forma en que está estructurado, tiene un comportamiento similar al de un autómata programable, es decir cíclico, y es capaz de ejecutar aplicaciones en tiempo real en un sistema computador industrial. Su versión más reciente ISaGRAF 5.0 fue lanzada al mercado en noviembre del 2005 por la ICS Triplex ISaGRAF Inc y ofrece ventajas en el trabajo con los diferentes lenguajes de programación y control industriales, es considerado además como un suministrador global de control abierto por su compatibilidad con las normas internacionales IEC 61131 e IEC61499.

Por sus cualidades propias el ISaGRAF y en especial ISaGRAF 5.0 puede ser aplicado a asignaturas tales como Medios Técnicos de Automatización, Sistemas Automatizados y Computación Aplicada. Con el propósito de demostrar el gran alcance de esta herramienta

para la incorporación de nuevos conocimientos sobre autómatas y su programación así como el trabajo con los sistemas de control y las nuevas tendencias internacionales es que se decide llevar a cabo un estudio de este software para el posterior desarrollo y descripción de aplicaciones sobre el mismo, a través de una forma lógica y amena, y que reflejen su modo de funcionamiento, estas aplicaciones serán puestas a disposición total de la docencia para lograr una mejor asimilación de este moderno producto tanto por parte de los estudiantes como de los profesores. El Laboratorio de MTA del Departamento de Automática y Medios Computacionales cuenta con la licencia de operación de este software gracias a la ICS TRIPLEX ISaGRAF.

El trabajo se estructuró esencialmente de la siguiente manera: Resumen, Tarea Técnica, Introducción, Desarrollo que consta de tres capítulos, Conclusiones, Recomendaciones y Anexos.

En el Capítulo I “Introducción al ISaGRAF” se expone un detallado estudio del ISaGRAF haciendo un recorrido por sus principales características y aspectos a destacar, a través de una revisión bibliográfica realizada, donde se hace referencia especial al ISaGRAF 5.0, la versión en cuestión.

En el Capítulo II “Estudio del Workbench y el HiBeam” se analizan profundamente los componentes Workbench y HiBeam pertenecientes a la mencionada versión mediante la descripción de sus modos de operación y configuración en el desarrollo de aplicaciones. Además se hace mención a los diferentes lenguajes de programación industrial.

En el Capítulo III “Desarrollo de Aplicaciones ” se lleva a cabo el montaje y descripción de aplicaciones mediante la utilización de los componentes Workbench y HiBeam unidos a una Máquina Virtual, donde se utiliza uno de los lenguajes normados internacionalmente, y cuyo objetivo es que funcionen como proyectos para la docencia, demostrando las variantes en el trabajo con el software.

Las conclusiones resaltan los principales resultados obtenidos en el trabajo y se realizan una serie de recomendaciones con el propósito de continuar con el estudio y desarrollo de aplicaciones con este software.

Los anexos muestran un esquema representativo de los lenguajes estandarizados y varias figuras de la operación con el ISaGRAF en la operación y montaje de aplicaciones.

CAPÍTULO I. INTRODUCCIÓN AL ISAGRAF

Introducción

En este capítulo se realizará un detallado estudio del ISaGRAF mediante la descripción de sus principales características así como de las opciones, componentes y ventajas que ofrece en la programación, el control y la automatización en general. Se conocerá de su robustez, estandarización a las normas internacionales, afiliación a los diversos lenguajes de control industrial y compatibilidad con diferentes plataformas. Se hace objeto de estudio además en la evolución de este paquete de software hasta llegar a su versión más reciente ISaGRAF 5.0, donde se tendrá énfasis especial por ser esta última la que se utilizará en el montaje de aplicaciones. Mediante este acápite se pretende demostrar la importancia del ISaGRAF en la industria moderna y su continuo desarrollo.

I.1 Introducción al ISaGRAF

El ISaGRAF es un paquete de software de control industrial formado por herramientas de programación que nos permite crear sistemas de control locales o dispersos. Ofrece una combinación de un motor de control robusto y altamente portátil (Máquina Virtual) y un ambiente de desarrollo de aplicación intuitivo (Workbench). El ISaGRAF cumple con la IEC 1131-3, la única norma estándar industrial reconocida internacionalmente en el ámbito de lenguajes de control industrial.

ISaGRAF es la referencia y de hecho el estándar en tecnología de softwares para la automatización. Introducido en 1990, fue diseñado originalmente para tender un puente sobre la abertura entre los sistemas de microordenadores y los PLC. Ha venido a convertirse en la tecnología principal del software para la automatización abierta, aplicable a la automatización tradicional, al control incorporado y a los mercados de lógica simple.

El ISaGRAF fue el primer ambiente de *Windows* basado en el desarrollo del medio para apoyar completamente a los cinco lenguajes de programación de PLC: Diagrama de funciones secuenciales (SFC), Diagrama de bloques de función (FBD), Diagrama de escalera o contactos (LD), Texto Estructurado (ST), y Lista de Instrucción (IL). Además, para este último en potencia y flexibilidad, ISaGRAF apoya funciones y bloques de funciones escritos en los lenguajes C e IEC1131-3.

El ambiente de desarrollo, también conocido como Workbench (banco de trabajo, en inglés), funciona en *Windows* 3.1, 3.11, 95 o NT y trabaja con los lenguajes de control de la norma estándar IEC 61131. La salida de este ambiente de desarrollo es seleccionable como un código fuente C portátil o código independiente de objetivo CIO (TIC en inglés). La Máquina Virtual ISaGRAF es un poderoso, rápido y optimizado software que ejecuta el código TIC. El banco de trabajo y la documentación de ISaGRAF está disponible en inglés, francés, alemán y japonés, con otras idiomas que se agregarán en el futuro.

Ideal para pequeñas aplicaciones que requieran estar contenidas, su diseño robusto puede manejar un número grande de Entradas/Salidas. Múltiples máquinas virtuales pueden operar en el mismo procesador a diferentes velocidades de barrido. Múltiples procesadores pueden ser incorporados en un solo proyecto, permitiendo la existencia de diferentes sistemas operativos y diferentes procesadores en una sola aplicación.

La Máquina Virtual y todas sus opciones se ofrecen ya listas para ser usadas con *Windows* NT, Linux, CE 3.0 y QNX. Además, este motor de control ha sido diseñado de tal manera que el código de fuente de la Máquina Virtual esté disponible como un juego de herramientas, así le proporciona portabilidad a cualquier fuente de salida en cualquier plataforma de hardware. Los enlaces permiten el uso de los datos de otras máquinas virtuales, ya sean locales o estén por toda la red o en un enlace de comunicación.

La serie de herramientas de ISaGRAF le permiten escribir sus propios *drivers*, agregar bloques de funciones específicos al mercado, conectar a sistemas de nivel más altos o convenientemente hacer su propio producto. Esto último lo convierte en un paquete único que preserva su propiedad intelectual.

Las aplicaciones para ISaGRAF se extienden desde el control simple de la máquina a los procesos complejos y al rendimiento de alta velocidad, alto rendimiento y a las altas tareas de confiabilidad. Con las opciones mejoradas de ISaGRAF el controlador gana elementos tales como la calidad de los datos, tiempos de ejecución extremadamente bajos, secuencias de eventos, tendencias, alarmas, sincronización del procesado, soporte de GPS y redundancia. Por ello casi todos los mayores proveedores de automatización están usando la tecnología ISaGRAF en su nueva generación de productos.

Los principales componentes de control de ISaGRAF son : La aplicación Workbench, meta de tiempo de operación, *drivers* y puertas de comunicación, y compatibilidad IEC 1131.

Las opciones de automatización que se ofrecen son las alarmas y eventos, las tendencias y los paquetes de sobrefallos. Aquí juega un papel fundamental el Juego de Herramientas de Desarrollo.

Algunas de las operaciones que se pueden realizar con el ISaGRAF van desde la programación gráfica y de texto hasta la organización de los trabajos múltiples y cambios del programa en línea. En estas operaciones se tiene en cuenta la lógica de la escala con los bloques de gran alcance de la función para darle la simplicidad de un PLC así como el lenguaje que se utilizará en la programación. Otras de las operaciones permitidas son la auto-documentación y la simulación del programa.

El ISaGRAF goza de una gran aplicación en variedad de industrias: Aeroespacio y Defensa, Industria Automotor, Industria Química, Paquetes de consumo: Bebidas y tabaco, alimentos, productos para el hogar, la salud y la belleza. Además para la electrónica y los semiconductores, computación y comunicaciones, maquinaria y equipamiento de la industria pesada, en la metalurgia para la fabricación y el ensamblaje, así como para los molinos y la fundición. También tiene amplia aplicación en la fabricación de miscelánea, industria farmacéutica, industria del plástico y la goma, la pulpa y el papel y en la industria textil. Es válido en entornos tan dispares como controlar el sistema de gerencia de la generación de energía en portaaviones de propulsión nuclear o en un simple control de un aparato de consumo doméstico.

Las plataformas donde puede operar el ISaGRAF son diversas, *Microsoft Windows NT/2000/XP* y *Microsoft Windows 95/98/ME* están entre las más utilizadas junto a *IBM VM/zVM* e *IBM OS/400* y *Apple Macintosh*.

Con ISaGRAF, el proyecto se divide en uno o varios lazos o “recursos” del PLC, identificando sus plataformas de hardware y definiendo los acoplamientos entre ellos. Los recursos contienen varias unidades de programación o “POUs” (Unidades de Organización de Programas, siglas en inglés), que pueden ser programas, funciones o bloques de funciones. Para el desarrollo de programas, el banco de trabajo de ISaGRAF proporciona los redactores gráficos y textuales de gran alcance e intuitivos, incluyendo barras de herramientas del puerto y las ventanas de tamaños variables y divididas.

Con el uso de ISaGRAF, para eliminar errores, no se requiere al creador volver a la lógica básica de control del proceso ni recordar la sintaxis exacta de todos los lenguajes.

Para supervisar las aplicaciones ISaGRAF proporciona una herramienta gráfica de gran alcance que define las listas de reloj que pueden ser exhibidas en conjunto como cuadros gráficos que se vinculan a las variables del proyecto de ISaGRAF o como listas durante la depuración.

I.2 Algunos componentes y opciones fundamentales del ISaGRAF

Como se mencionó anteriormente los principales componentes de control del ISaGRAF son la aplicación Workbench, la meta de tiempo de operación, los *drivers* y puertas de comunicación y la compatibilidad IEC 1131. En este caso se hará referencia a los *drivers* y las puertas, del Workbench se tendrá referencia en el siguiente capítulo de una forma amplia y detallada.

Las puertas y *drivers* de comunicación del ISaGRAF apoyan la serie de características dominantes siguientes para asegurar un flujo confiable y eficiente de la información en tiempo real a través de su organización:

1. Soporte de la estadística, el diagnóstico, la eliminación de errores, y la recuperación de error automática.
2. Caja de herramientas del *driver* (modificar y certificar sus propios *drivers* para los requisitos particulares)
3. Paquete de función en serie (conectar con los dispositivos usando propiedad serial o conexión TCP/IP).
4. Paquete de ODBC, para tener acceso a datos en una base de datos acomodada en ODBC.
5. Servidor de OPC, para conectar con HMI estándar y otros clientes de OPC.
6. Capacidad maestro-esclavo.
7. Tiempo de estampado en milisegundos.
8. Redes múltiples de entrada-salida en un nodo.
9. Entradas redundantes de las comunicaciones, ayuda redundante de la LAN (sin hardware especializado).

Un componente importante de ISaGRAF es el motor de control *Runtime* que incluye a la Máquina Virtual, está especializado y optimizado para aplicaciones de control de alto funcionamiento, es análogo a Java, ejecuta el Código de Objetivo Independiente (TIC, siglas en inglés) generado por la aplicación desarrollada en el medio, y puede correr

aplicaciones con ciclos de barrido por debajo de los milisegundos. Se encuentra disponible para sistemas operativos mayores (NT, RTX, CE, LINUX, VvWORKS, QNX, OS9, ThreadX, PSOS etcétera). Junto con la Máquina Virtual, el motor de control debe sujetar a otros componentes, como son el director de comunicación de campo, servidores de tendencias y alarmas/eventos, y con el administrador de sobrefallos. La Máquina Virtual (PLC virtual) lee variables y puntos de entrada/salida, ejecuta sus aplicaciones de control de procesos, y escribe las variables y puntos de entrada/salida en un lazo cíclico.

Las opciones fundamentales de automatización del ISaGRAF son las alarmas y eventos, las tendencias y los paquetes de sobrefallos como también se había mencionado anteriormente y que son vitales en el buen funcionamiento del programa.

El paquete de Alarmas y Eventos de ISaGRAF va con un paso superior a los sistemas de dirección de eventos convencionales. Dicho paquete presenta alarmas que inteligentemente y eficientemente mejoran la habilidad para responder rápidamente y con acierto. Solo se necesita dar un click derecho en el nombre de la variable y se comienza a configurar esta variable instantáneamente, esto le hará salvar una enorme cantidad de tiempo en la configuración. El paquete de las alarmas y eventos de ISaGRAF sabe como y cuando entrar en contacto con el personal de control crítico. ISaGRAF le proporciona la notificación inmediata de alarma si usted está en la sala de mando, en el piso de la planta, en su oficina, en su hogar o cualquier lugar del mundo. La alarma inteligente muestra el resumen e historia de las alarmas así como el resumen e historia de la acción. Cada alarma se puede configurar con el índice de ofrecimiento de hasta 16 niveles de alarma o de las respuestas de las opciones del cambio y de la desviación. Cada nivel puede tener su propia opción del agente de acción. Las alarmas se generan en el nivel del regulador, donde funciona debajo de un servidor en tiempo real de alta confiabilidad. Las alarmas se generan a la velocidad del tiempo de ejecución (1 ms de duración aproximadamente).

Las tendencias de ISaGRAF le dan el acceso a los datos en tiempo real e históricos recogidos de controlar el motor. El visualizador de tendencias está empotrado dentro de su contenedor *Active X* que permita que se realce su capacidad del tratamiento y del análisis de datos. Existe un colector y un visualizador de tendencias, cada uno con una función propia. Para la integridad máxima de la seguridad y de los datos, las tendencias se recogen y se almacenan en el nivel del regulador, sus datos estarán seguros aunque se pierda

momentáneamente el sitio de trabajo. El Visualizador de Tendencias permite que los usuarios absorban mucha información rápidamente, los ingenieros y los operadores aprecian la manera excepcionalmente rápida y fácil de ver los datos de las tendencias. La versión profesional de tendencias de ISaGRAF también provee para cambiar la orientación del gráfico y su ampliación, clasificar y alinear la leyenda, exportar datos al fichero y precisar la autoescala.

La versión actual de ISaGRAF incluye un esquema de redundancia simple y eficiente basado en el software y el hardware, llamado sobrefallos. Cuando se permite un sobrefallo para una configuración, automáticamente se indica la duplicación de todos los recursos, es decir, máquinas virtuales, unidas a esta configuración. Los recursos duplicados se unen a una segunda configuración que funcione en un segundo nodo del objetivo, totalmente independiente de la otra. Sobre la falta del regulador primario del objetivo, el secundario, recibiendo actualizaciones regulares de la base de datos en tiempo real del primario, asume el control dentro de un ciclo de exploración.

I.3 La versión ISaGRAF 5.0

ISaGRAF 5.0 es el primer software de automatización del mundo en ser compatible con estándares industriales del IEC 61131 y del IEC 61499. Este software principal está incluido en un poderoso grupo de nuevas características que prometen cambiar la manera en que se construyen los sistemas de control.

Desarrollado por la Comisión Electrotécnica Internacional, el IEC 61499 presenta las pautas para el uso del bloque de funciones en sistemas distribuidos del proceso industrial, y en sistemas de medición y control. Este estándar que emerge proporciona un número de ventajas notables incluyendo:

1. La regulación del flujo de decisiones de control para interactuar con sistemas de control distribuido.
2. Suministran la consistencia de los datos.
3. Suministran los medios para asegurar la operación síncrona entre los dispositivos.
4. Eliminan la necesidad de tener esquemas de sincronización separada.
5. Facilitan el desarrollo y el mantenimiento de los sistemas de control robustos.

El IEC 61499 también tiene la capacidad para encapsular la funcionalidad de la automatización tales que los constructores de la máquina pueden crear los bloques de la función del IEC 61499 para diversos componentes de la misma y necesitar solamente montarlos para alcanzar la operación deseada. Proporciona las ventajas de la programación orientada a objeto en un ambiente accesible a los ingenieros en la automatización.

La tecnología tiene todos los componentes al prototipo y crea un producto de control de alta calidad. Ofrece una combinación del software del motor del control usado por la mayoría de los fabricantes de PLC. Una caja de herramientas está disponible para adaptarla a cualquier plataforma de hardware y sistema operativo.

