

José Martínez Torres  
José Manuel Díez Aznar

Aprenda WinCC

Descubra como:

- Diseñar
- Configurar
- Diagnosticar las comunicaciones
- Aprender a programar en VBS
- Automatizar los desarrollos mediante VBA

# Aprenda WinCC

Jose Martínez y Jose Manuel Díez

Aprenda paso a paso a diseñar y desarrollar sus proyectos

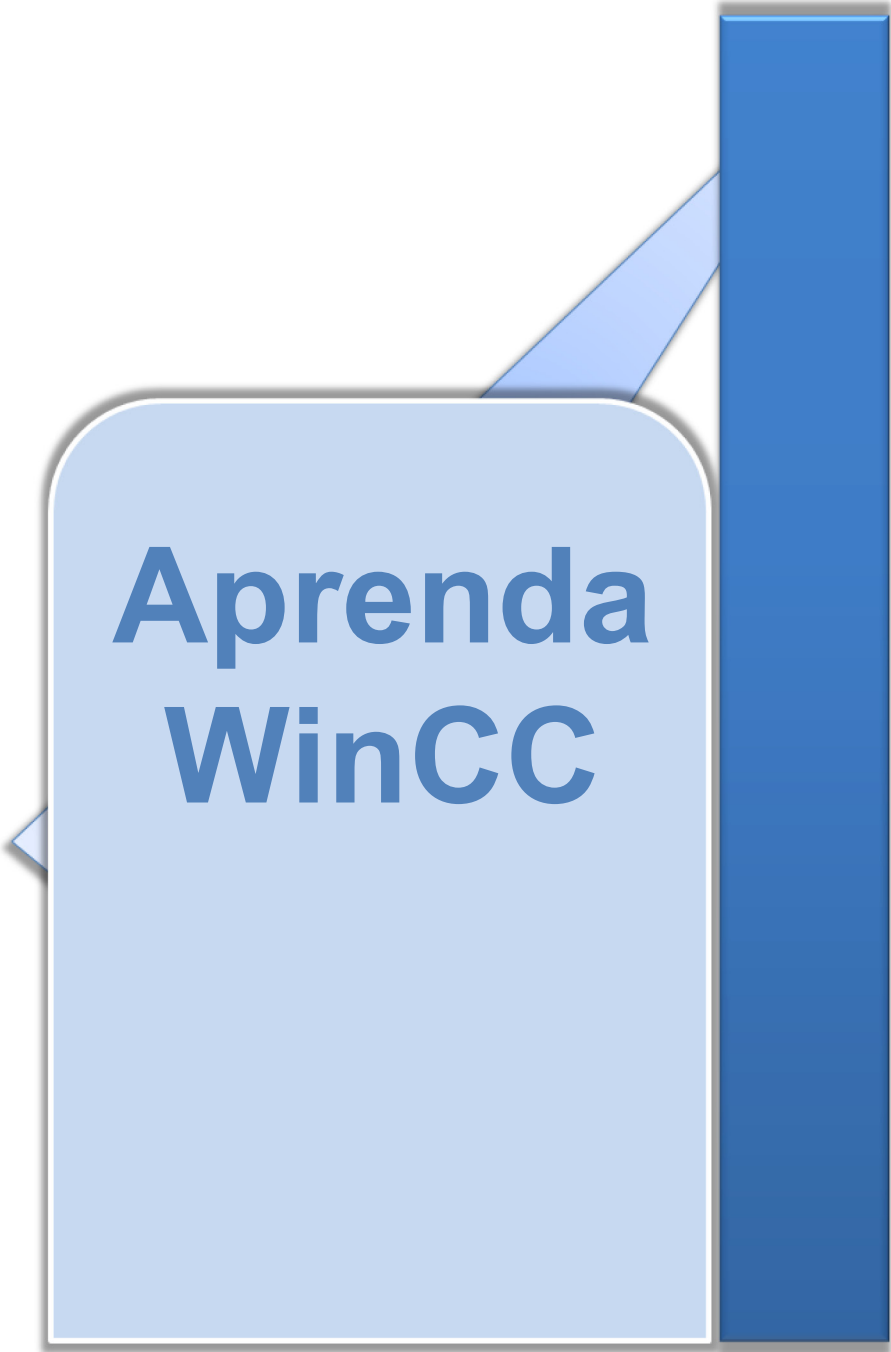


Contiene CD-Rom con versión de prueba del software WinCC



EDITORIAL  
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

José Martínez Torres  
José Manuel Díez Aznar



# Aprenda WinCC

EDITORIAL  
UNIVERSITAT POLITÈCNICA DE VALÈNCIA

**Agradecimientos:**

*Al catedrático D. José Giner, padre del proyecto  
A Javier Díez por saber más incluso que los señores Win y CC juntos*



*Esta editorial es miembro de la UNE, lo que garantiza la difusión y comercialización de sus publicaciones a nivel nacional e internacional*

Primera edición, 2011

© de la presente edición:  
Editorial Universitat Politècnica de València  
[www.editorial.upv.es](http://www.editorial.upv.es)

*Distribución:* [pedidos@editorial.upv.es](mailto:pedidos@editorial.upv.es)  
Tel. 96 387 70 12

© José Martínez Torres  
José Manuel Díez Aznar

**Colaboradores:**

Alejandro Sevilla  
Antonio Revuelta  
Alfredo Canet

Imprime: By print percom sl.

ISBN: 978-84-8363-762-3  
Depósito Legal: V-4332-2011  
Ref. editorial: 931

Queda prohibida la reproducción, distribución, comercialización, transformación, y en general, cualquier otra forma de explotación, por cualquier procedimiento, de todo o parte de los contenidos de esta obra sin autorización expresa y por escrito de sus autores.

Impreso en España

## Dedicado a...

Marc y Carolina, los dos torpeditos

Pilar y Marisu.

Nadie es imprescindible excepto vosotras.

Gracias por acompañarnos en este sueño.