ISaGRAF 5.0 representa la evolución natural de la tecnología en el mundo, la nueva versión de ISaGRAF ofrece excitantes nuevas características, más flexibilidad para la integración de las soluciones asociadas y lo más notablemente posible: conformidad con el estándar del IEC 61499. Con ISaGRAF 5.0, los usuarios tendrán la capacidad de construir los sistemas de control tradicionales donde las interacciones entre los dispositivos serán reguladas y sincronizadas automáticamente por los diagramas de bloque de las funciones del IEC 61499 mas bien que con el uso de algoritmos manualmente puestos en ejecución, los dispositivos podrían ser PLCs, microcontroladores o instrumentación inteligente del campo tal como metros o válvulas de flujo. ISaGRAF 5.0 también se beneficia de la famosa tecnología integral TIC de las versiones anteriores de ISaGRAF.

Al utilizar ISaGRAF 5.0, por ejemplo, los metros de flujo inteligentes pueden actuar como regulador y controlador de una válvula mientras son parte de un sistema de control distribuido integrado.

Desde su lanzamiento en el 2005 la versión ISaGRAF 5.0 ha sido merecedora de dos importantes premios, el primero fue el Gran Premio en la Innovación o gran premio Leonard, como también se le conoce, en la categoría de Solución de Software, otorgado en el show 2005 de Optimización en la Automatización, en la ciudad de París, por la visión global del análisis a la programación que propone este programa. “El acercamiento se estructura y utiliza objetos gráficos para centrarse más en el aspecto visual que las tendencias de programación actuales, esta solución proporciona más sitio para el análisis, una biblioteca de los bloques de funciones dibujados y que se respete el estándar del IEC 61499 son aspectos muy útiles”, señaló el jurado al entregar el premio a un representante

de ICS Triplex ISaGRAF, compañía que tiene a su cargo la producción y distribución del software y cuyo producto es el buque insignia de la firma.

El segundo premio recibido fue el de los Redactores de la Ingeniería de Control, otorgado a principios del año 2006 en el encuentro de la Semana Nacional de la Fabricación celebrado en la ciudad de Chicago, dicho galardón es presentado anualmente para esos productos considerados por haber tenido un impacto extraordinario en el mercado y la elección se hizo de una lista de cuarenta preseleccionados por parte de los ingenieros en control suscritos que emitieron su voto.

Consideraciones finales

Con este análisis se da por finalizada la introducción al ISaGRAF mediante el estudio de sus características generales y la reseña de su versión mas reciente, la que se utilizará para el montaje y descripción de varias aplicaciones posteriormente. A través de lo expuesto se refleja la gran variedad de servicios que es capaz de prestar el software, la robustez en su funcionamiento y adaptación a diferentes plataformas de trabajo, y la importancia que este ha adquirido en la industria y los sistemas de control, reflejándose además en el impacto que ha tenido en el mercado. ISaGRAF se ha convertido en un producto de excelencia, ideal para la industria, donde cada versión supera con creces a la anterior, y contar con sus servicios garantizan la eficiencia en la operación y control de los sistemas, desde el lanzamiento de ISaGRAF 5.0 hasta los días de hoy su demanda ha crecido notablemente.

CAPÍTULO II. ESTUDIO DEL WORKBENCH Y EL HIBEAM

Introducción

El Workbench y el HiBeam son dos piezas claves dentro de la estructura del ISaGRAF, constituyen elementos fundamentales en su funcionamiento y que garantizan con una correcta operación asociada a una Máquina Virtual el éxito de su proyecto con todas sus aplicaciones incluidas. Por su importancia nos detendremos en este capítulo para realizar a fondo un análisis del Workbench y el HiBeam pertenecientes a la versión ISaGRAF 5.0, que en esta nueva variante incluyen nuevas características como la compatibilidad con la norma IEC 61499 y que mantienen algunas de ediciones anteriores dígame operar con los diferentes lenguajes de control estandarizados internacionalmente. El desarrollo de proyectos será otro de los aspectos a tratar unido a la conexión ente el Workbench y el HiBeam dentro del proceso de operación.

II.1 El Workbench

El Workbench es el medio en el cual se desarrollan los proyectos de control multiprocesos hechos para correr en máquinas virtuales sobre componentes de hardware llamados nodos designados. El desarrollo de los procesos consiste en crear proyectos hechos por configuraciones representando nodos individuales en el cual una o más instancias de recursos (por ejemplo máquinas virtuales) son descargadas. En tiempo real las máquinas virtuales corren en estos nodos. Los proyectos pueden ser desarrollados usando cualquiera de los cinco lenguajes de la IEC 61131 estándar y cuyas nombres aparecen en inglés:

SFC: *Sequential Function Chart* (o Grafcet). (Diagrama de Funciones Secuenciales).

FBD: *Function Block Diagram*. (Diagrama de Bloques de Función).

LD: *Ladder Diagram*. (Diagrama de Escalera o contactos).

ST: *Structured Text*. (Texto Estructurado).

IL: *Instruction List*. (Lista de Instrucción).

Además se puede usar el *Flow Chart* (FC) (Diagrama de Flujo) y el lenguaje de la IEC 61499 (por ejemplo para métodos de distribución) que permite la distribución de bloques de funciones alrededor de recursos múltiples. Cuando se trabaja, los recursos son

compilados para producir rápidos “Códigos Independientes Designados” (TIC, siglas en inglés) o código C.

Dentro de los recursos se pueden declarar variables usando tipos simples (*Boolean* , *integer*, *real*, *string* y *time*) o que el usuario defina los tipos como arreglos o estructuras. Para definir variables se pueden preparar alarmas, eventos y tendencias. Además la comunicación en el campo permite conectar variables al equipo en ese medio. Los recursos pueden compartir variables que usan los lazos internos o externos, los internos son entre los recursos dentro del mismo proyecto y los externos son entre recursos que pertenecen a diferentes proyectos. Para programas IEC 61499 los lazos entre bloques de funciones que se declaran en diferentes recursos son automáticamente creados .

Se pueden crear proyectos en plataformas de desarrollo de *Windows* en el Workbench y editores de lenguaje. El Workbench gráficamente representa y organiza configuraciones, recursos, unidades de organización de proyectos (POU, siglas en inglés) y redes dentro de proyectos desde seis puntos:

1. Arquitectura de Vínculos.
2. Arquitectura de Hardware.
3. Diccionario.
4. Cableado de Entrada/Salida.
5. Lista de Distribución.
6. Lazos.

Las bibliotecas se componen de configuraciones y recursos que permiten definir funciones y bloques de funciones para reutilizarlas a lo largo de los proyectos. Los recursos individuales de la configuración haciendo un proyecto son transmitidos usando la red ETCP o ISARSI (eslabón de serie) a los nodos designados en los sistemas operativos de tiempo real actuales. La comunicación entre configuraciones puede ser implementada usando una red TCP/IP, o se puede escoger para implementar cualquier otro tipo de red.

Se puede escoger simular un proyecto, después de construirlo, usando herramientas procesadoras de alto nivel , antes de realmente transmitir los recursos haciendo configuraciones a los nodos designados.

Usted puede poner cuatro niveles de control de acceso a las aplicaciones del Workbench:

1. Contraseña de protección y modo de “sólo lectura” para proyectos completos.

2. Contraseña de protección y modo de “solo lectura” para recursos individuales.
3. Contraseña de protección para POU individuales.
4. Contraseña de protección para un objetivo.

II.1.1 Apariencia del Workbench

El Workbench está compuesto por una serie de componentes que conforman su ambiente de trabajo en el desarrollo de las aplicaciones :

1. Barra de Título (*Title Bar*).
2. Barra de Menú (*Menu Bar*).
3. Barras de Herramientas (*Tools Bars*).
4. Espacio de Trabajo (*Workspace*).
5. Ventana de Salida (*Output Windows*).
6. Barra de Estado (*Status Bar*).

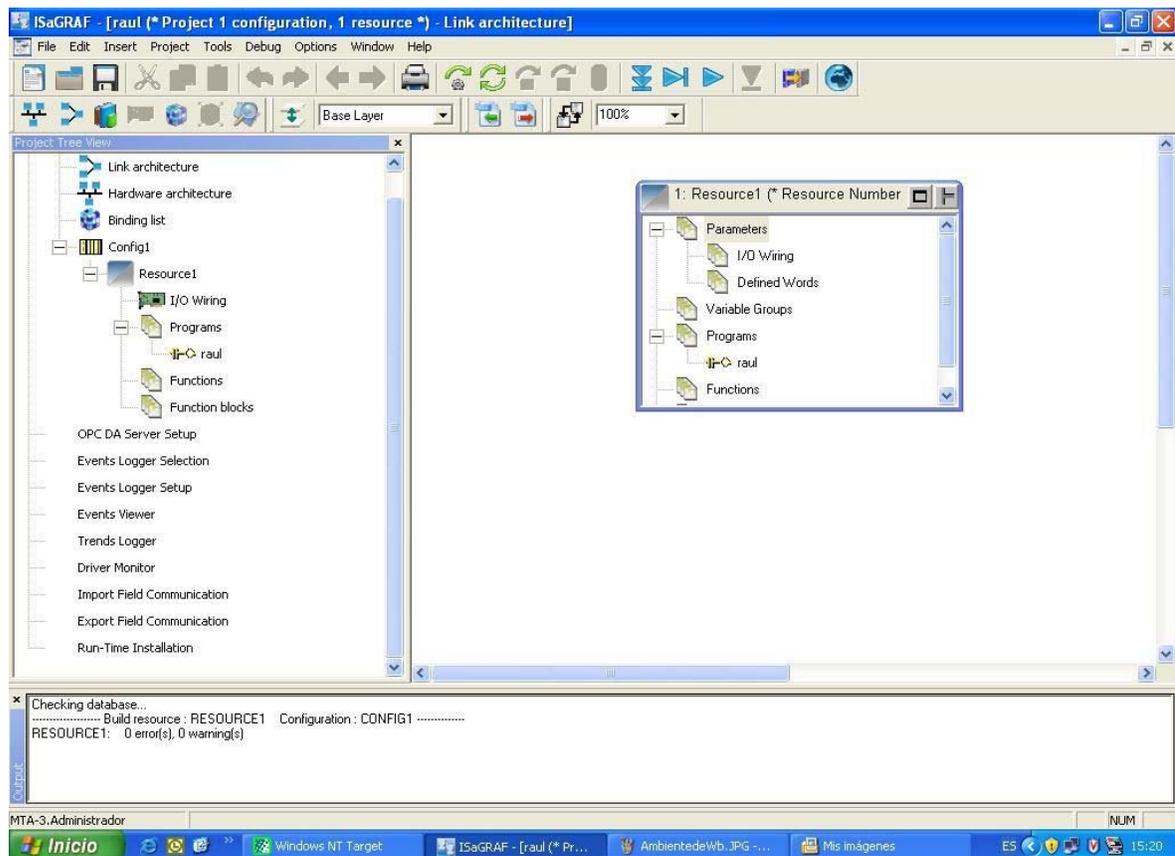


Figura II.1 Ambiente del Workbench.

-Barra de Título: Muestra los nombres de las aplicaciones y el nombre de los ficheros del proyecto activo si alguno es abierto, junto con la vista actual (Arquitectura de Vínculos, de Hardware, Diccionario o Cableado de E/S).

-Icono de Control: Se encuentra al final a la izquierda, es usado para acceder al Menú de Control. Doble click sobre el icono de control cierra el Workbench.

-Menú de Control: Es usado para colocar la ventana principal o para salir.

-Botones de *Windows*: Los botones estándares aparecen a la derecha y al final de la barra de título. Se usan para reclasificar el tamaño o cerrar el *Windows*.

-Barra de Menú: Las opciones disponibles de la Barra de Menú difieren levemente de la vista de la Arquitectura de Hardware y de la de Vínculos de los proyectos. Existen opciones para ficheros (*FILE*), edición (*EDIT*), insertar (*INSERT*), para proyectos (*PROJECT*), herramientas (*TOOLS*), depurar (*DEBUG*), opciones (*OPTIONS*), ventanas (*WINDOWS*), y una ayuda (*HELP*). Cuando no está abierto ningún proyecto solo *FILE* y *HELP* son visibles. Si se abre un proyecto y no se muestra en Cascada o en modo “*Tile*” la Barra de Menú tiene el Icono de Control a la izquierda y este muestra la vista corriente.

-Barra de Herramientas: Varias barras de herramientas realizan diversas tareas que están disponibles en la vista de las Arquitecturas de Hardware y de Vínculos:

1. Barra de herramientas estándar.
2. Barra de herramientas de depurar.
3. Barra de herramientas de botones de *Windows*.
4. Barra de herramientas de capas.
5. Barra de herramientas de fuentes de versión.
6. Barra de herramientas de opciones.
7. Barra de herramientas de cableado de E/S.

Mientras se realizan las tareas del Cableado de E/S en la vista de esta, su barra se hace disponible. La Barra de Herramientas se puede mostrar u ocultar y mover además.

Espacio de Trabajo: Puede ser dividida en un máximo de cuatro vistas simultáneas, donde las sub ventanas son ampliadas independientemente. Cuando se edite en SFC, FC, LD y FDB, el POU también se puede ajustar el factor de ampliación a través del editor de lenguaje, ya que normalmente se hace por la opción “*LAYOUT*” .

Ventana de Salida: Muestra información resultante de la construcción de proyectos, recursos y programas, así como puede mostrar también errores en el *runtime* del Workbench. Esta ventana se muestra automáticamente cuando se realiza un programa y está disponible desde el editor de lenguaje.

Barra de Estado: Aparece como un botón de la ventana principal mostrando información sobre comandos, operaciones y proyectos.

II.1.2 Modificaciones

Se pueden elegir cambios de colores y fuentes para varios aspectos del Workbench al igual que para las preferencias de trabajo del sistema. Para modificar los colores y fuentes se usa:

- Para el Diccionario: Fijar el usado para texto, alcance y casos (*TEXT*, *SCOPE*, *INSTANCES*).
- Para el editor IL y ST: Fijar el usado para *background* y texto (sintaxis básica).
- Para el editor FDB (que incluye IEC 61499): Fijar el usado para *background*, texto, conexión y líneas del contorno del elemento, sombra de líneas y elementos seleccionados, y además el camino para los elementos principales.
- Editores de LD, FC y SFC: Fijar el usado para *background* y texto, y también el camino para los elementos principales.
- Editor de FDB y LD: Puede fijar el color por los comentarios y por los valores Booleanos (*True* y *False*) mostrados mientras que adentro esté en modo “*Debug*”.

II.1.3 Trabajo con Proyectos

Se puede trabajar con los proyectos en ISaGRAF en uno de los dos modos de edición de proyectos: Normal y Recurso Simple. El modo normal permite acceder a un usuario a todos los recursos y POU hechos en el proyecto, mientras que en el modo de recurso simple otros usuarios ajenos no pueden acceder a proyectos o sus recursos. Antes de abrir un proyecto en modo normal, múltiples usuarios pueden acceder a recursos individuales de un proyecto para propósitos de edición (por ejemplo: modo de edición de recurso simple). El modo de edición de recurso simple limita el acceso para usuarios individuales a un recurso y su POU, y otros usuarios pueden acceder a otros recursos del mismo proyecto.

Hay que asegurarse de construir un proyecto completo en modo normal antes de editar recursos simples.

Sólo un usuario puede acceder a recursos a cualquier hora dada, mientras esté en uso el recurso permance bloqueado para todos los demás usuarios. Por ejemplo, cuando se edita un proyecto en modo normal, todos los recursos que componen el proyecto son automáticamente bloqueados para su uso excepto estos que están abiertos en ese momento en modo de recurso simple. Los recursos abiertos son mostrados en el espacio de trabajo (*Workspace*) pero permanecen bloqueados y marcados en gris con el símbolo de bloqueo en su Barra de Título (*Title Bar*). Para desbloquear recursos abiertos corrientes en modo de recurso simple se hace por otro usuario seleccionando el recurso, entonces se elije "Unlock Resource" del Menú de Herramientas (*Tool Menu*).

Con precaución se debe decir que la opción de desbloquear un recurso solo se debe usar cuando sea necesario. Cuando se desbloquea un recurso abierto en ese momento por otro usuario se debe asegurar que el Workbench remoto no esté corriendo por mucho tiempo.

El Workbench automáticamente asigna un nombre de usuario a un proyecto cuando corre o está en la red. Este nombre es mostrado en la Barra de Estado (*Status Bar*) y en las propiedades de control de acceso de recursos. La asignación del nombre depende del modo de edición:

Modo Normal: El nombre es siempre "_ISaGRAF_Admin_"

Modo de Recurso Simple: El nombre es el del usuario de Windows para el usuario que edite el recurso.

Los recursos corrientes abiertos por otro usuario asignan el nombre de ese usuario en sus propiedades. En el modo de recurso simple un proyecto es mostrado en la vista de la Arquitectura de Vínculos con el proyecto e identificación del recurso en la Barra de Título (*Title Bar*) del recurso simple. La vista de Arquitectura de Hardware y la lista obligatoria no son permitidas. En el Diccionario y el Cableado de E/S solo las variables y el cableado definido por el recurso son mostrados. El límite de las variables de otros recursos al igual que los tipos y palabras definidas son en modo de solo lectura, mientras que en modo de recurso simple se puede cambiar el proyecto a correr o el modo de simulación.

Cuando se trabaja en modo normal se pueden realizar las siguientes operaciones: Crear proyectos, renombrarlos y adicionarles una descripción. En tanto cuando es un proyecto en

modo de recurso simple se pueden realizar varias operaciones aunque con limitaciones: Abrir y cerrar proyectos, salvarlos, imprimirlos y acceder a ellos.

Para el caso de abrir y cerrar proyectos, solo se puede abrir uno a la vez. En el caso que se realicen cambios al abrir un proyecto el sistema automáticamente avisa sobre salvar dichos cambios antes de cerrar el proyecto o abrir otro. Para abrirlos deben estar en uno de los dos tipos de modos de edicion conocidos. El nombre del fichero de proyectos es siempre “PRJLIBRARY.MDB” y el nombre del directorio de proyectos representa el nombre del proyecto dado. Cuando se abre un proyecto o se crea, la vista de la Arquitectura de Hardware y de Vínculos son despejadas; y si se relocaliza un proyecto es necesario redefinir el vínculo dentro del Workbench antes de abrirlo.

Para salvar proyectos se utiliza el nombre de este para crear una única estructura de directorio y se guardan en una base de datos cuyo nombre en inglés es “MS-Access”, así como también los ficheros relacionados con el proyecto. Los cambios hechos en proyectos en modo de recurso simple solo se pueden salvar para los recursos editados.

II.1.4 Control de Acceso a Proyectos

En el tema de la seguridad de los proyectos, se puede controlar el acceso a estos a través de una contraseña (*password*) o aplicar el modo de “sólo lectura” para el proyecto completo y que incluye todos los recursos y POU hechos en este. Sin embargo los recursos individuales y POU pueden tener su propio control de acceso a través de una clave asignada. Mientras esté aplicado el método de “sólo lectura” no se podrá construir un proyecto. Cuando se importen y exporten proyectos teniendo el control de acceso de los mismos las claves definidas se conservan. Para fijar el control de acceso a un proyecto se puede elegir que usuarios que no tienen contraseña pueden abrir el mismo en el modo de “sólo lectura”.

II.1.5 Importar y Exportar elementos del Workbench

Se puede importar y exportar del Workbench elementos como proyectos, configuraciones, recursos y POU de un proyecto a otro. Cuando se exporta un elemento este se remueve del proyecto y crea un fichero comprimido de intercambio (.PXF) agrupando todos los datos con excepción de los de la lista de espías y la información que se ejecuta paso a paso, esto

permite además copiar y pegar elementos de un proyecto a otro. Al importar y exportar elementos que tienen control de acceso las claves definidas son conservadas .

Si se quiere importar elementos es imprescindible que previamente hayan sido exportados y almacenados como ficheros comprimidos de intercambio, y que no tengan el mismo nombre de otros elementos del proyecto. Antes de importar un elemento se puede elegir la opción de crear una reserva automática del proyecto.

II.1.6 Montar elementos del Workbench desde un objetivo

Desde cualquier proyecto se pueden montar elementos del Workbench a otro proyecto cuando los códigos de los recursos han sido almacenados en un objetivo (si existen almacenes no volátiles para la plataforma). El fichero de fuente del elemento está comprimido y contiene todos los datos de dicho elemento, y tiene el mismo formato (.PXF) usado cuando se importan y exportan elementos de un proyecto a otro. Antes de montar el fichero se necesita descargar su código de fuente del objetivo.

El proceso de montaje consiste en colocar el fichero de fuente que contiene el elemento del objetivo a la computadora local para el acceso al mismo, entonces se importa el elemento en el proyecto desde el fichero. Como se conoce, antes de importar se puede crear una reserva automática del proyecto.

Es preciso chequear el tipo de red y la conexión en la cual se monta el elemento así como sus parámetros (dirección IP para ETCP).

II.1.7 Arquitectura de Vínculos y de Hardware

La Arquitectura de Vínculos tiene una vista que muestra gráficamente los recursos del proyecto y las transmisiones de datos de recursos entre ellos. Sale por defecto en el Workbench y permite crear recursos y transmitirlos, definir grupos de variables, crear y manipular POU, manejar el cableado de E/S. Cada recurso es mostrado en ventanas separadas de la arquitectura y en el caso de que se cree uno nuevo este automáticamente es asignado a la primera configuración, además se pueden renombrar y copiar desde otra fuente y pegarlos así como borrarlos. Si la ventana del recurso presenta en su parte superior izquierda un triángulo azul, el recurso se encuentra en modo de edición, y si el triángulo es de color verde está en modo de simulación. Las variables también se pueden importar y

exportar, ya sean alarmas o eventos de otro tipo, usando ficheros delimitados por coma (CSV, siglas en inglés) en el editor de texto.

La Arquitectura de Hardware muestra gráficamente las configuraciones entre un proyecto y sus vínculos de red. Mediante esta arquitectura se pueden manejar varios aspectos de un proyecto, como puede ser crear configuraciones, insertar recursos a las configuraciones, mover recursos entre configuraciones, crear redes, definir propiedades de recursos de red o configuración de conexión .

Cuando se trabaje en la configuración no se debe cambiar el objetivo de la misma porque puede causar la destrucción del cableado de E/S de todos los recursos dentro de la configuración y conexiones de red. Se deben asignar objetivos de configuración como primer paso en el desarrollo del proyecto y cuando se instalen las opciones avanzadas se pueden cambiar si se descargan las opciones más características como son alarmas y definición de eventos, tendencias y configuración de servicios.

La red una vez creada y definida sus propiedades permite la comunicación entre configuraciones. Es necesario utilizar el llamado *Network Driver* (Controlador de Red) para definir las propiedades en el tiempo que se crea. La Figura II.2 muestra el ambiente de la Arquitectura de Hardware:

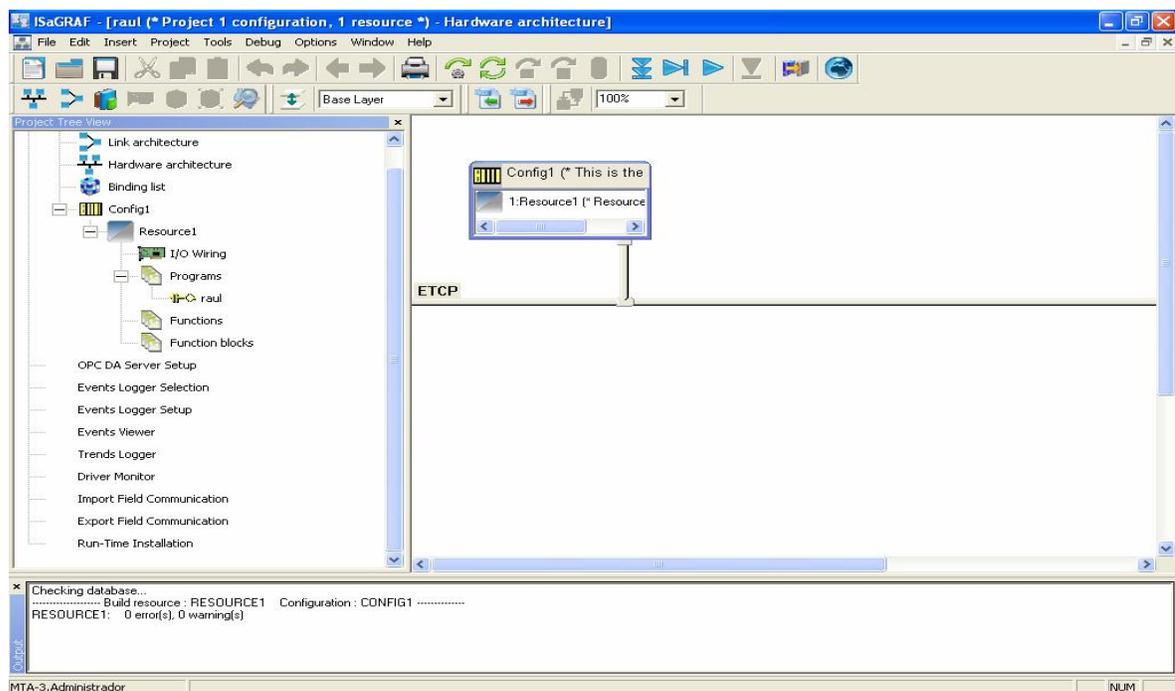


Figura II.2 Arquitectura de Hardware

II.1.8 Diccionario

El diccionario es la herramienta de edición usada para la declaración de variables, funciones, parámetros de bloques de funciones, tipos de usuarios y palabras definidas del proyecto. Los componentes del diccionario están clasificados y se dividen según la jerarquía en cuatro grupos, y es necesario declarar variables antes de proceder al proceso de cableado de E/S.

El diccionario se muestra maximizado en el espacio de trabajo, donde se refleja su menú de opciones. Dentro de los grupos del diccionario se pueden crear estructuras y realizar operaciones con las mismas y además trabajar con las tablas donde se muestran las características y los valores de los componentes a través de las palabras definidas.

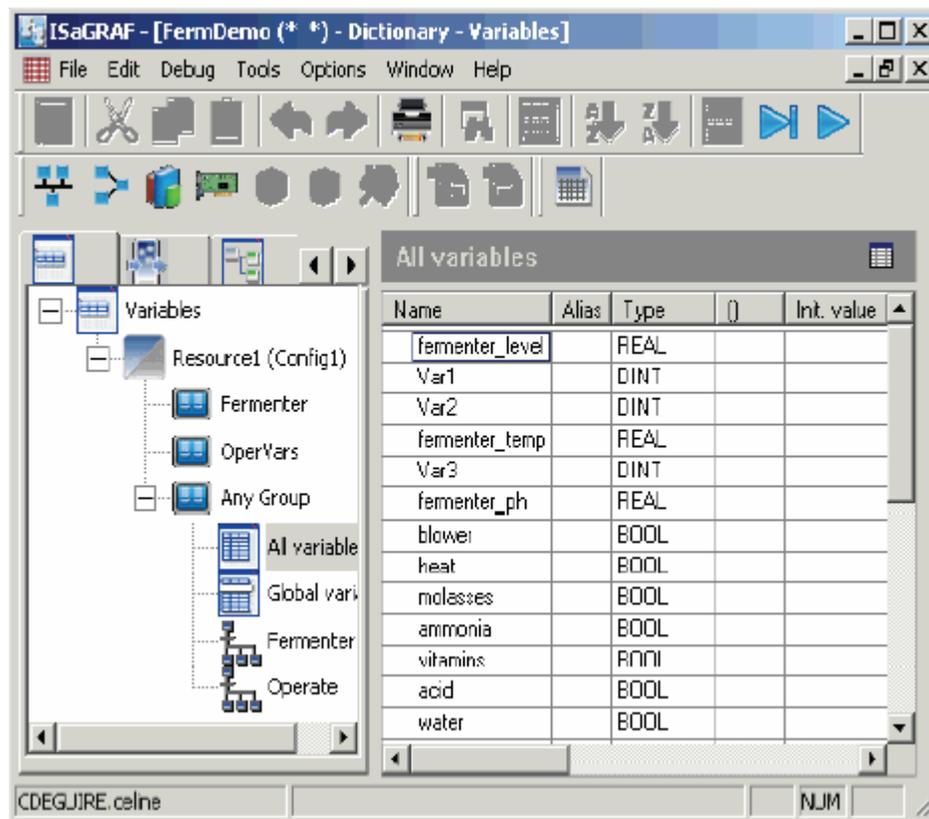


Figura II.3 Ambiente del Diccionario

II.1.9 Cableado de Entrada / Salida

El Cableado de Entrada/Salida permite definir vínculos entre variables definidas en el proyecto y los canales de desarrollo existentes en el sistema de objetivo. El cableado se

II.1.10 Algunos elementos y definiciones importantes.

-Distribución IEC 61499: Permite la supervisión de los programas de la IEC 61499 a través de los múltiples recursos del proyecto. La distribución está compuesta por una ventana superior y una inferior, las cuales sirven para mostrar las configuraciones de los recursos del proyecto y modificarlas, y para mostrar la distribución de los programas de la IEC 61499 con sus bloques de funciones declarados en configuraciones individuales o recursos.

-Editores de Lenguaje: El Workbench presenta varios editores de lenguaje, como el *Common Editor Features* (Editor de Características Comunes, en español) que se utiliza para los lenguajes SFC y FC, y un editor multilingüe. En la Barra de Herramientas aparecen los editores de que se dispone.

-Bibliotecas: Son proyectos especialmente compuestos de configuraciones y recursos en los cuales se definen funciones y bloques de funciones para ser reutilizados totalmente en proyectos de ISaGRAF. Las bibliotecas permiten además modular proyectos y aislar funciones y bloques de funciones para poder validarlos por separado. En las bibliotecas podemos encontrar POU y definición de variables globales, y antes de poder usarlas es necesario crearlas (se crean igual que se crea un proyecto).

-Depurar o eliminar errores: Cuando se desarrolla una aplicación se pueden detectar los errores y eliminarlos, para ello existen dos métodos:

1. Modo de simulación: En este caso las entradas y salidas no se manejan por la máquina virtual. El resto es ejecutado por la plataforma estándar de *Windows* y cada recurso será ejecutado por una máquina virtual en la PC que funciona el Workbench.

2. Modo Online: Cada recurso es ejecutado por una Máquina Virtual en una plataforma real. Se requiere una operación de descarga de los códigos de cada recurso a la plataforma correspondiente.

Para permitir depurar un proyecto, primero se debe hacer que se use el comando "*Build Project*".

-Referencia a los lenguajes: Los lenguajes que se utilizan en el trabajo con el Workbench (SFC, FC, ST, IL, FBD y LD) sirven para describir las Unidades de Organización de Programas (POU, siglas en inglés), y varios de estos unidos a la vez conforman los recursos. Estos lenguajes definen la arquitectura del proyecto y en un mismo programa no

se pueden usar varios lenguajes a excepción del LD y FBD. Cuando se crea un programa se decide que lenguaje se va a utilizar y una vez hecho no se puede cambiar.

El lenguaje SFC (*Sequential Function Chart*) es utilizado para describir operaciones secuenciales y el proceso se representa como un sistema de pasos bien definidos ligados por transiciones.

El FC (*Flow Chart*) es el lenguaje gráfico usado para describir operaciones secuenciales. El diagrama del FC está compuesto de acciones y pruebas, y entre ellos está el vínculo orientado que representa el flujo de datos.

El lenguaje FBD (*Functional Block Diagram*) es otro lenguaje gráfico, el cual permite que el programador construya procedimientos complejos tomando funciones existentes de la biblioteca estándar o de la sección de funciones y bloques de funciones.

El LD (*Ladder Diagram*) es una representación gráfica de ecuaciones Booleanas combinando Contactos (argumentos de entrada) con Bobinas (resultados a la salida). Este lenguaje permite la descripción de pruebas y modificaciones de datos tipo *Boolean* poniendo símbolos gráficos dentro del diagrama del programa.

El lenguaje ST (*Structured Text*) es un lenguaje estructurado de alto nivel diseñado para procesos de automatización, que se usa principalmente para implementar procedimientos complejos que no pueden ser fácilmente expresados con lenguajes gráficos.

El lenguaje IL (*Instruction List*) es el lenguaje de más bajo nivel, las instrucciones siempre están relacionadas con los resultados corrientes (o registro IL). El operador indica la operación que se debe hacer entre el valor actual y el operando.

El lenguaje IEC 61499 es el método de distribución que permite repartir bloques de funciones individuales IEC 61499 que pertenecen a programas del mismo tipo a través de recursos múltiples. Sus bloques de funciones se encuentran disponibles en la biblioteca de la IEC 61499.

II.2 ELHiBeam

El HiBeam lo trae el ICS Triplex ISaGRAF enclavado con tecnología Web, este gira su dispositivo o controlador hacia la Web acompañante y un servidor de datos. Válvulas, bombas, metros pueden ahora mostrar sus datos sobre internet. El HiBeam incluye herramientas para interfaces con controladores basados en datos a través de un buscador

estándar de Web, y son utilizados ambos para mostrar los datos del dispositivo y/o actuar en el controlador de datos remotamente colocado sobre internet. El servidor de datos empotrado es un evento básico y minimiza el uso del ancho de banda.