## **Autores**

José Martínez Torres, Dep. Automatización Siemens  
José Manuel Diez, Catedrático de Escuela Universitaria

## **Colaboradores**

D. Alejandro Sevilla, experto en procesos industriales  
D. Antonio Revuelta, Dep. Proyectos Aguas de Alicante  
D. Alfredo Canet, Ing. Técnico Industrial Eléctrico y Mecánico

## **Agradecimientos**

Al catedrático D. José Giner, padre del proyecto  
A Javier Diez por saber más incluso que los señores Win y CC juntos



# Índice

1 .	Introducción a los sistemas scadas.....	13
1.1 .	Funcionamiento básico del WinCC.....	17
1.2 .	Evolución histórica de los scadas.....	23
1.3 .	Tipos de configuraciones.....	26
1.3.1	Sistema Monopuesto.....	26
1.3.2	Sistema Multipuesto.....	29
1.3.3	Sistema Distribuido.....	32
1.3.4	Sistema Redundante.....	34
1.3.5	Sistema Servidor Web.....	35
1.4 .	Fases del diseño.....	38
1.5 .	Fases del desarrollo.....	47
2 .	Primeros pasos con WinCC.....	49
2.1 .	Instalar y desinstalar el WinCC.....	53
2.2 .	Grupos de usuarios y derechos de acceso.....	63
2.3 .	Licencias.....	65
2.4 .	Estructura de un proyecto.....	67
2.5 .	Migración de versiones anteriores.....	69
2.6 .	Arrancar y parar el WinCC.....	71
2.7 .	Copias de seguridad.....	75
2.8 .	WinCC Tray.....	77
2.9 .	Diagnóstico del sistema.....	78





3 .	Diseño de la interface de usuario.....	81
3.1 .	Recomendaciones sobre diseño de pantallas .....	84
3.2 .	Tipos de interfaces de usuario.....	86
3.3 .	Jerarquía de pantallas.....	88
3.4 .	Organización de la pantalla principal .....	90
3.5 .	Un monitor vs multimonitor.....	91
3.6 .	Los colores en WinCC .....	93
3.7 .	Los idiomas en runtime .....	95
4 .	El Explorador de WinCC.....	97
4.1 .	Propiedades del proyecto.....	102
4.2 .	Propiedades del equipo.....	108
4.3 .	El Graphics Designer .....	120
4.4 .	Editor de Menús y Barras de herramientas.....	129
4.5 .	Las referencias cruzadas.....	135
4.6 .	Basic Process Control.....	136
4.6.1	OS Project Editor.....	137
4.6.2	Picture Tree Manager .....	138
4.6.3	Lifebeat Monitoring .....	138
4.6.4	Horn .....	139
4.6.5	Time Synchronization .....	139
5 .	Variables y comunicaciones .....	141
5.1 .	El administrador de variables .....	144
5.2 .	Organización de las variables en el autómatas .....	147
5.3 .	Grupos de variables .....	149
5.4 .	Variables internas.....	151
5.5 .	Variables de comunicaciones .....	155
5.6 .	Estructuras de variables .....	157

---

5.7 .	Exportación e importación de variables .....	160
5.8 .	Simulación de variables .....	164
5.9 .	Conceptos de comunicaciones .....	169
5.10 .	Comunicación con Simatic S7.....	172
5.11 .	OPC.....	176
5.12 .	System Info.....	182
5.13 .	Diagnos de las comunicaciones .....	183
5.13.1	Estado de las variables.....	183
5.13.2	Estado de las conexiones .....	185
5.13.3	El Channel Diagnosis .....	186
5.13.4	Diagnos mediante el asistente dinámico.....	188
5.13.5	Diagnos mediante página Web .....	189
6 .	Dinamización de objetos .....	193
6.1 .	Acceso a las propiedades y eventos .....	196
6.2 .	Programar eventos de los objetos.....	199
6.3 .	Cómo dinamizar propiedades.....	202
6.4 .	Tipo de objetos .....	205
6.5 .	Agrupación de objetos.....	208
6.6 .	Ejemplos de dinamización de objetos .....	209
6.6.1	Representación binaria .....	209
6.6.2	Activación binaria .....	220
6.6.3	Representación analógica .....	222
6.6.4	Activación analógica .....	233
6.7 .	Representación indirecta.....	245
6.8 .	Objetos personalizados .....	247
6.9 .	Objetos Faceplate.....	248
6.10 .	Ventanas parametrizables.....	250

7 .	Programación en VBS .....	257
7.1 .	El editor Global Script VBS.....	260
7.2 .	Subrutinas y funciones .....	264
7.3 .	Acciones.....	266
7.4 .	Declaración de variables.....	268
7.5 .	HMIRuntime, ScreenItems, Screen y Parent .....	269
7.6 .	Operadores.....	275
7.7 .	Control de flujo de programa .....	284
7.8 .	Ficheros de texto .....	287
7.9 .	Leer y escribir la hora .....	289
7.10 .	Depuración de código en VBS .....	289
7.10.1	La ventana de diagnostico .....	289
7.10.2	La ventana de GSC Runtime.....	291
7.10.3	El Microsoft Script Debugger .....	291
7.10.4	La instrucción MsgBox .....	293
7.10.5	Control de errores: On error.....	295
8 .	Integración con Step 7.....	297
8.1 .	WinCC dentro de Step 7 .....	300
8.2 .	Utilizar variables de Step 7 .....	305
8.3 .	Diagnóstico del hardware de Step 7 .....	308
8.4 .	Referencias cruzadas de Step 7 .....	309
8.5 .	Notificar errores de sistema .....	310
8.6 .	Estación de ingeniería ES.....	315
9 .	El sistema de avisos.....	321
9.1 .	Alarm Logging.....	324
9.1.1	Tipos de disparos de avisos.....	325
9.1.2	Bloques de avisos.....	329