La Pantalla de Construcción (*Screen Builder*) genera las páginas industriales del HTML. En el tiempo de ejecución se recopila y se envía el dato, mientras que la Caja de Herramientas le da acceso al código de fuente portátil de modo que usted pueda poner la tecnología de HiBeam en ejecución en cualquier dispositivo.

Los datos del HiBeam, FTP y servidores Web pueden ser encajados en cualquier posición: En un controlador, en un sistema SCADA/RTU o en un sensor o transmisor. En cada arquitectura el servidor de datos y de Web son usados para recopilar datos del controlador o dispositivo y distribuir dichos valores de datos a aplicaciones de Java corriendo en buscadores Web.

La comunicación entre el servidor de datos y las aplicaciones es rápida, y los eventos basados para pequeñas redes y alto funcionamiento se cargan. El sistema del HiBeam permite a las aplicaciones (operando en una pantalla) la comunicación con varios servidores de datos que operan sobre los controladores o corriendo en plataformas de hardware separadas. La implementación del servidor de datos y su sistema de comunicación han sido diseñados para una alta portabilidad, este garantiza un alto funcionamiento y requiere pocos procesadores o recursos de memoria.

La Caja de Herramientas de Desarrollo del HiBeam permite un alto volumen de fabricantes de equipos para implementar los servidores de HiBeam en muchos motores de ejecución diferentes, usando la Pantalla de Construcción (*Screen Builder*) estándar para diseñar páginas gráficas o sus procesos o sistemas.

Un conjunto completo de productos de la Caja de Herramienta de Desarrollo son disponibles para implementar a los servidores y al *Screen Builder*, que incluye:

1. Aplicaciones de JAVA habituales.
2. Interface de comunicación.
3. Implementación de herramientas para el HiBeam Screen Builder.

La Caja de Herramientas de Desarrollo del HiBeam proporciona el código "C" desarrollador de ficheros de fuente que requieren del puerto del servidor de datos para

plataformas de hardware específicas y sistemas de operación, vínculos con datos de otros servidores o cualquier otro dato de fuente e implementación del *Screen Builder*.

II.2.1 Screen Builder

La Pantalla de Contrucción (*Screen Builder*) es el medio en el que se crean pantallas con gráficos usando interfaces que consisten en páginas con aplicaciones de Java, desde la cual se puede monitorear o correr procesos de control desde cualquier computadora local o una locación remota, vía red o por internet. Se vinculan estas pantallas con proyectos de control creados en el Workbench. Una aplicación operada por un programa ejecutado desde otra aplicación puede consistir en varias pantallas vinculadas usando botones *hot link*.

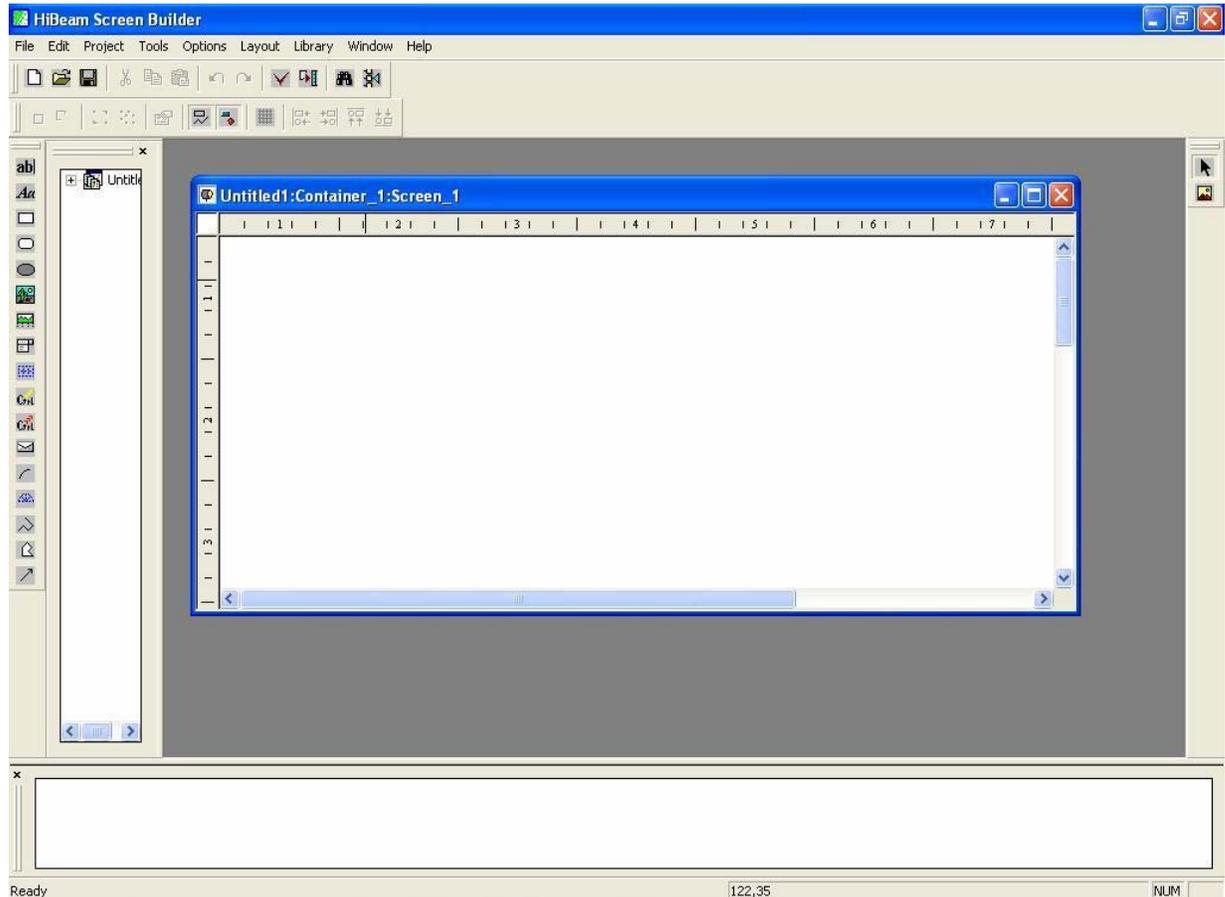


Figura II.5 Ambiente del HiBeam Screen Builder.

II.2.2 Barra de Herramientas (Toolbars)

La Barra de Herramientas cuenta con cuatro barras de trabajo fundamentales para operar: Barra Principal, Barra de Dibujo, Barra de Partes Animadas y Barra de Partes Estáticas.

La Barra Principal (*Main*, en inglés) sirve para crear, abrir, salvar, remover, tomar copias e insertar documentos. Además se puede cancelar y restaurar la ultima acción, compilar el documento, acceder a lo descargado-enviado de la ventana del servidor y a la caja de diálogo de encontrar variables, y búsquedas de documentos de todas las variables y mostrar los resultados en la ventana de salida.

La Barra de Dibujo (*Drawing*, en inglés) permite ejecutar varias operaciones en objetos seleccionados en el espacio de trabajo (*Wokspace*): Alinear a la izquierda, derecha, arriba, abajo así como agrupar y desagrupar objetos, colocar objetos al frente o detrás e intercambiar objetos.

La Barra de Partes Animadas (*Animated Parts*, en inglés) ofrece las opciones de insertar objetos en el espacio de trabajo (textos, botones, rectángulos, elipses, imágenes, operaciones básicas, pre registros, arcos, líneas).

La Barra de Partes Estáticas (*Static Parts*, en inglés) permite la selección y manipulación de objetos existentes. Cuando se activa no se pueden insertar nuevos objetos en el documento. Puede insertar una imagen en el fondo del croquis de un documento animado.

II.2.3 Crear documentos animados

Estos se almacenan en ficheros de extension “.HAD”. Los proyectos del Screen Builder se vinculan a los proyectos del Workbench. Las pantallas se construyen con partes animadas y estáticas: Una figura de fondo y una imagen dinámica; y editando modos se pueden manipular las diferentes partes de la pantalla. La figura de fondo es la parte gráfica estática mientras que la imagen dinámica es la parte gráfica animada, y esta siempre va delante de la estática. También se pueden ver las pantallas en tecnología Java utilizando direcciones IP de los nodos donde se descarga el documento compilado. Otra opción es la de crear una pantalla por defecto que automáticamente abra cuando se entre solo la dirección IP del nodo, a la cual se puede llamar “default.html”.

Las tareas de dirección de un documento animado que se pueden modificar son las siguientes:

1. Poner propiedades al proyecto.
2. Cambiar modos de edición.
3. Crear un *background* (fondo).
4. Insertar objetos animados.
5. Revisar objetos.
6. Definir efectos de animación para objetos.
7. Seleccionar variables.
8. Personalizar la plantilla del HTML.
9. Usar Selectores MultivARIABLES.
10. Usar la Biblioteca de Objetos Gráficos.

Cuando se le ponen propiedades a un proyecto o se le modifican es necesario realizar algunos cambios en algunas tareas: Unir pantallas (documentos animados) de proyectos del Workbench, definir contraseña de usuario y de administrador, definir servidores de datos y modificar las conexiones al mismo.

Existen tres modos de edición y se debe escoger uno de ellos. El primero es el llamado Diseño de Fondo, que permite la manipulación de las partes estáticas del documento. El segundo, Diseño Animado, permite la manipulación de las partes animadas; y el tercero se conoce como modo Libre y permite la manipulación tanto de la parte estática como dinámica del documento. Cuando se trabaja en modo de Diseño de Fondo la barra de herramientas del modo de Diseño Animado no se encuentra disponible y viceversa.

Para crear un fondo o *background*, existe la posibilidad de colocar imágenes como partes estáticas del Diseño de Fondo, y es necesario trabajar en dicho modo o en el modo Libre. El modo de Diseño de Fondo se muestra solo en los artículos que constituyen el fondo del documento.

En la tarea de insertar objetos animados se debe trabajar en modo de Diseño de Fondo o en el Libre. Estos objetos pueden ser de una gran variedad, desde imágenes y entradas de texto hasta correos electrónicos y botones.

En el caso de revisar objetos para su edición existen variantes como agruparlos y moverlos, y además seleccionarlos, cortarlos, pegarlos y borrarlos. Los efectos de animación para objetos que están permitidos definir tienen que ver con el aspecto que este tenga y la acción que se vaya a realizar: Mover, rotar, cambiar el tamaño o la apariencia.

En la selección de variables, las propiedades dinámicas de los objetos usan constantes o variables para definir su comportamiento en el tiempo de ejecución. Las variables pueden ser escogidas de cuatro orígenes: Datos del PLC, selectores, variables internas o variables del sistema. También se permite mapear variables de cualquier tipo de dato o parámetro.

Para personalizar la plantilla del HTML es necesario conocer primero que dicha plantilla define el tamaño de la pantalla mostrada en el recipiente (página HTML) así como la vista del contenedor de muestra. La plantilla predefinida pone los recipientes al mismo tamaño como las pantallas que ellos visualizan. El tamaño de las pantallas se puede modificar y además se puede usar una plantilla diferente para un mismo recipiente.

Un Selector Multivariable consiste en una configuración de una lista de etiquetas y una lista de variables que se pueden asociar. Usando el selector se consigue una configuración particular, existe la posibilidad de trabajar con una Web genérica y las variables básicas y los objetos permanecen inalterables, solo cambia su valor.

En el uso de la Biblioteca de Objetos Gráficos, los objetos individuales o grupos de objetos pueden ser reutilizados entre los documentos enviando los objetos a la biblioteca. La ventana de la biblioteca se divide en Buscador y Área de Chequeo, el Buscador muestra las carpetas y ficheros contenidos en la biblioteca y el Área de Chequeo muestra el contenido de los ficheros seleccionados en el Buscador.

II.2.4 Visualizar Pantallas

Desde cualquier plataforma en tecnología Java (habilitada por el navegador) se pueden visualizar pantallas, por ejemplo Internet Explorer o Netscape. Entrando a la dirección IP del nodo donde se encuentra localizada la pantalla y su nombre (HTML) en el buscador se visualiza. Además existe la posibilidad de crear una pantalla por defecto (predefinida) que automáticamente abra cuando sólo sea entrada la dirección IP del nodo en el buscador.

Para visualizar una pantalla usando Internet Explorer es necesario que la computadora en la que se trabaja tenga conexión TCP/IP al nodo donde opera el servidor de datos y la pantalla de descarga.

II.2.5 Víncular pantallas

Existe la posibilidad de construir una aplicación multipágina a través de la vinculación de pantallas pertenecientes a un mismo contenedor (página HTML). Para crear tal uso, en un contenedor se construyen las pantallas individuales y entonces se conectan juntas usando partes animadas especiales. El ambiente del Screen Builder le permite editar varias pantallas al mismo tiempo en el espacio de trabajo.

II.2.6 Compilar y cargar documentos

Es necesario compilar los documentos animados en páginas HTML luego de enviarlos al servidor Web donde pueden ser visualizados en un buscador, conectados al servidor de datos suministrando datos en tiempo real. El compilador realiza tres tareas: Chequear todas las referencias de los datos del PLC, chequear los vínculos con otros documentos, construir una página HTML, cuando no sean detectados errores en el documento.

Consideraciones finales

De esta manera se da por concluido el recorrido por los componentes Workbench y el HiBeam pertenecientes a la versión ISaGRAF 5.0, donde se realizó una amplia descripción de como trabajar con los mismos a partir de las opciones que ofrecen y poder llevar a cabo la realización de proyectos (desarrollo de aplicaciones). Se hizo una explicación de los elementos más importantes a detallar en ambos casos y que son imprescindibles conocer : Como configurar un proyecto, con que lenguajes trabajar, como seleccionar variables, funciones y modos de operación. Es válido destacar que aunque se ha hecho mención a todos los lenguajes de control nos apoyaremos en el lenguaje de contacto (LD) para la ejecución de proyectos, lenguaje este gráfico como se mencionó anteriormente, que se utiliza solo para operaciones booleanas y que no ofrece una gran complejidad (no se utiliza para la programación de alto nivel como el SFC o el FC) , de modo que se pueda hacer más asequible el trabajo. Para que se puedan conectar correctamente el Workbench y el Hi Beam es necesario que se haya configurado correctamente la aplicación hecha en el banco de trabajo de tal forma que cuando sea llamada por el HiBeam no existan problemas y el sistema indique algún tipo de error.El éxito de su trabajo estará fijado por realizar los pasos correctamente navegando a través de estos dos importantes componentes del ISaGRAF.

CAPÍTULO III. DESARROLLO DE APLICACIONES

Introducción

Después de conocer detalladamente al paquete de software ISaGRAF mediante un estudio de sus principales características y funciones, y de reseñar sobre su versión mas reciente ISaGRAF 5.0, del que se realizó un profundo análisis de su Workbench y HiBeam como componentes fundamentales que son dentro de su funcionamiento, es el momento de llevar a la práctica todos esos elementos teóricos mediante el montaje de aplicaciones utilizando el Workbench, el HiBeam y la Máquina Virtual unido al lenguaje de control industrial LD para la programación gráfica. El objetivo de describir y crear estas aplicaciones es que faciliten el aprendizaje del modelo de trabajo del software y que funcionen como proyectos tipo para la docencia en un futuro, como casos de estudio. Se comenzará por una aplicación sencilla y se irá ganando en complejidad tratando de que se abarquen la mayor cantidad de variantes posibles y que logren demostrar la gran cantidad de servicios que es capaz de prestar el ISaGRAF y porque es tan reconocido a nivel internacional como un producto de excelencia.

III.1 Pasos comunes en la programación en el Workbench

Para realizar el montaje de una aplicación cualquiera es preciso que se ejecuten una serie de pasos que son comunes para cualquier proyecto, tanto en el Workbench como en el Hi Beam. En este epígrafe se hará referencia a los del Workbench, ya que es en este por donde se comienza a trabajar en la creación de una aplicación. Los pasos comunes a llevar a cabo en la programación del HiBeam serán expuestos más adelante según se vaya avanzando en la construcción de los programas.

El primer paso a ejecutar es abrir la Máquina Virtual (NT Target), tarea esta imprescindible en la operación, y que se encuentra en el Menú de Inicio, posteriormente se abre el Workbench, que también se encuentra en el Menú de Inicio de *Windows*.

Al abrir el Workbench este aparecerá activado por defecto con la Arquitectura de Vínculos, dentro de la ventana de recursos que aparece, en la opción de programa accionando click derecho aparecen los tipos de lenguajes(se escoge el LD, con el cual se desarrollarán todas

las aplicaciones, ver Anexo 3). Posteriormente se crea el programa y se le asigna un nombre y una localización donde será guardado para su posterior utilización.

Al ejecutar el programa creado (ejecutando un doble click sobre el nombre) se abre una ventana nueva conocida como Banco de Programación (o de Trabajo) donde se realizará la programación gráfica con el lenguaje LD.



Figura III.1 Ambiente de la Máquina Virtual

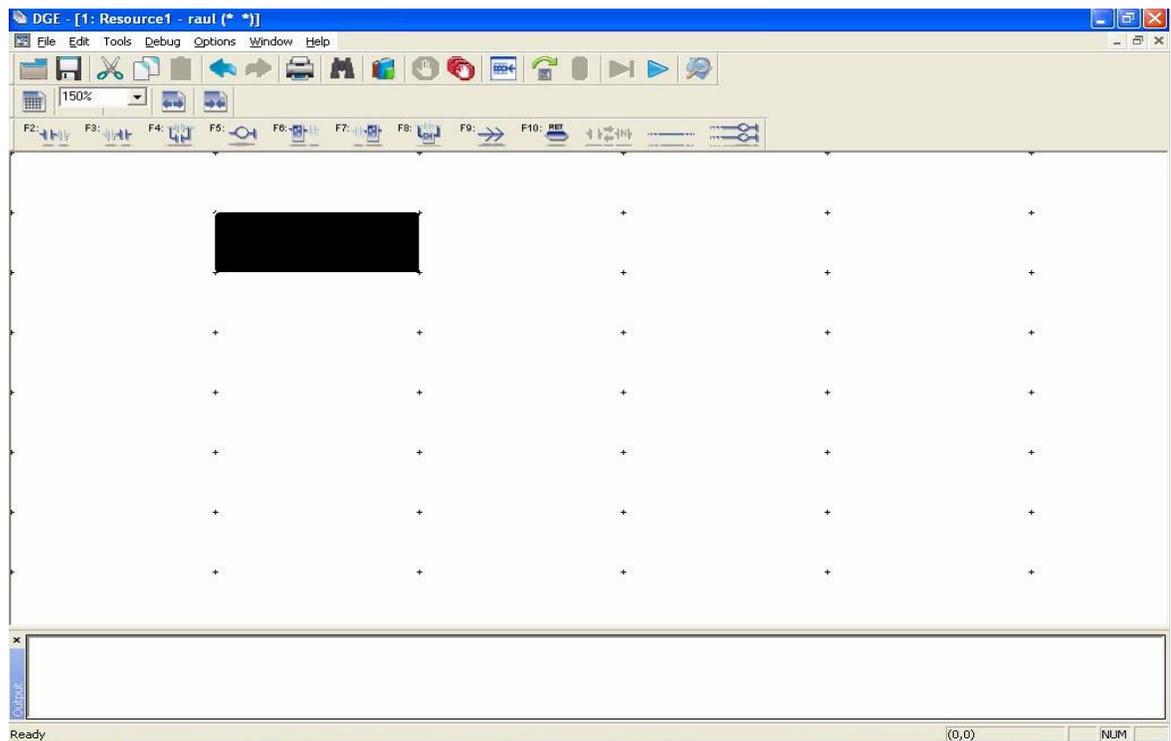


Figura III.2 Banco de Programación

Este banco ofrece una diversa gama de opciones para la programación que van desde funciones hasta bloques de funciones, ambas con la posibilidad de incluir variables en su contenido. En el caso del lenguaje LD las funciones y variables van unidos a los contactos y bobinas combinadas con ecuaciones booleanas. Las señales utilizadas pueden ser digitales y analógicas según convenga.

III.2 Primera Aplicación :El Interruptor

La primera aplicación consiste en montar un interruptor, que puede ser utilizado para encender una lámpara o un motor y que opera con una señal digital. Se conectará a una lámpara en este caso para hacer el proceso más sencillo y claro.

Una vez realizados los pasos comunes en el Workbench y abierta la Máquina Virtual se comienzan a realizar los pasos específicos de cada aplicación, según sus características y objetivos, incluyendo los del HiBeam además.

Una vez abierto el Banco de Programación se comienzan a escoger los componentes gráficos que formarán parte del programa. Primero se selecciona un Contacto por la Derecha marcando con un click sobre el icono del mismo(u oprimiendo la tecla F3) y se llama a la variable de dicho contacto “ent1”, ya que será la variable de entrada del sistema. Automáticamente a este contacto le aparece un Contacto por la Izquierda y a cuya variable se define como “sal1” por fungir como salida del sistema. Ambas variables son booleanas y se definen sus características ejecutando un doble click sobre el contacto respectivo.

Si en vez de seleccionar primero un Contacto por la Derecha se selecciona un Contacto por la Izquierda(con un click sobre el icono u oprimiendo F2), automáticamente aparece un Contacto por la Derecha conectado al mismo y se repite el mismo procedimiento para las variables al que se hizo referencia anteriormente.

Al finalizar la ejecución del procedimiento descrito se da por concluido el trabajo con el Banco de Programación y con el Workbench para esta aplicación.El próximo paso es realizar la parte de la programación en el HiBeam a través del Screen Builder (Constructor de Pantallas, en español), así que lo primero es abrir el HiBeam, que se encuentra también en el Menú de Inicio de *Windows* con el nombre de HiBeam Screen Builder.

La Figura III.3 indica como queda representado el recurso del interruptor en el Banco de Programación:

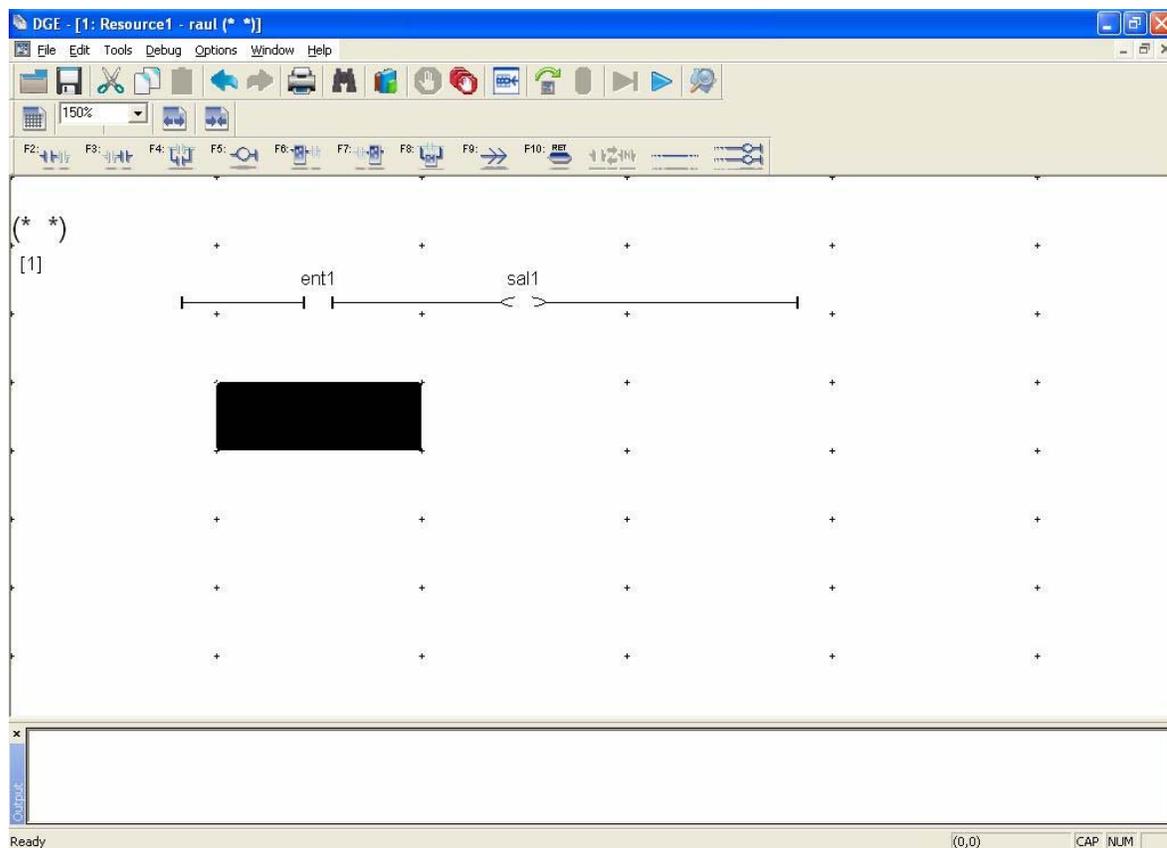


Figura III.3 Recurso del Interruptor en el Banco de Programación

En el Screen Builder del HiBeam se crea un nuevo documento y se le asigna una dirección donde será salvado. De la Barra de Partes Animadas, situada a la izquierda de la pantalla, se selecciona un botón (que asume la función de un interruptor) y es colocado en el Workspace (Espacio de Trabajo).

Posteriormente en la Barra Principal se accede a la Biblioteca, abriendo la carpeta que se encuentra dentro de ella, y de la misma se escoge el tipo de lámpara que se utilizará, para este caso se selecciona la opción “*Lamp1.Had*” que representa a las lámparas de 2 estados, y de los diferentes tipos de diseños se escoge uno y se copia para el Espacio de Trabajo. Estas lámparas son capaces de cambiar de color cuando se acciona el interruptor y su tamaño al igual que el del botón se pueden modificar.

La configuración del botón y la lámpara es la siguiente acción a realizar, y que requiere se haga correctamente para evitar que el sistema señale algún tipo de error y puedan conectarse sin problemas.

Para configurar cualquier elemento lo primero es ejecutar un doble click sobre el mismo y en la ventana que aparece modificar las opciones que sean necesarias.

Al ejecutar doble click sobre el botón aparece una ventana con varias opciones, la primera es para el estilo (*Styles*, en inglés) y donde solo se modifica la variante *Text* (texto, en español) colocándole el valor “ent1”. En la opción Variables se coloca el valor “ent1” (es la misma variable que se utilizó en el Banco de Programación para la entrada) de origen “*Internal*” y booleana. Para la casilla Action (Acción, en español) ya aparece predefinido el estilo de acción que consiste en variar el estado según se opere el *mouse*, se selecciona ejecutando un doble click sobre “*pushed/released values*” (valores oprimido/liberado, en español) o sea cambiar el valor con solo dar un click. La opción Mover no será utilizada en esta aplicación y la casilla de Aspecto solo si se quiere modificar el color del botón. Así queda configurado el botón, se asoció a la variable “ent1” porque como ya se dijo este funcionará como interruptor.

Para configurar la lámpara el proceso es sencillo, solo se modifica la opción Variables donde se coloca el valor “sal1” (la misma variable de salida del Banco de Programación) de origen “*Internal*” y booleana.

El sistema se encuentra listo para funcionar y solo falta configurar el proyecto en general para que el HiBeam se conecte al Workbench y pueda operar con el programa y las variables creadas en el Banco de Programación. En la Figura III.4 se muestra como queda confeccionado el interruptor en el espacio de trabajo del Screen Builder.

III.2.1 Pasos comunes en la programación en el HiBeam

La configuración del proyecto en el HiBeam requiere una serie de procedimientos que son comunes para todas las aplicaciones, el primero, cuando ya se encuentran configurados todos los componentes, es ir a la Barra Principal y en la opción “*Project*” escoger la variante “*Settings*”, donde se configura el proyecto, la contraseña, el servidor de datos y la conexión. El proyecto es de tipo ISaGRAF PRO y al *Path* (trayectoria, en español) se entra con un click y se le indica el siguiente camino a seguir:

ISaGRAF 5.0/prj/nombre del programa en el Workbench.

Posteriormente se va a la biblioteca ISaGRAF 5.0 del proyecto en la base de datos y se abre, automáticamente aparecerá el vínculo con el Workbench.

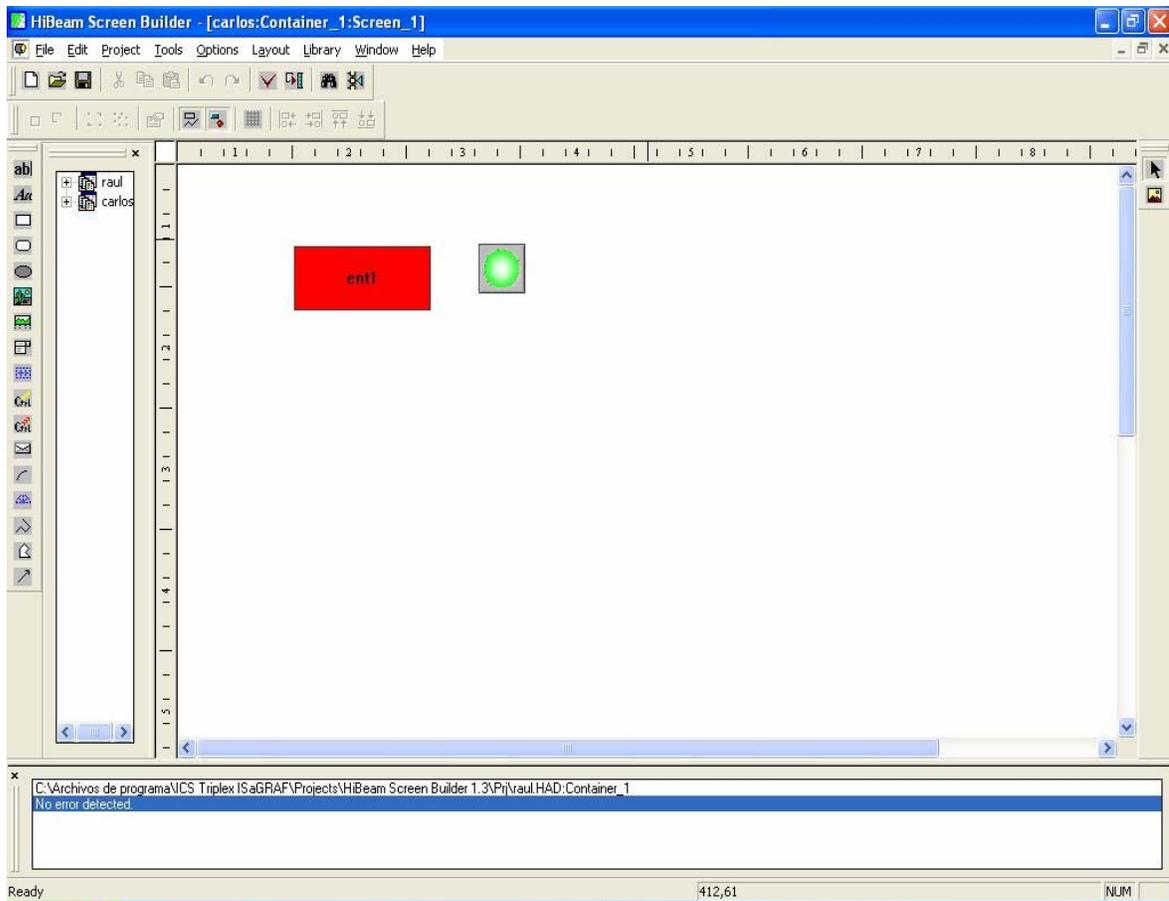


Figura III.4 El Interruptor en el HiBeam Screen Builder

En el servidor de datos aparece una dirección IP que depende de la computadora en que se esté trabajando, cuando es en la misma computadora se le configura con 127.0.0.1 para que opere correctamente.

Se compila el programa en el HiBeam a través de la opción “*Compile*” de la Barra Principal, y si no tiene ningún error se descargará para la Máquina Virtual. Al marcar el botón aparece una ventana con opciones para la descarga, donde se entra en “*Options*” a través de un click y aparece otra ventana donde hay que seleccionar una serie de datos de los que aparecen en la misma, tal como indica la Figura III.5.

En Descargar se marca:

- Enviar clases de Java
- Enviar todas las imágenes
- Enviar ficheros Web Extra

En FTP se escribe en Nombre de Usuario: virgo2000.

Al terminar con esta ventana se regresa a la anterior donde se escribe la contraseña v2000 tal y como muestra la Figura III.6, y se carga en la Máquina Virtual mediante la opción “Load”, luego se cierra la ventana.