9.1.3	Clases de avisos.....	334
9.1.4	Tipos de avisos .....	337
9.1.5	Avisos de grupos .....	339
9.1.6	Histórico de avisos .....	345
9.2 .	Avisos de sistema .....	349
9.3 .	Exportar e importar avisos .....	351
9.4 .	WinCC Alarm Control.....	354
9.5 .	Filtrado de avisos.....	378
9.6 .	Tratamiento de avisos .....	380
9.6.1	Función “GMSFunction” .....	380
9.6.2	Recuperar avisos antiguos .....	381
9.6.3	Crear alarmas en VBS.....	383
9.6.4	Alarmas acústicas.....	384
9.6.5	Enviar un correo electrónico.....	386
10 .	Almacenamiento y representación de datos .....	387
10.1 .	El Tag Logging .....	392
10.2 .	Configuración de ficheros .....	392
10.3 .	Configuración de una variable de fichero .....	399
10.4 .	Archive Tool: crear valores de fichero desde Excel.....	401
10.5 .	Tipos de almacenamiento .....	406
10.6 .	El OnlineTrendControl .....	410
10.7 .	El “RulerControl” .....	427
10.8 .	El “OnlineTableControl” .....	429
10.9 .	El “FunctionTrendControl” .....	430
10.10 .	Actualizar un proyecto existente .....	432
11 .	Gestión de usuarios .....	433
11.1 .	El User Administrator .....	436

11.2 .	Gestión de usuarios.....	441
11.2.1	Cómo cambiar el password de un usuario.....	441
11.2.2	Asignar nivel de autorización.....	442
11.3 .	Login de un usuario en runtime .....	443
11.4 .	Uso de las autorizaciones en el runtime .....	445
12 .	Archivos de usuario.....	449
12.1 .	El editor User Archives .....	452
12.2 .	Archivos de usuario .....	454
12.3 .	Propiedades de un campo.....	459
12.4 .	Vistas .....	461
12.5 .	El UserArchiveControl .....	462
12.6 .	Recetas de productos.....	470
13 .	Informes de impresión.....	475
13.1 .	Trabajar con formatos de página .....	478
13.1.1	Objetos del formato de página .....	479
13.1.2	Crear un formato de página. Ejemplos de aplicación. ....	480
13.1.3	Incluir un formato de página dentro de otro.....	491
13.1.4	Formatos predefinidos.....	491
13.2 .	Trabajos de impresión.....	492
13.2.1	Configurar un trabajo de impresión.....	493
13.2.2	Ejecutar un trabajo de impresión .....	496
13.3 .	Informes usuales .....	499
13.4 .	Documentación de un proyecto de WinCC .....	501
14 .	Opciones de WinCC.....	505
14.1 .	Connectivity Pack .....	507
14.1.1	Acceder a los avisos de WinCC.....	509
14.1.2	Acceder al histórico de datos del WinCC .....	519

14.1.3	Informes desde Excel .....	523
14.1.4	Acceder a las tablas de usuario del User Archives.....	527
14.1.5	WinCCDBVb: el comprobador de código .....	527
14.2 .	Calendar Options.....	530
14.2.1	El editor de Calendar Options .....	531
14.2.2	Crear y configurar calendarios .....	533
14.2.3	Manejo del Calendar Scheduler .....	534
14.2.4	Manejo del Event Notifier.....	541
14.2.5	Calendar Options en runtime .....	550
14.3 .	Multipuesto.....	552
14.3.1	Configurar un sistema cliente-servidor.....	552
14.3.2	El Simatic Shell .....	557
14.4 .	El Web Navigator.....	559
14.4.1	Configuración del servidor Web Navigator.....	560
14.4.2	Publicar un proyecto Web. ....	563
14.4.3	Administración de usuarios .....	566
14.4.4	Acceso de clientes WinCC Web Navigator.....	567
14.4.5	Navegación mediante “Internet Explorer” .....	569
14.4.6	Gestión del área de descargas. ....	571
14.4.7	Aplicación integrada en WinCC: “WinCCViewerRT” .....	576
15 .	Visual Basic for Automation .....	581
15.1 .	El Editor VBA .....	583
15.2 .	Menús y barras de herramientas personalizadas .....	585
15.3 .	Formularios en VBA.....	591
15.4 .	Personalizar el entorno de trabajo mediante VBA.....	593
15.5 .	Trabajar con la librería de objetos .....	595
15.6 .	Personalizar las propiedades y eventos mediante VBA .....	597



15.6.1	Cambiar propiedades de un objeto .....	597
15.6.2	Dinamización directa o indirecta de una propiedad .....	598
15.6.3	Personalizar una propiedad mediante código VBS .....	599
15.6.4	Personalizar todos los objetos de un determinado tipo .....	600
15.6.5	Personalizar los eventos de objetos mediante VBA .....	601
15.7 .	Trabajar con variables mediante VBA .....	602
15.8 .	Trabajar con alarmas mediante VBA .....	605
15.9 .	Un ejemplo de VBA: la "magicbar". .....	607

# INTRODUCCION A LOS SISTEMAS SCADAS

1



Funcionamiento básico  
del WinCC



Evolución histórica de los  
scadas



Tipos de configuraciones



Fases del diseño



Fases del desarrollo





La evolución tecnológica en el campo de la automatización de procesos industriales, ha convertido hoy en día, a la tarea de la supervisión y adquisición de información en, probablemente la de mayor relevancia para la optimización del proceso productivo y el consiguiente ahorro de costes.

El WinCC es el SCADA<sup>1</sup> de Siemens© para todo tipo de sistemas de automatización, desde sistemas monopuesto para un solo ordenador hasta sistemas redundantes con integración en Web. Una de sus principales características es su modularidad, tanto para ser escalable a nivel de datos como para poseer funcionalidades que se pueden incorporar al mismo, a través de módulos de software adicionales también conocidos como *addons*.

Las principales tareas de las que se encarga el WinCC en una instalación industrial son las siguientes:

- Representar el proceso mediante pantallas que se dinamizan con el estado de las variables de comunicaciones.
- Permitir controlarlo, modificando el estado de las variables de los autómatas desde las pantallas del scada.
- Almacenar información tanto de los avisos como de los valores de proceso a intervalos de tiempo o por eventos, para visualizarla mediante graficas o tablas.
- Advertir de la aparición de eventos que se puedan producir en una instalación y, en caso necesario, solicitar al operador que los acuse.
- Aglutinar la información de las variables de proceso, para poder servirla a otras aplicaciones, ya sea en tiempo real o mediante datos almacenados a través de una consulta SQL a su base de datos.