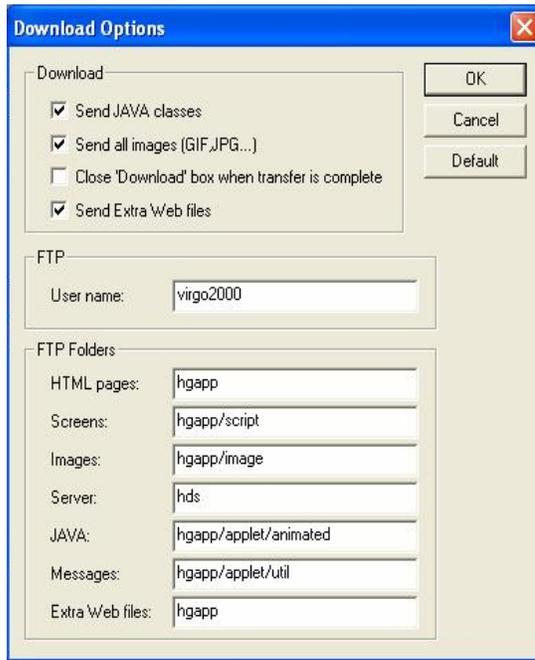


Figura III.5 Ventana de datos

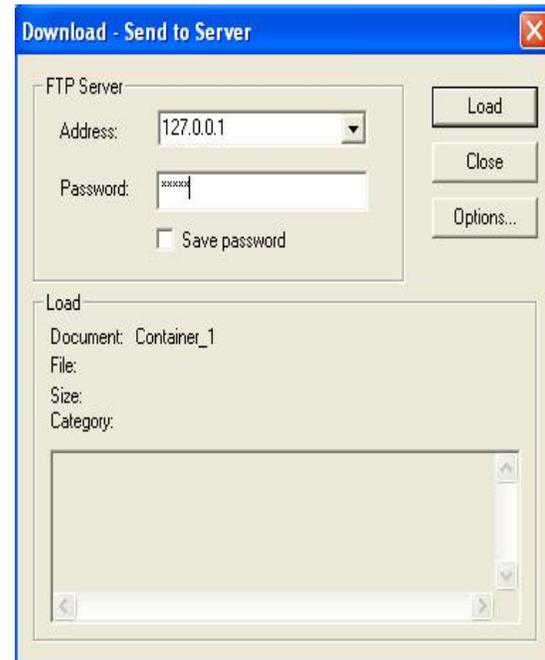


Figura III.6 Ventana para la descarga

Al realizar esta operación se genera la página Web donde aparecerá la animación de lo que se programó en el HiBeam. Para verla se abre una ventana de Microsoft Internet Explorer y en el Menú Principal en Archivo/Abrir, se marca “Examinar” y se selecciona la página Web container_1.html para abrirla(ver Anexo 4). En el caso de que cuando se examine y no aparezca el contenedor en la ventana hay que ir a C:\Archivos de programa\ICS Triplex ISaGRAF\Run Time\Windows NT\App\hgapp, y en esa localización se encuentra.

En la página Web aparecerá el documento animado, operable a través del mouse y listo para mostrar su funcionamiento.

Para el ejemplo del interruptor, a través de un click sobre el botón, este encenderá y apagará la lámpara alternativamente. Después de llevar a cabo todos estos pasos comunes que se realizan en la programación del HiBeam y la configuración del proyecto, el interruptor se verá en la página Web como indica la Figura III.7.

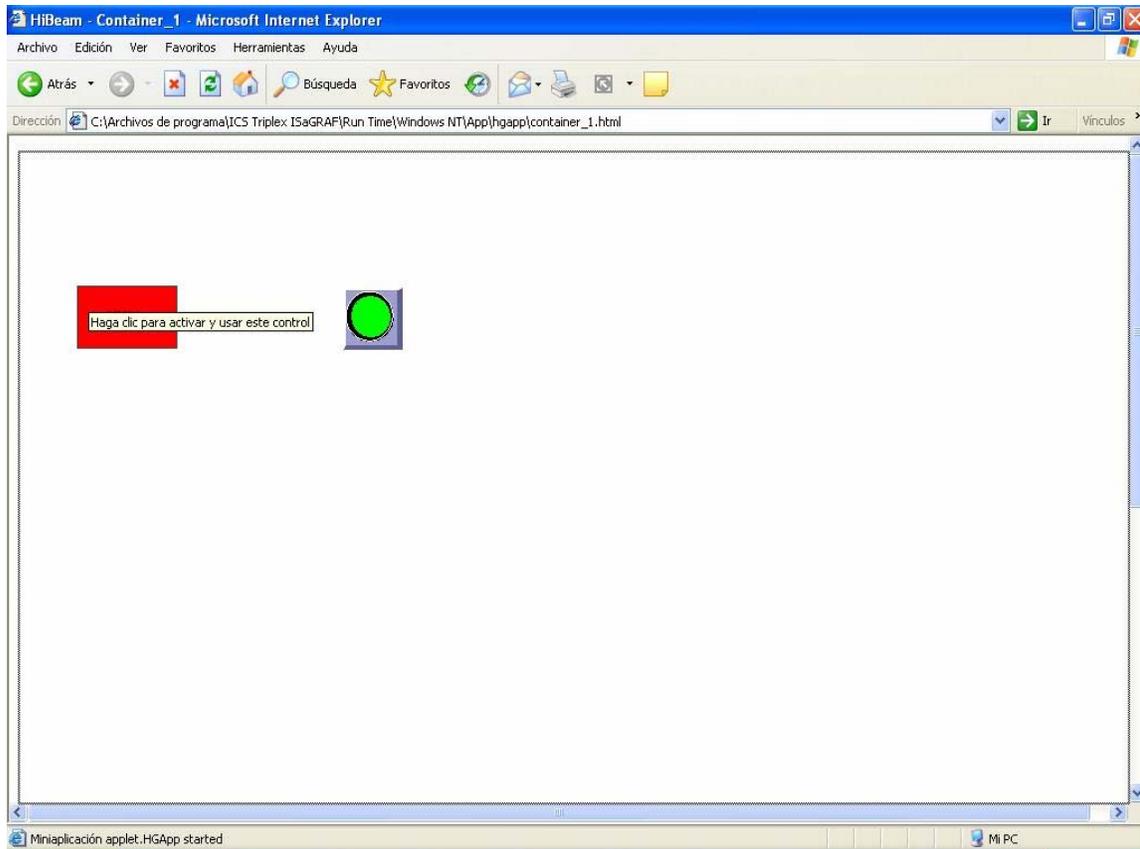


Figura III.7 El Interruptor visto en la página Web

III.3 Segunda Aplicación: Trabajo con señales analógicas y digitales

La segunda aplicación consiste en montar un sistema que sea capaz de trabajar con señales analógicas y digitales a la vez, a través de indicadores y potenciómetros. Realizar esta aplicación requiere una mayor complejidad en el trabajo que con el interruptor debido al trabajo con dos tipos de señales diferentes.

Lo primero es llevar a cabo los pasos comunes del Workbench descritos anteriormente, con la Máquina Virtual incluida, hasta abrir el Banco de Programación para comenzar a construir el recurso.

Como serán utilizadas señales digitales, para este caso la programación de la parte digital es igual a la de la primera aplicación (el interruptor) que trabaja con una señal de ese tipo. Por lo tanto se coloca un Contacto por la Derecha, al que se le asigna como variable “ent1”, y un Contacto por la Izquierda, que lleva como variable “sal1”.

Para las señales analógicas el procedimiento es un poco más complejo ya que requiere más entradas y salidas unidas a los bloques. Con el lenguaje de control LD se hace necesaria la utilización de cuatro pasos gráficos para el proyecto completo, el primero se refiere a la parte de las señales digitales y es el que se acaba de describir.

El segundo paso, para las señales analógicas, consiste en colocar un Bloque por la Izquierda (click sobre el icono o pulsando F6), se escoge uno de dos entradas que funcione como sumador, a continuación del bloque el sistema coloca automáticamente otra salida (un Contacto por la Izquierda) aparte de la salida del bloque. Las variables de entrada del bloque son “ent2” y “ent3” y la salida del mismo “sal2”, la variable de la salida que agrega el sistema será “sal3”. Los dos primeros pasos quedan mostrados en la Figura III.8.

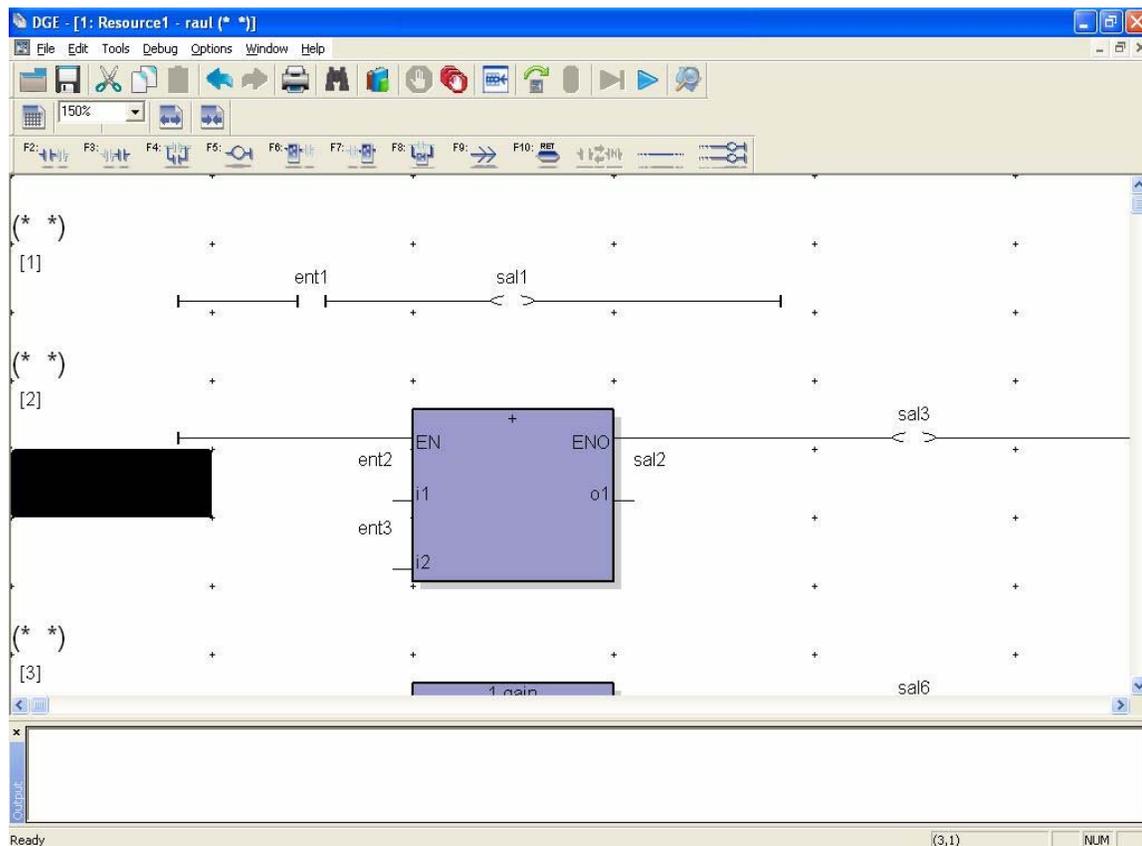


Figura III.8 Los dos primeros pasos gráficos del proyecto

A continuación para el tercer paso se coloca un Bloque por la Derecha (o por la izquierda según se quiera) que sea de tipo “Ganancia 1”, donde la salida sea igual a la entrada. La

variable de entrada “ent2” y la de salida “sal4”, automáticamente el sistema añade la variable “sal6”(un Contacto por la Izquierda). Este tipo de bloque se utiliza para separar las señales analógicas y digitales que van al indicador.

El cuarto paso es similar al tercero, se coloca un bloque igual y solo se diferencian en las variables, la de entrada será “ent3” y la de salida del bloque “sal5”, la salida que aparece por defecto será “sal7”. El tercer y cuarto paso quedarán tal y como indica la Figura III.9.

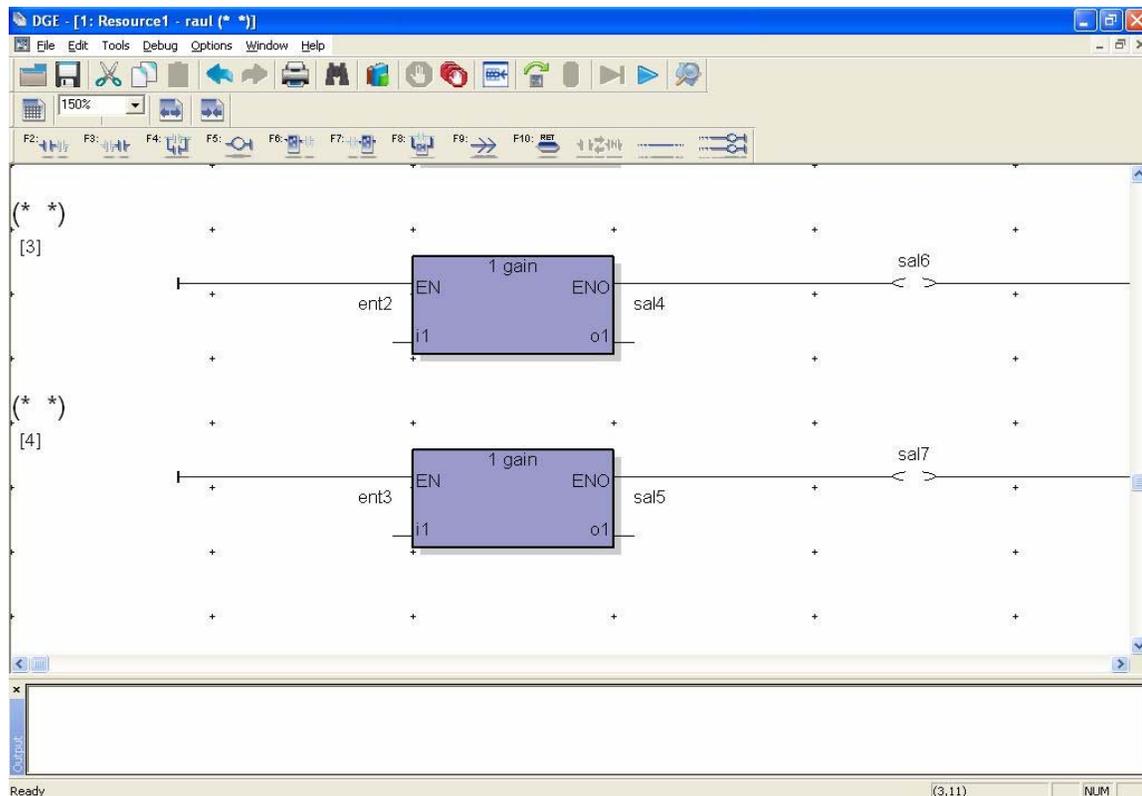


Figura III.9 El tercer y cuarto paso gráfico del proyecto

De esta manera se da por concluida la operación en el Banco de Programación del Workbench para esta aplicación y se comienza la parte correspondiente al HiBeam.

En el Screen Builder el montaje y configuración para el trabajo con las señales digitales es igual a la de la primera aplicación(seleccionar un botón, una lámpara y configurarlos). Para las señales analógicas aumenta la complejidad, lo primero es buscar en el Menú Principal la Biblioteca y abrirla para seleccionar los componentes. Se escogen dos indicadores tipo “*Display.Had*”, un indicador lineal “*Linear indicator.Had*” y dos potenciómetros

“Potentionmeter.Had” para el espacio de trabajo.Cada componente debe ser configurado para su correcta conexión y funcionamiento.

Al indicador linear se le asigna la variable “sal2”(que proviene del recurso hecho en el Banco de Programación al igual que el resto de las variables que se utilicen) en Variables, en Estilos se define el rango de la escala del indicador en 0-200 como valores mínimo y máximo. El resto de las opciones no se modifica.

A uno de los potenciómetros se le asigna la variable “ent2”, y como valores por la izquierda 0 y por la derecha 100 en la casilla Estilos. Al otro potenciómetro se le colocan los mismos valores por la izquierda y la derecha respectivamente, y como variable de entrada se coloca “ent3”.

El primer indicador se nombra “ent2” y se le asignará la variable de operación “sal4”, y al segundo indicador se nombra “ent3” y su variable designada “sal5”. El resto de los aspectos no se varía en la configuración. El sistema quedará tal y como muestra la Figura III.10.