El incremento de las opciones para utilizar la información almacenada en el WinCC, lo ha convertido en el centro neurálgico de la toma de decisiones de

---

<sup>1</sup> SCADA: Acrónimo que procede del término inglés "Supervisory Control And Data Acquisition" (Supervisión, Control y Adquisición de Datos)

multitud de instalaciones, ya sea en relación con el nivel de productividad, del equilibrio de las cargas en las líneas de producción o del control de la calidad, así como un sin fin de decisiones empresariales estratégicas que utilizan esta información como herramienta de decisión.

Desgraciadamente, en demasiadas ocasiones los proyectos de WinCC se realizan sin una sistemática ni análisis previo, obviando por ejemplo, la importantísima relación existente entre la dinamización de los objetos de las pantallas del scada y la disposición de dichas variables dentro de los autómatas. Los errores de planteamiento producidos por la ausencia de una metodología en la elaboración del proyecto se detectan en numerosas ocasiones de manera tardía, siendo usualmente complicada cualquier tipo de modificación posterior debido al riesgo que representa para el proceso, el cual suele encontrarse ya en productivo, cuando no por el elevado coste en horas de desarrollo que implicaría rediseñar de nuevo la aplicación scada.

En este libro, hemos querido realizar un enfoque del WinCC desde la perspectiva del desarrollador, introduciendo ejemplos que puedan mostrar las diferentes funcionalidades tanto de la herramienta como de los principales *addons* de la misma. Estos ejemplos, esperamos sirvan de base al lector para permitirle poder realizar los suyos propios.

Adicionalmente, encontrará información de gran utilidad sobre las diferentes funcionalidades del WinCC en la dirección de internet:

<http://support.automation.siemens.com/WW/view/es/10805584/133000>

A lo largo de los quince capítulos que componen el presente libro se ha intentado exponer la información de la manera más clara y concisa posible, con un lenguaje por y para programadores. Se intentará que realice este viaje por los entresijos del WinCC de una forma amena, en ocasiones con la ayuda de pequeñas dosis de humor (si no aprende, por lo menos diviértase).

## **Los Autores.**

## 1.1 . Funcionamiento básico del WinCC

El WinCC a grandes rasgos se compone de las cuatro partes conceptuales siguientes:



Figura 1.1: Partes conceptuales del WinCC



**Herramienta de desarrollo (CS):** Software con el que se diseña la aplicación. Está dividido en editores, cada uno de ellos dedicados a una tarea específica, los cuales se agrupan bajo el programa WinCC Explorer. Mediante esta herramienta se definen todos los aspectos del WinCC: las alarmas, las pantallas, las gráficas, etc., pudiéndose ejecutar de dos maneras diferentes:

- Mediante el ejecutable:  
“C:\Program Files\Siemens\WinCC\bin\WinCCExplorer.exe”<sup>2</sup>.
- Ejecutando el fichero “<nombre\_del\_proyecto>.mcp” desde el explorador de Windows, el cual se encuentra en el raíz de nuestro proyecto.

**Base de datos del desarrollo:** Base de datos SQL Server en la que se almacena la información generada con la herramienta de desarrollo. La base de datos se encuentra en el raíz de nuestro proyecto y posee el formato “<nombre\_del\_proyecto>.mdf”. Dentro de dicha base de datos, se encuentra, por ejemplo, la tabla de usuarios del sistema y sus *passwords*, la definición de las variables y estructuras, y todo aquello que hemos desarrollado con la herramienta CS.

<sup>2</sup> Se supone que no ha modificado la ruta de instalación por defecto que es C:\Program Files\Siemens. En caso contrario modifique esta ruta para que coincida con la de su instalación.

**Software Runtime (RT):** Software que, utilizando la información de la base de datos de desarrollo, se encarga de mostrar las pantallas del scada, controla la aparición de eventos de alarma y almacena la información en la base de datos runtime. Esta acción coloquialmente se suele denominar “scada ejecutándose” o también “runtime activo”. Si esta aplicación no se está ejecutando, no se produce comunicación con los autómatas, ni se almacena información, ni se visualizan las pantallas desarrolladas, lo que se suele denominar “scada detenido o parado”. Para ejecutarlo podemos utilizar cualquiera de los métodos siguientes:

- Si la última vez que salimos de la herramienta de desarrollo teníamos el runtime en marcha, al abrir de nuevo el WinCC Explorer arrancará de forma automática.
- Si estando dentro del WinCC Explorer, pulsamos sobre el icono de arranque, se abrirá el runtime por la pantalla que hayamos definido como principal en las propiedades de nuestro equipo. Esta es la forma más usual de arrancarlo. En un sistema monopuesto siempre será la misma, mientras que en un multipuesto dependerá de la que hayamos configurado para el equipo en cuestión. 
- En el Graphics Designer podemos pulsar sobre el icono de ejecución. En este caso, se ejecuta en el runtime la pantalla que actualmente tenemos abierta en el editor (no la pantalla principal definida en el proyecto). 
- Si hemos parametrizado la opción “Autostart” (ver capítulo 2 ), al arrancar el ordenador automáticamente se ejecutará el proyecto de WinCC que hayamos asociado. Esta es la forma habitual de ejecutar el runtime una vez finalizado el desarrollo, ya que nos asegura que cada vez que se arranque el ordenador, se ejecutará automáticamente el runtime de WinCC.
- También nos puede interesar poseer varios iconos en el escritorio, cada uno con un acceso directo a diferentes proyectos, y que pulsando sobre ellos arranque el runtime directamente sin necesidad de abrir el WinCC Explorer. Para ello, crearemos un acceso directo en nuestro escritorio con el enlace siguiente:

```
“C:\Program Files\SiemensWinCC\bin\AutoStartRT.exe” <proyecto>.mcp
/Activ:yes
```

A este código, le podemos añadir al final del mismo, el comando “/EnableBreak:yes” que nos visualiza un botón que nos permite cancelar el arranque.

**Base de datos Runtime:** En esta base de datos se encuentra almacenada la información que nos interesa adquirir del proceso productivo. La base de datos runtime se almacena en dos localizaciones distintas:

- Parte de la información, se encuentra contenida en el fichero “<nombre\_de\_proyecto>RT.mdf” en el raíz de nuestro proyecto. Dentro de esta base de datos se encuentran, por ejemplo, los valores de las variables internas persistentes.
- El resto de valores de la base de datos runtime se encuentran en el subdirectorio “ArchiveManager” de nuestro proyecto, el cual es generado en el momento inicial de almacenamiento del runtime (si nunca ha almacenado datos, el subdirectorio aun no aparecerá en su proyecto). Esta información está dividida en tres subdirectorios: “AlarmLogging” para el histórico de alarmas, “TagLoggingFast” para el histórico de valores cuyo periodo de almacenamiento es inferior a un minuto o por evento, y “TagLoggingSlow” para el resto de valores.



**Figura 1.2 : Base de datos runtime de un proyecto y los segmentos SQL que la componen**

Tanto la base de datos de desarrollo como la de runtime se encuentran en formato SQL Server. A título informativo, indicar que mediante la herramienta “Inicio->Programas->Microsoft SQL Server 2005->SQL Server Management Studio” se puede abrir tanto la base de datos de desarrollo de WinCC como la de runtime.<sup>3</sup>. En el caso de que desee visualizar estas bases de datos en el Manager Studio, es necesario tener abierto el proyecto en el WinCC Explorer previamente. Las bases de datos de un proyecto de WinCC dentro del SQL Server presentan la siguiente estructura:

**CC\_<los 8 primeros caracteres de nuestro proyecto>\_<fecha y hora de creación><sup>4</sup>**

<sup>3</sup> Recuerde que siempre gestionaremos las bases de datos de WinCC a través del *WinCC Explorer* y sus editores, nunca a través de la herramienta de gestión de bases de datos de Microsoft.

<sup>4</sup> La base de datos de runtime es similar pero finalizar con el carácter R.

Mediante “Tables-><tabla>-> ->Open Table” se puede ver los datos de cada una de las tablas, Figura 1.3.

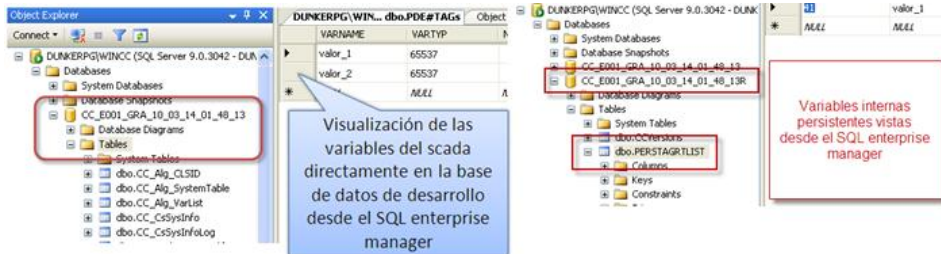


Figura 1.3 : Tablas de la base de datos de desarrollo y runtime de WinCC

Las diferentes combinaciones a la hora de ejecutarse el desarrollo y runtime de WinCC nos dan los siguientes escenarios posibles:

Tipo	Significado
<p><b>Ejecutarse únicamente el desarrollo</b></p>	<p>Se puede desarrollar el proyecto de WinCC sin tener el runtime en ejecución. Esta posibilidad de desarrollo no es muy común, ya que es preferible observar el resultado de lo que se está diseñando en tiempo real a través del runtime. Este escenario solo se utiliza en las fases iniciales del desarrollo, cuando con frecuencia no se dispone todavía de los autómatas con los que se debe de comunicar. En cualquier caso, aun en esta circunstancia, es recomendable utilizar herramientas como el simulador de Simatic PLCsIm. A través de este software podremos simular los autómatas S7 de nuestra instalación (más información en capítulo 0).</p>
<p><b>Ejecutarse únicamente el runtime</b></p>	<p>Es la más común de las situaciones, ya que siempre que no se esté realizando ninguna modificación en el desarrollo del sistema, el runtime será la única herramienta que se es necesario ejecutar. El hecho de que únicamente se ejecute el runtime y no se encuentre abierto el desarrollo en paralelo en las instalaciones es principalmente por un motivo: evitar que una posible confusión en el operador le haga acceder al desarrollo y modificar involuntariamente el proyecto creyéndose que se encuentra en el runtime.</p>
<p><b>Ejecutarse en paralelo</b></p>	<p>Es lo que comúnmente se denomina “desarrollo online”, es decir, se realizan modificaciones en el scada mientras se está ejecutando la aplicación. Es la forma más común de desarrollo ya que no requiere de ningún tipo de configuración previa (se alterna entre las dos aplicaciones, runtime y desarrollo a través de la combinación de teclas “ALT+TAB”).</p>

Existen dos posibles filosofías a la hora de desarrollar un proyecto de WinCC:

Filosofía	Ventaja	Inconveniente
<b>Desarrollar en el mismo ordenador</b> donde se está ejecutando el runtime.	Se requiere únicamente un ordenador tanto para el runtime como para el desarrollo.	Mientras el programador está desarrollando la aplicación, el operador de planta no puede realizar sus funciones y viceversa, interrumpiéndose mutuamente en las fases finales del proyecto.
<b>Desarrollar en otro ordenador</b> , denominado "estación de ingeniería" o ES, que transfiere las modificaciones al proyecto que se está ejecutando en un segundo ordenador, denominado: "estación de operación" o OS, en el cual se encuentra el operador gobernando el proceso.	Permite al operador utilizar el scada mientras el programador desarrolla en la estación de ingeniería.	Requiere de dos ordenadores para el desarrollo de la aplicación y además es necesario realizar unos ajustes en el proyecto de WinCC descritos en el capítulo 8.6 . Se utiliza en proyectos cuya puesta en marcha previsiblemente va a ser prolongada y se solapará con la fase de arranque de la producción.

El runtime ejecutará de manera cíclica las siguientes tareas desde que arranca hasta que se detiene, ver Figura 1.4. A esta ejecución cíclica de las tareas durante el runtime se le denomina "ciclo del runtime".