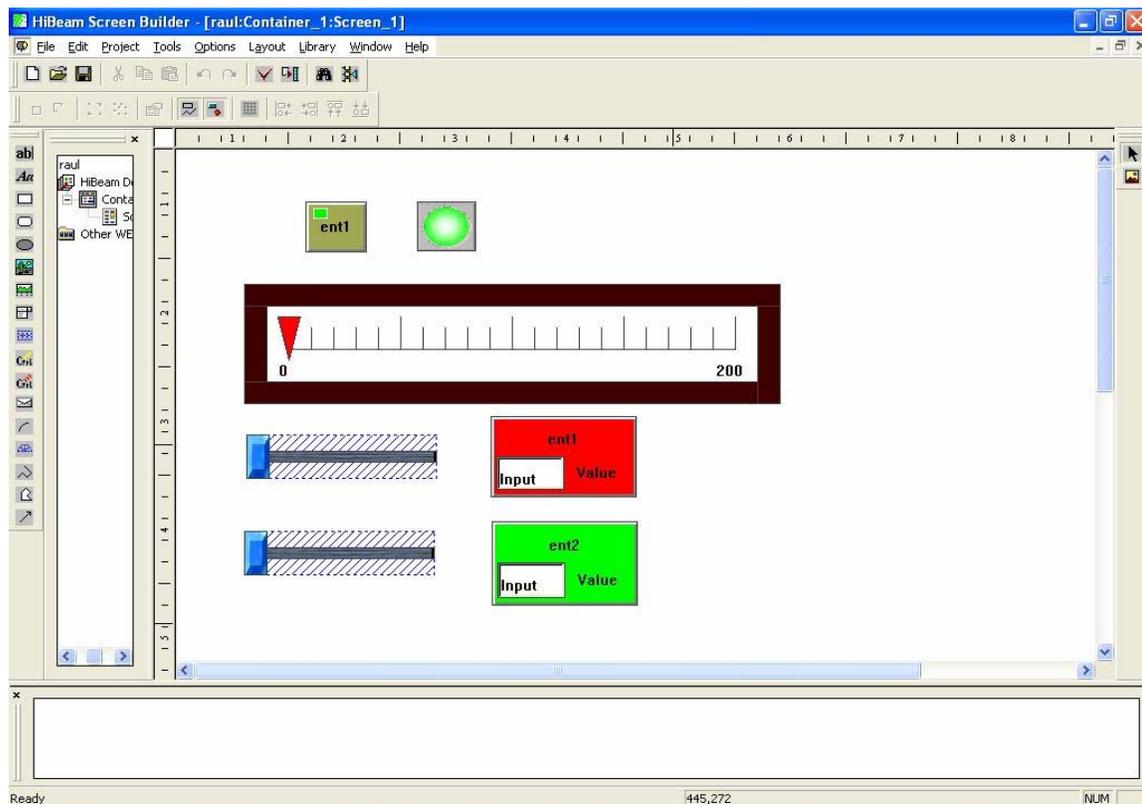


Figura III.10 Vista de la aplicación en el Screen Builder

Luego se configura el proyecto en su totalidad, en la opción “*Project*” de la barra principal, llevando a cabo los pasos comunes de la configuración, compilación y descarga de proyectos en el HiBeam. Los pasos hasta que aparezca el documento animado en la página Web son los mismos para todas las aplicaciones como ya se explicó, y se podrá operar el mismo a través del *mouse* cuando este aparezca en pantalla. El resultado final en la página de Microsoft Internet Explorer es el que se muestra en la Figura III.11.

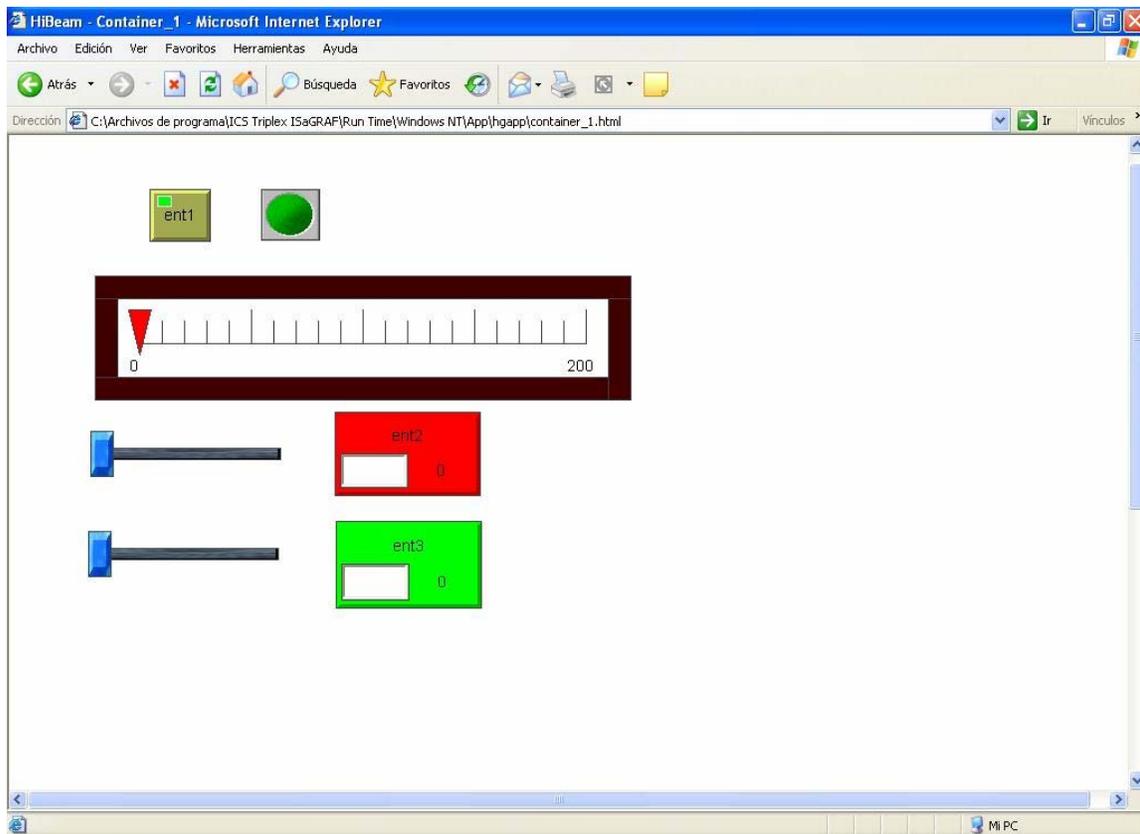


Figura III.11 Ambiente de la aplicación en la página web

Al mover con el *mouse* la barra de uno de los potenciómetros provocará el corrimiento de la aguja señalizadora del indicador lineal y cambiará el valor de los indicadores..

III.4 Tercera Aplicación: Control en Red de la Máquina Virtual

La tercera aplicación consiste en operar un programa (proyecto) desde una Máquina Virtual remota(colocada en otra PC). Para este caso se puede utilizar la primera o segunda

aplicación, ya elaboradas, con el objetivo de demostrar su funcionamiento a través de la red. Es necesario e imprescindible que el programa haya sido creado previamente utilizando el Workbench y el HiBeam, que en la PC remota la Máquina Virtual se encuentre abierta, para recibir la descarga, y además se debe tener conocimiento de su dirección IP. La conexión es de tipo maestro-esclavo, tal como muestra el diagrama de la Figura III.12 :

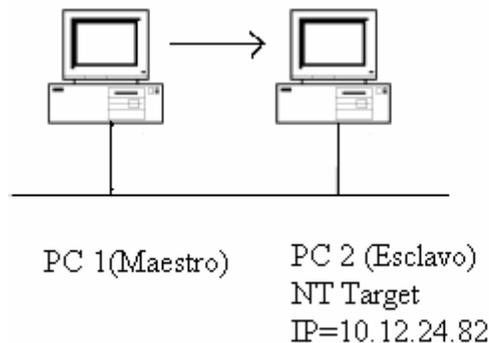


Figura III.12 Esquema de la conexión en red

El primer paso es abrir el Workbench y localizar el proyecto en la Arquitectura de Vínculos, posteriormente se accede a la Arquitectura de Hardware para configurar la conexión con la dirección IP de la computadora donde se va a descargar el programa confeccionado.

Dentro de la Arquitectura de Hardware, en las propiedades de la conexión, aparece una ventana con los parámetros de la red, donde se encuentra por defecto el valor de la dirección IP de la PC en que se está trabajando, dicho valor debe ser modificado por el de la PC remota. Dicha ventana es tal y como ilustra la Figura III.13.

A continuación se procede a compilar el proyecto, para verificar que no contenga errores, y a su descarga, utilizando los botones de la Barra de Herramientas diseñados para estas operaciones básicas.

En la descarga aparece una ventana donde se selecciona que recurso hecho en el Banco de Programación es el que se desea descargar para la Máquina Virtual remota (ver Figura III.14). Se debe marcar el nombre y accionar un click sobre “*Download*” para completar el procedimiento.

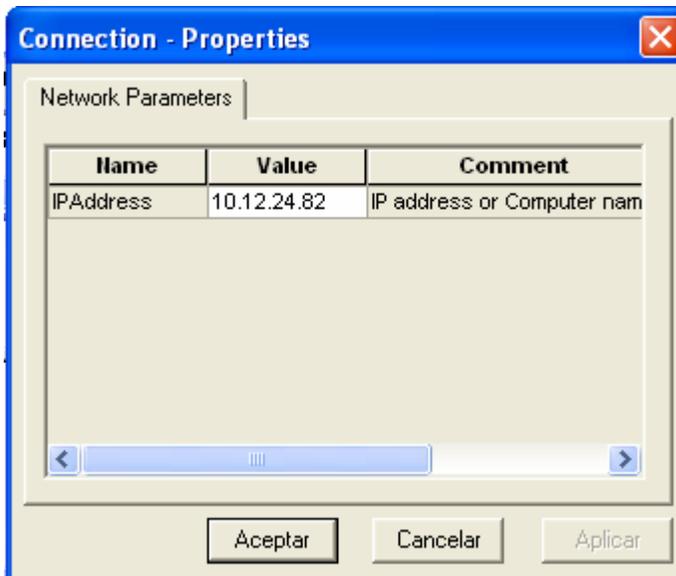


Figura III.13 Ventana de Conexión de Parámetros de Red

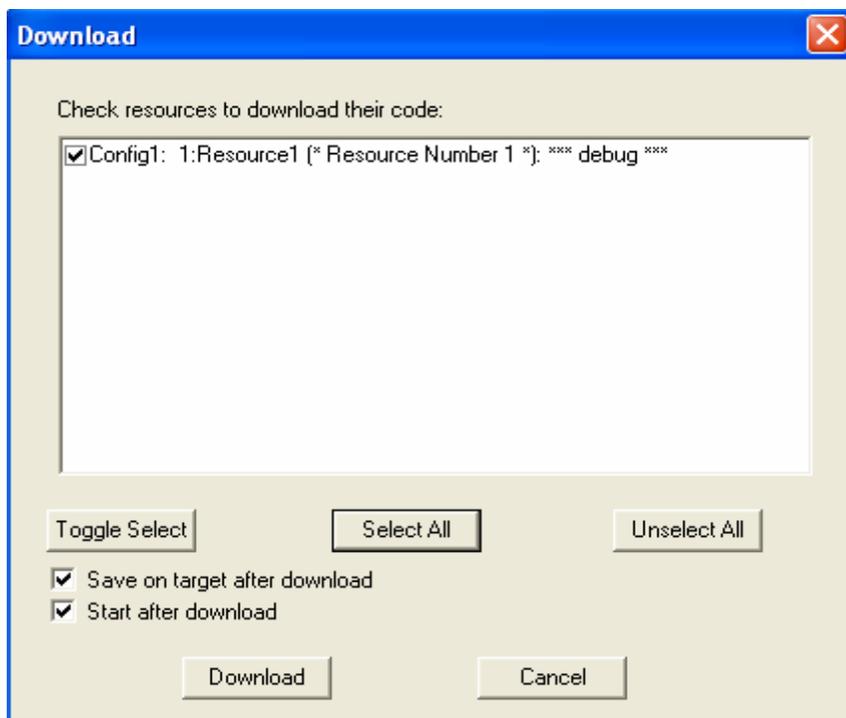


Figura III.14 Ventana para la Descarga en la Máquina Virtual remota

De esta manera el proyecto es descargado para la Máquina Virtual de la computadora conectada en red y se encuentra listo para ser ejecutado desde la misma. El Hi Beam no es necesario abrirlo, ya que al construir la aplicación el sistema crea el vínculo entre este y el

Workbench (como se explicó para la primera y segunda aplicación), por lo tanto se puede acceder a ver el ambiente del programa en la página Web directamente después de realizado el procedimiento de la descarga.

Para acceder a la página Web operando en red el procedimiento es el mismo que cuando se trabaja de forma local, se abre una ventana de Microsoft Internet Explorer y se busca el contenedor, automáticamente desde la Máquina Virtual se descarga el programa. En el caso del interruptor se podrá encender y apagar del mismo modo que se hace desde la propia computadora, y para el ejemplo de las señales analógicas y digitales también se opera de la misma manera.

Además mediante este funcionamiento en red se permite tomar datos de un PLC virtual o descargar a otras Máquinas Virtuales, de forma tal que optimicen el trabajo y garanticen una mejor eficiencia en la operación.

Consideraciones finales

Con el desarrollo y descripción de las aplicaciones se pone fin al trabajo con los componentes del ISaGRAF 5.0 y la puesta en práctica de los conocimientos adquiridos sobre el mismo. La detallada descripción de todo el procedimiento de construcción de los programas se hizo con el objetivo de que funcionen como proyectos para la docencia y puedan ayudar al aprendizaje de la operación con este software. Se trataron algunas variantes de forma que mostraran diversidad y flexibilidad en la programación, desde la elaboración de una aplicación para un proceso simple hasta el trabajo en red. Es importante reseñar la ejecución de los pasos comunes de cada componente en la programación y su correcta configuración, algo habitual en cada uno de los programas que se realicen. Todas las aplicaciones mostraron resultados satisfactorios en su compilación y descarga, muestra del rendimiento que es capaz de ofrecer esta versión del ISaGRAF, el software de la automatización por excelencia.

CONCLUSIONES

A partir de la revisión bibliográfica y estudio del ISaGRAF, donde nos detuvimos en su versión ISaGRAF 5.0 mediante un análisis de sus componentes Workbench y HiBeam, se construyeron y describieron una serie de aplicaciones utilizando dichos componentes con el objetivo de fungir como proyectos tipo para la docencia y mostrar el modelo de funcionamiento del software en cuestión. El desarrollo de las aplicaciones mencionadas posibilitó el cumplimiento de los objetivos propuestos en el presente trabajo de diploma, y al concluir el mismo se han llegado a las siguientes conclusiones:

1. El ISaGRAF ofrece gran cantidad de opciones y ventajas en el trabajo con los sistemas de control, específicamente su versión ISaGRAF 5.0 muestra una gran robustez al ser capaz de funcionar en varias plataformas y ser compatible con diferentes normas internacionales.
2. Los componentes Workbench y HiBeam son los elementos fundamentales donde se desarrolla un proyecto, y se hace necesario una correcta configuración y operación de los mismos de forma tal que permitan su posterior conexión y éxito en la simulación del programa realizado.
3. La Máquina Virtual juega un importante papel en la descarga y posterior representación en una página Web del proyecto realizado, una vez concluida la programación del mismo en el Workbench y el HiBeam.
4. Las aplicaciones creadas muestran variantes en el trabajo con el ISaGRAF y los resultados obtenidos en la simulación comprueban su buen funcionamiento.
5. La descripción que se hizo en la construcción de las aplicaciones facilita el aprendizaje en la operación con el ISaGRAF 5.0. Todos los proyectos desarrollados se encuentran a disposición del sistema docente y de quien los necesite para ser utilizados como objetos de estudio.

Como conclusión final reafirmamos que los objetivos propuestos en este trabajo fueron cumplidos satisfactoriamente.

RECOMENDACIONES

Con el objetivo de profundizar en el estudio y aprendizaje del ISaGRAF recomendamos los siguientes aspectos:

1. Insertar el software en el proceso docente de la carrera de Ingeniería Automática, como una opción para el estudio y operación de los sistemas de control.
2. Continuar con el análisis y conocimiento del software ISaGRAF con vista a dominar completamente su campo de operación.
3. Trabajar en el desarrollo de nuevas aplicaciones que requieran una mayor complejidad y que además puedan funcionar como proyectos tipo para la docencia.
4. Realizar pruebas reales con el ISaGRAF 5.0 en ambientes industriales con el objetivo de una futura incorporación del mismo a este medio en el país.

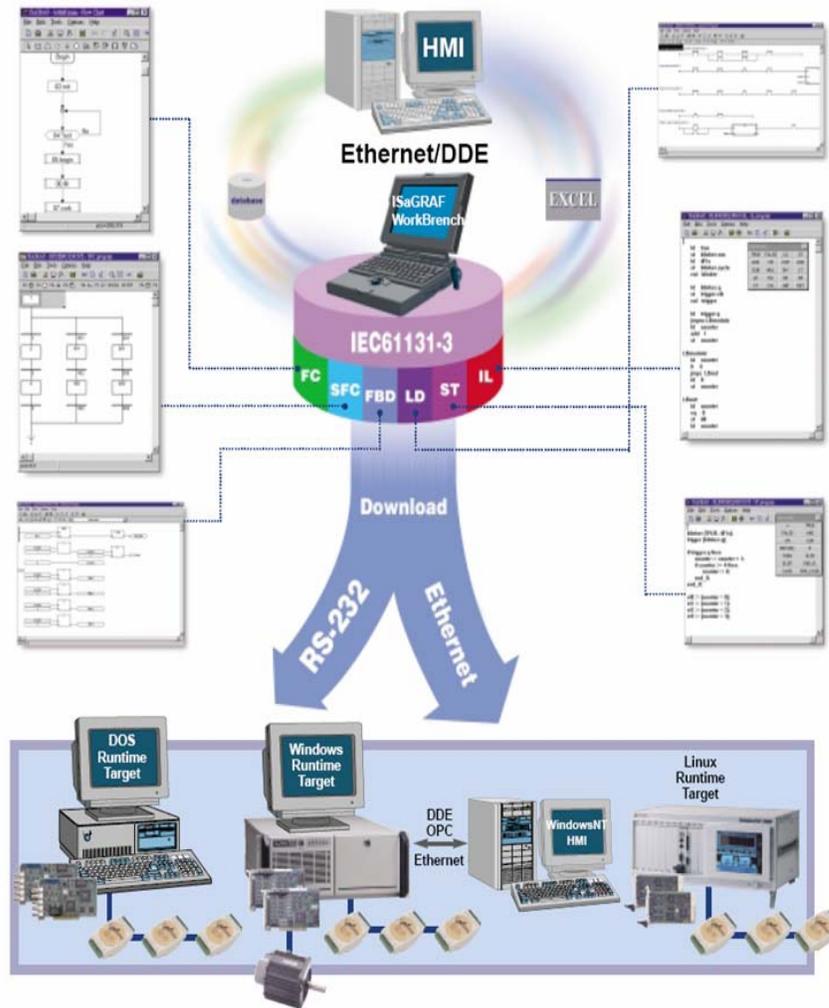
REFERENCIAS BIBLIOGRÁFICAS

- [1]. Engineeringtalk Editorial Team. (2005). ISaGRAF goes IEC 61499 compliant, disponible en: <http://www.engineeringtalk.com/news/ics/ics124.html> , accedido el 07/02 /2006.
- [2]. ICS Triplex ISaGRAF Inc.Using HiBeam Screens [pdf].Febrero, 2004.
- [3]. ICS Triplex ISaGRAF Inc.ISaGRAF General Specifications [pdf].Abril, 2005.
- [4]. ICS Triplex ISaGRAF Inc.HiBeam Screen Builder [pdf].Noviembre, 2005.
- [5]. ICS Triplex ISaGRAF Inc.ISaGRAF [pdf].Noviembre, 2005.
- [6]. ICS Triplex ISaGRAF Inc. (2006).HiBeam, disponible en: <http://www.isagraf.com/pages/products/HiBeam/screenbuilder.htm>, accedido el 15/03/2006.
- [7]. ICS Triplex ISaGRAF Inc. (2006).Control Products, disponible en: <http://www.isagraf.com/pages/products/products/home.htm> , accedido el 15/03/2006.
- [8]. ICS Triplex ISaGRAF Inc. (2006).Application Workbench, disponible en: <http://www.isagraf.com/pages/products/Isagraf/appworkbench.htm>, accedido el 17/03/2006.
- [9]. ICS Triplex ISaGRAF Inc. (2006).Runtime Target, disponible en: <http://www.isagraf.com/pages/products/Isagraf/runtime.htm> , accedido el 22/03/2006.
- [10]. ICS Triplex ISaGRAF Inc. (2006).Alarms and Events, disponible en: <http://www.isagraf.com/pages/products/Isagraf/alarms.htm> , accedido el 25/03/2006.

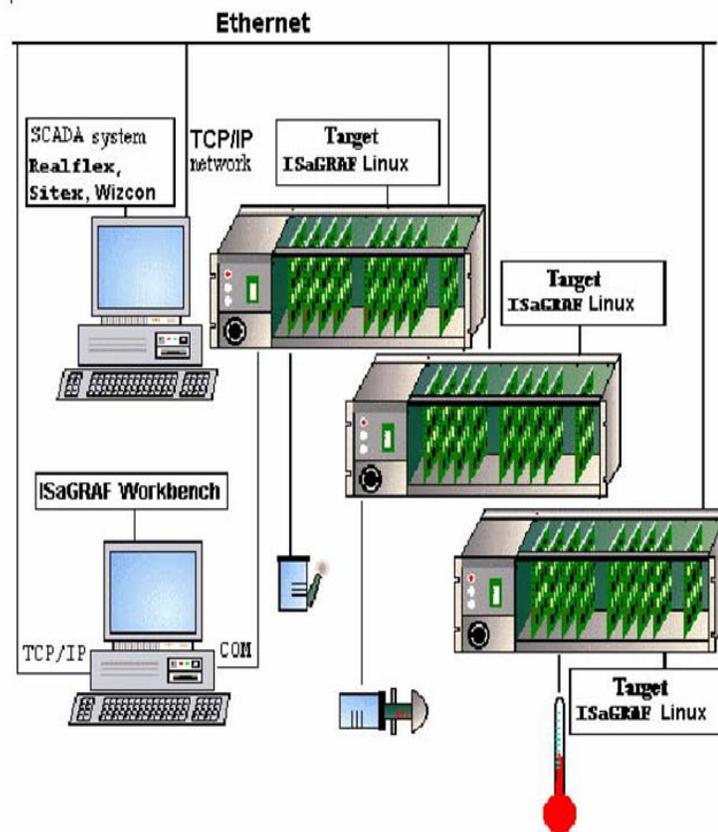
[11]. Industrial Control Links Inc. (2002).ISaGRAF Workbench Manual, disponible en:
<http://www.iclinks.com/support/ISaGRAF/DownloadISaGRAFMans.html>, accedido el
12/02/2006.

[12]. Latin-Tech Inc. (2006).ISaGRAF, disponible en:
<http://www.lt-automation.com/winceplcs.htm>, accedido el 26/02/2006.

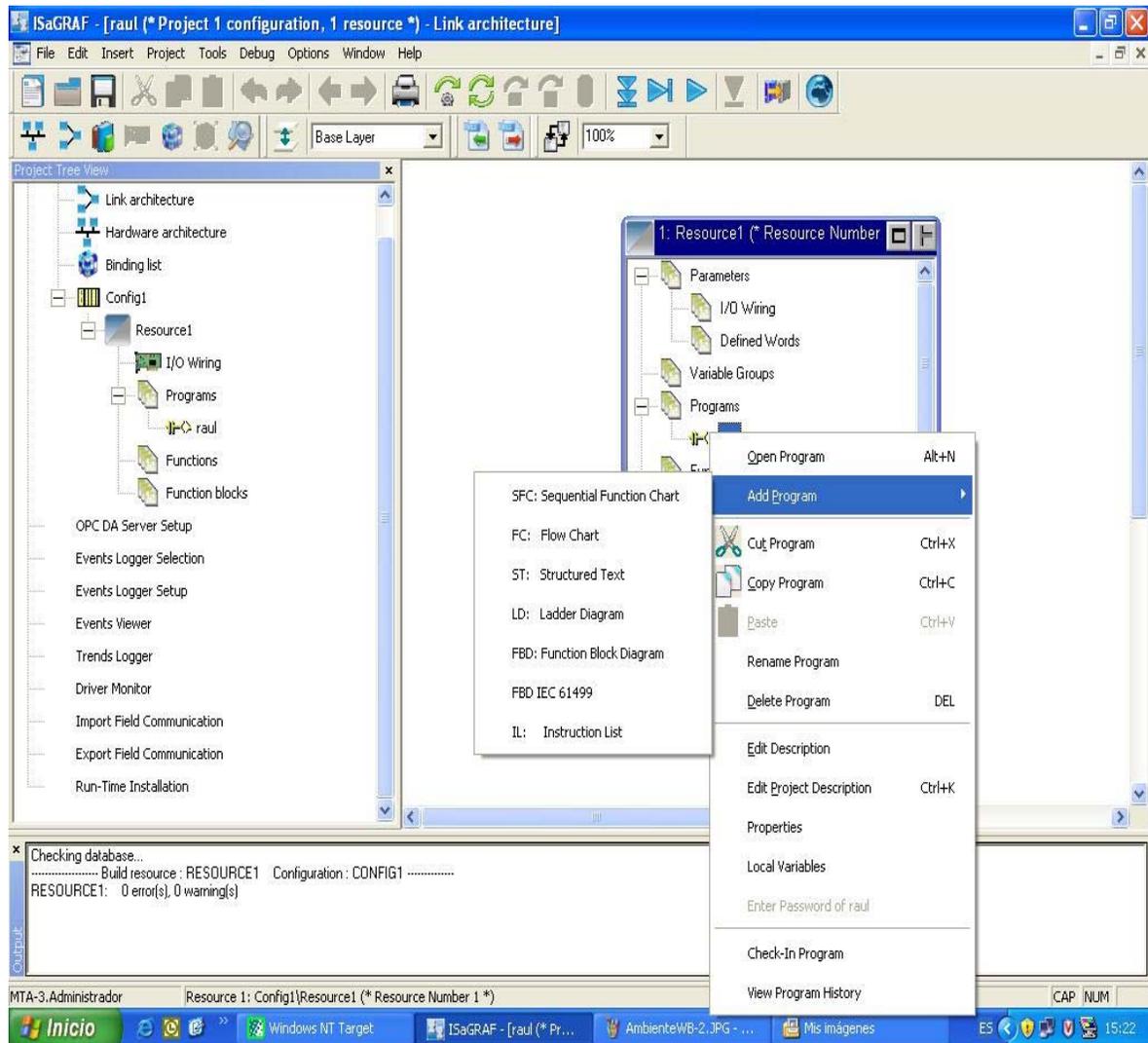
Anexo 1. Representación de los lenguajes de control.



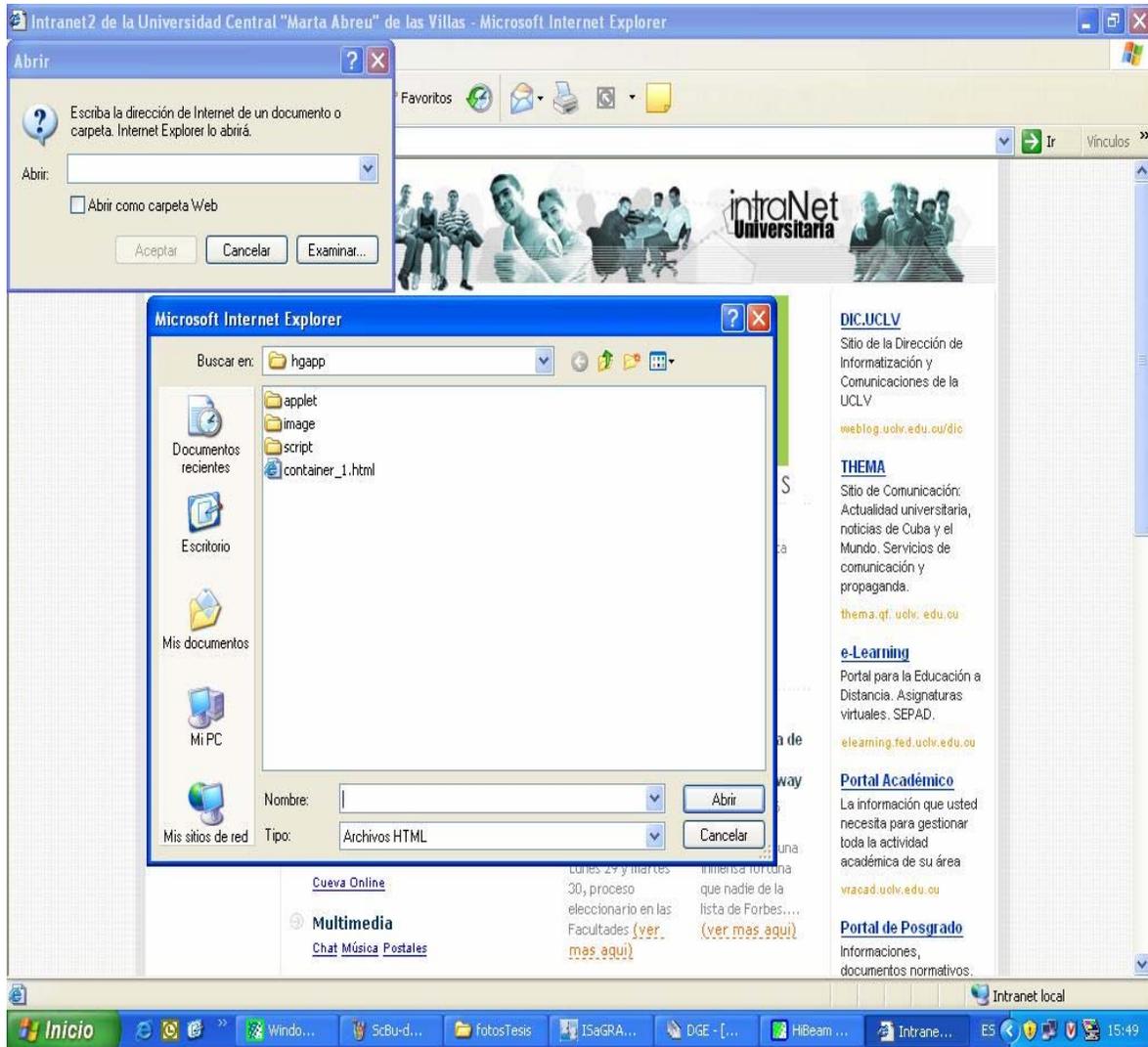
Anexo 2. Esquema de operación del ISaGRAF en ambiente Linux.



Anexo 3. Selección del lenguaje de programación en el Workbench.



Anexo 4. Ejemplo de la descarga del proyecto en la página Web.



Anexo 5. Glosario de Términos

Active X: Lenguaje para la elaboración de aplicaciones exportables a la red y capaces de operar sobre cualquier tipo de plataforma a través de navegadores Web. Desarrollado por la Microsoft.

Apple Macintosh: Nombre de una serie de ordenadores fabricados y comercializados por Apple Computer desde 1984.

CE3.0: Variante del sistema operativo Windows.

Driver: Controlador de un dispositivo.

ETCP: *Extended Transmission Control Protocol* (Protocolo de Control de Transmisión Extendida). Red desarrollada por ICS Triplex ISaGRAF Inc que usa el protocolo TCP/IP.

GPS: Global Positioning System (GPS) o Sistema de Posicionamiento Global.

HMI: *Human/Machine interface* (Interface hombre/máquina).

HTML : *Hipertext Mark-up Language* (Lenguaje de Marcación de Hipertexto). El HTML es el lenguaje de programación que permite el funcionamiento de la Web.

IBM: *International Business Machines Corporation*, conocida coloquialmente como el Gigante Azul, es una empresa que fabrica y comercializa hardware, software y servicios relacionados con la informática.

IEC 1131-3: Norma creada en 1993, que constituyó el primer esfuerzo real para agrupar los lenguajes de programación usados en la automatización industrial y estandarizarlos.

IEC 61131: Norma que representa el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar.

Java: Lenguaje de programación desarrollado para la creación de pequeñas aplicaciones exportables a la red y que son capaces de operar sobre cualquier plataforma normalmente a través de navegadores Web. Desarrollado por la empresa Sun, permite dar dinamismo a las páginas Web.

LAN: *Local Area Network*(Red de Área Local). Red pequeña que conecta a varios ordenadores y que está limitada físicamente.

Netscape: Navegador Web creado por la compañía Netscape Communications, irrelevante en plataforma Windows al expandirse el Internet Explorer de Microsoft.

ODBC: *Open DataBase Connectivity* (Conectividad Abierta para Bases de Datos). Es un estándar de acceso a Bases de Datos desarrollado por Microsoft, que hace posible el acceso a cualquier dato de cualquier aplicación sin importar que sistema gestor los almacene.

OPC: *OLE for Proceses Control*(OLE para Procesos de Control).Creada por un grupo de empresas con el objetivo de llevar al ambiente industrial el estándar OLE(*Object Linkink and Embedding*) creado por Microsoft .El OLE permite crear objetos en una aplicación y luego insertarlo en otra,compartiendo información entre las aplicaciones.

OS: Operating Sistem, sistema operativo.

PLC: *Programmable Logic Controller* (Programador Lógico Programable). Pequeña computadora usada para la automatización de procesos industriales. Usualmente utilizan un microprocesador.

PSOS: Plug-in Silicon Operating System o Sistema Operativo encendido por Silicio, es un sistema operativo en tiempo real y fue creado en 1982 por Alfred Chao.

QNX: Sistema operativo de tiempo real basado en Unix, es desarrollado principalmente para el uso en dispositivos empotrados.

RTX: Plataforma de extensión en tiempo real para el control de Windows.

SCADA: *Supervisory Control and Data Acquisition*(Control Supervisorio y Adquisición de Datos). Sistemas de control que comprenden un ordenador central y uno o más campos de datos y unidades de control.

TCP/IP: *Transmission Control Protocol/Internet Protocol*. Protocolo que se utiliza en Internet.

ThreadX: Sistema operativo en tiempo real diseñado específicamente para aplicaciones demasiado empotradas, está implementado como una librería del lenguaje C.