Figura 1.4 : Tareas que realiza el runtime de WinCC cíclicamente



Conocer el ciclo del runtime es importante para el desarrollador, ya que su comprensión nos puede aclarar una serie de dudas que son comunes durante el proceso de aprendizaje del WinCC, como pueden ser:

Cuestión	Explicación
<b>Refresco de variables</b>	<p>Solo se actualiza el valor de las variables que se utilizan en la pantalla actual para dinamizar objetos, las que disparan alarmas, las que se deben almacenar en el histórico porque se produce el evento correspondiente, y aquellas que disparan acciones globales cíclicas.</p> <p>Por este motivo el tiempo que tarda el ciclo de runtime del WinCC no depende tanto de la cantidad de variables que tengamos definida en nuestro proyecto (ya que muchas no se refrescan), como de la habilidad del programador para programar el disparo de eventos únicamente cuando sea necesario. Por ejemplo, si en un proyecto con pocas variables dinamizamos las propiedades de los objetos de una determinada pantalla por el evento "al cambiar" y almacenamos datos en el histórico cada segundo, nuestro ciclo de runtime será mucho mayor que si en otro proyecto de muchas más variables dinamizamos muchos más objetos pero mediante eventos condicionados a variables y cada segundo, y también almacenamos los valores cada minuto en la base de datos.</p>
<b>Refresco de alarmas</b>	<p>Todas las variables que disparen alarmas se refrescan cada segundo (valor prefijado por el sistema), independientemente de si se utilizan o no en la pantalla actual. Esto es necesario porque necesitamos saber que una determinada alarma se ha producido en nuestra instalación aunque no la utilicemos en la pantalla que estamos representando actualmente en el runtime.</p>
<b>Refresco de tareas</b>	<p>Todas las acciones cíclicas tanto de "C" como de "VBS" que hayamos programado deben de poseer un tiempo de ejecución suficientemente amplio como para no entorpecer el ciclo del runtime. En caso contrario (por ejemplo acciones globales que se ejecuten cada 500 ms) se produce un efecto catastrófico sobre el ciclo de runtime.</p>
<b>Periodo de almacenamiento de valores</b>	<p>Determinar el periodo de almacenamiento necesario para que posteriormente un análisis de los datos recopilados no sea pobre, pero que a la vez no contenga "demasiada información irrelevante" es un aspecto crucial. Si la temperatura de la sala no varía de manera apreciable en 5 minutos, carece de sentido almacenar dicho valor cada minuto (y mucho menos cada segundo).</p>

## 1.2 . Evolución histórica de los scadas

Los scadas han sufrido en los últimos años una rápida evolución, partiendo de sistemas personalizados para cada una de las instalaciones mediante lenguaje compilado (C o Delphi principalmente), hasta la situación actual, en la que las herramientas comerciales que existen en el mercado cubren prácticamente la totalidad de las necesidades de las instalaciones industriales. Dicha evolución se puede resumir en las siguientes fases, agrupadas aproximadamente por décadas:

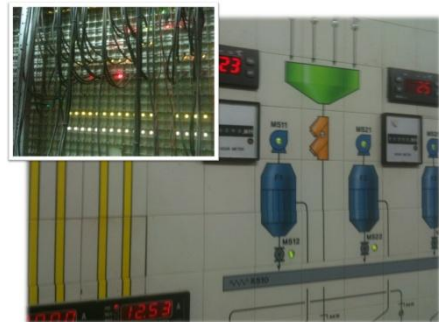
### Década de los años 70

Para el control del proceso se empleaban grandes armarios eléctricos en los cuales la automatización se realizaba mediante series de relés. Como sistema de supervisión y control se utilizaban cuadros sinópticos que representaban el proceso.

Estos cuadros, construidos con metacrilato serigrafiado, tenían troqueladas una serie de lámparas que según su posición dentro del dibujo, representaban el estado del elemento que se encontraba en dicha posición. También se utilizaban displays encastrados en el sinóptico para representar los valores analógicos en tiempo real.

El principal inconveniente de este sistema era la rigidez frente a las modificaciones en la instalación, lo que muchas veces los dejaba obsoletos al poco tiempo de construirse (a veces incluso antes de inaugurar la instalación). Se intentó solventar este problema dividiendo los sinópticos en cuadrados de pequeño tamaño (conocidos como dados). De esta forma si se deseaba modificar una parte del sinóptico sólo era necesario desmontar un grupo de ellos (los pertenecientes a la zona afectada).

Posteriormente se introdujeron en dichos sinópticos tarjetas de comunicaciones entre los mismos y los autómatas, con el objetivo de evitar tener que utilizar una salida de autómata por cada piloto del sinóptico).



**Figura 1.5 : Sinóptico mural con detalle del cableado posterior de los pilotos luminosos**

## Década de los años 80

Los fabricantes incorporan en el mercado tarjetas con capacidad de intercambiar datos con los autómatas a través de buses de comunicaciones. Se comienzan a utilizar pantallas alfanuméricas, los ancestros de los actuales paneles de operador, que a través de funciones de programación que se ejecutaban en la memoria de los autómatas eran capaces de ir variando la información que mostraban. Debido a las enormes deficiencias técnicas de los sistemas de supervisión, en esta época se mantiene para el control manual de la instalación los selectores físicos en las puertas de los cuadros eléctricos e innumerables displays para poder realizar el seguimiento del proceso.

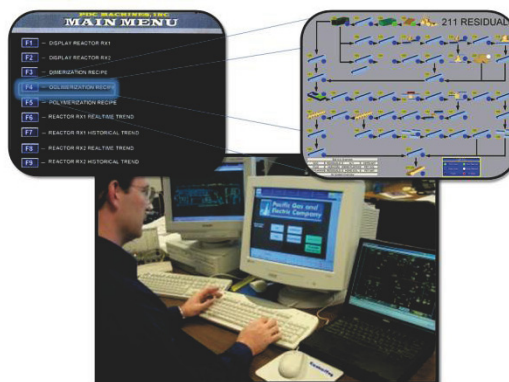


Figura 1.6 : Scada programado en C

Ya existen en esta época scadas comerciales orientados a grandes instalaciones, manteniéndose el uso de aplicaciones personalizadas para las instalaciones de tipo medio y pequeño. El elevado coste de las licencias, unido a la limitación funcional de los scadas comerciales, hace que su campo de actuación sea muy limitado, generalizándose la proliferación de aplicaciones personalizadas. Esta situación genera la siguiente problemática:

- Al ser un sistema cerrado, cuyo código interno solo es conocido por la ingeniería que desarrolla el scada, se crea una fuerte dependencia del cliente hacia la misma de cara al futuro, ya que en muchos casos sustituir a la ingeniería implica comenzar prácticamente de cero.
- La necesidad de compilar el código para poder visualizar el resultado en runtime impedía las modificaciones online con el proceso en productivo, siendo necesario parar la aplicación para realizar los cambios.
- No existía en la mayoría de los casos la posibilidad de crear niveles de *passwords* para diferentes tipos de usuarios, ni registrar los accesos de los mismos al sistema.
- No existía un sistema de comunicación estándar entre aplicaciones informáticas. La integración de los scadas con otros sistemas informáticos se realizaba usualmente a través de lo que se denominaba “fichero plano”, es decir, un fichero de texto en el que previamente se había de-

terminado el orden en que debían de aparecer los datos, de tal manera que emisor y receptor eran capaces de crear y leer el fichero de texto de manera coordinada.

- Sistema exclusivamente monopuesto: la aplicación funcionaba siempre de forma local en un ordenador. Al no existir sistemas redundantes era de vital importancia realizar copias periódicas de la base de datos para poder restaurarla en caso de error.

### Década de los 90 en adelante

La rápida evolución de la informática impulsa a su vez la de los scadas. Para hacer frente a las nuevas necesidades no es suficiente con adquirir información del proceso en cuestión y mostrarla in situ, se convierte en imprescindible compartir y poder explotar dicha información.

A partir de la mitad de la década, gracias al avance tecnológico de las comunicaciones, se introduce la intercomunicabilidad entre cualquier dispositivo del proceso desde cualquier parte del mundo a través de redes basadas en el protocolo TCP/IP.

Hoy en día, carece de sentido utilizar programación específica (salvo para acciones puntuales que un scada en concreto no tenga contemplada), y todas las instalaciones se diseñan con productos estándar. Dentro de esta oferta de software en el mercado, el WinCC presenta todas estas interesantes características:

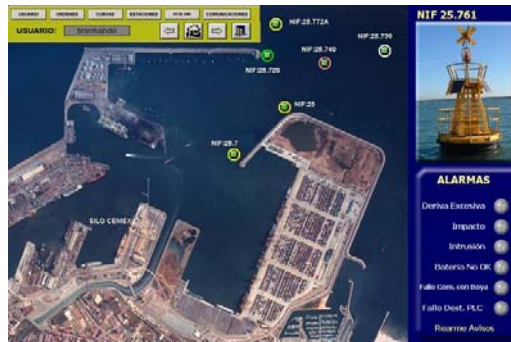


Figura 1.7 : Scada foto realista (cortesía de la Autoridad Portuaria de Valencia)

- La posibilidad de integrar objetos “ActiveX” o “.Net” abre a los programadores un abanico de posibilidades a la hora de integrar sus propios desarrollos dentro de WinCC.
- Conectividad con una amplia variedad de autómatas y buses: el WinCC se suministra con numerosos drivers de comunicación para la mayoría de los fabricantes de autómatas, lo que nos permite acceder directamente al mapa de memoria de los mismos. Aquellos autómatas para los que no dispone WinCC de driver nativo pueden interconectarse mediante un servidor OPC que posea dicho driver.

- Capacidad de intercambio de datos con otras aplicaciones: el WinCC permite intercambiar datos con servidor o cliente OPC, o extraer información de su base de datos mediante la herramienta Connectivity Pack desde aplicaciones externas.
- Mejora en la representación gráfica de datos: las posibilidades graficas de los objetos que podemos insertar en nuestras pantallas (degradados, sombreados, efectos 3D) permite reducir considerablemente, si no evitar en casi todas las ocasiones la necesidad de acudir a herramientas de dibujo gráfico para generar nuestras imágenes que representen los objetos de nuestra instalación.
- Lenguaje de programación estándar: el WinCC permite programar tanto en Visual Basic como Ansi C sus acciones y dinimizaciones.
- Seguridad en las aplicaciones: es posible realizar la gestión de los usuarios del scada desde el administrador de usuarios de Windows, descargando de la tarea de dar de alta y baja a dichos usuarios por parte del programador del scada. Existen ya certificados de firma digital para procesos de alto riesgo (principalmente alimentarios y químicos), con objeto de asegurar la autoría de las modificaciones en el proceso desde el scada.
- Acceso Web: el acceso desde internet al WinCC es actualmente completo, permitiéndonos tanto visualizar como modificar valores de proceso desde cualquier navegador, o utilizar uno propio de WinCC.

### 1.3 . Tipos de configuraciones

El WinCC es un sistema escalable lo cual significa que las distintas configuraciones que vamos a analizar a continuación pueden ser implementadas con unas mínimas modificaciones en el proyecto original.

En cualquier caso, siempre es interesante conocer las posibles configuraciones típicas y así poder dimensionar correctamente la cantidad de licencias que necesitaremos en nuestro proyecto.

#### 1.3.1 Sistema Monopuesto

Este tipo de configuración es la más básica y también la más habitual. En ella, en un solo ordenador reside tanto la aplicación como la base de datos runtime. El sistema monopuesto está enlazado con los autómatas mediante un bus industrial (generalmente Profibus o Profinet), adquiriendo de los mismos la información que almacena en su base de datos. Se necesita exclusivamente la licencia de las variables de comunicaciones que utilizemos. En el caso de que

almacenemos en el histórico de datos una cantidad mayor de 512 variables de fichero, necesitaremos una licencia de ampliación del SQL Server (pudiendo ir desde 1.500 variables de archivo hasta un máximo de 120.000).

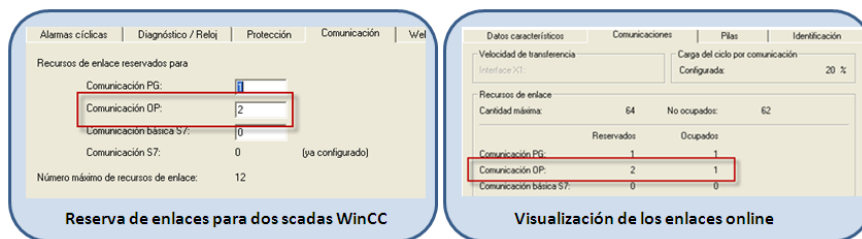


Figura 1.8 : Topología del sistema monopuesto

La configuración monopuesto no implica necesariamente que sólo exista un ordenador. Se puede perfectamente diseñar un sistema monopuesto, y que otros ordenadores mediante aplicaciones informáticas accedan a la base de datos del WinCC para leer información o cambiar datos en la misma (por ejemplo, con aplicaciones como Excel o Access).

Tampoco se debe pensar en una aplicación monopuesto como un único WinCC. Puede interesarnos aumentar la disponibilidad del sistema duplicando el scada actual, de tal manera que únicamente necesitaremos otra licencia de runtime (no licencias adicionales de servidor o de redundancia). Evidentemente debemos tener en cuenta los siguientes aspectos a la hora de duplicar el WinCC dentro de un proyecto de automatización:

- Duplicamos la cantidad de variables a solicitar de los autómatas, o lo que es lo mismo, aumentamos al doble el tiempo de adquisición para las variables del scada.
- Consumimos recursos de la CPU del autómata. Estos recursos deben de ser reservados en el hardware del autómata en el caso de los S7 300. Si optamos por esta solución (duplicar el WinCC) será necesario reservar los enlaces que utilizarán los WinCC en el hardware del autómata, como muestra la Figura 1.9. Observe que el tipo de enlace de comunicación que utiliza el WinCC es comunicación OP.



**Figura 1.9 : Reserva de dos enlaces en Step 7 para dos WinCC monopuesto**

- Conseguimos disponibilidad (el sistema sigue funcionando si cae un ordenador) pero no coherencia de las bases de datos (al volver el sistema caído a estar operativo no recupera los datos perdidos, por lo que siempre tendremos un “hueco” durante dicho intervalo de tiempo en el ordenador que no estuvo operativo).
- Las modificaciones que realizamos en uno de ellos no se reflejan en el otro hasta que no realizamos una copia del proyecto en el segundo ordenador, lo que obliga a detener el runtime.

Las principales **ventajas** que tiene este tipo de configuración son las siguientes:

- Implementación sencilla: simplemente se necesita instalar la aplicación en un ordenador y configurar las comunicaciones con los autómatas.
- Coste reducido: por un lado únicamente necesitaremos una licencia de runtime, y a la vez los requerimientos hardware son inferiores a las otras configuraciones, ya que no debe servir información a otros ordenadores.
- Centralización de datos: todos los datos de la aplicación están ubicados en una única base de datos. Esto simplifica las copias de seguridad, ya que generalmente, salvaguardando un subdirectorio del ordenador, se posee una copia de todos los datos de la instalación.

Los principales **inconvenientes** de este tipo de configuración son los siguientes:

- Acceso limitado del número de usuarios: sólo puede acceder un usuario simultáneamente al WinCC.
- Baja disponibilidad: una anomalía en el ordenador impide el control y visualización de la instalación. En instalaciones que permiten un cierto margen de parada, se puede utilizar como solución alternativa la implementación de una copia exacta del scada en un ordenador de reser-

va. En caso de que deje de funcionar el actual, se puede realizar una sustitución del equipo averiado en poco tiempo, teniendo en consideración que la información a partir de ese momento se encontrará segmentada entre los dos ordenadores: el averiado y el de *backup*.

Ejemplos de aplicaciones típicas de este tipo de configuración serían control de máquinas de producción que necesiten almacenar información para su explotación en local, control de procesos tecnológicos concretos, como un sistema de depuración de aguas, control de la climatización, alumbrado, etc.

### 1.3.2 Sistema Multipuesto

Un sistema multipuesto está compuesto por un ordenador servidor (que solicita y envía información a los autómatas) y ordenadores clientes, los cuales están conectados a dicho servidor (ver Figura 1.10). Se utiliza este tipo de configuraciones cuando se debe visualizar y/o controlar el proceso desde varias ubicaciones separadas entre sí.

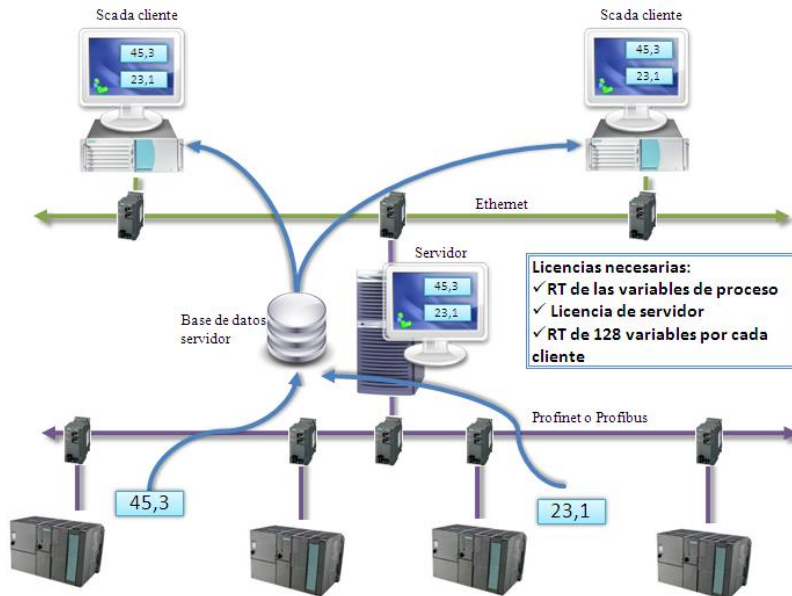


Figura 1.10 : Sistema multipuesto

La información de los autómatas se recibe en el servidor a través del bus de comunicaciones industrial. Los clientes acceden a dicha información siempre a través de Ethernet. Por consiguiente, los clientes realmente no acceden a la información de los autómatas, sino a la base de datos del servidor. Esto tiene



**Para seguir leyendo haga click aquí**