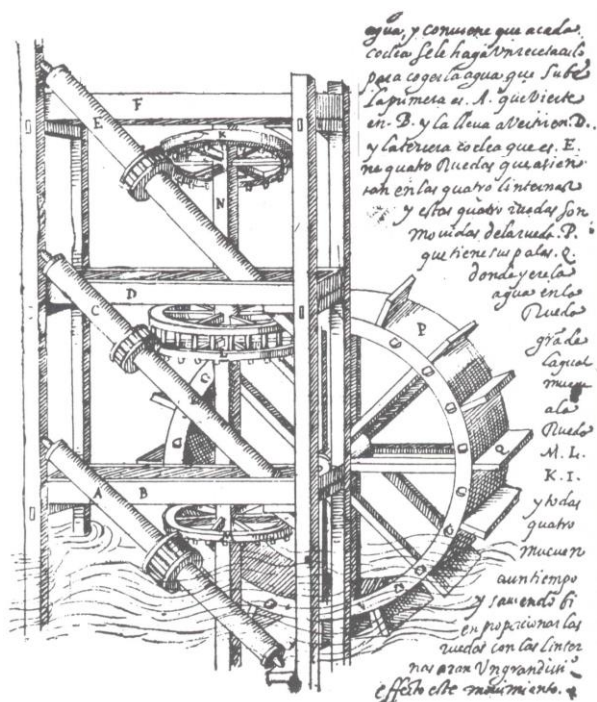




UNIVERSIDAD DE GRANADA

Departamento de Arquitectura y Tecnología de
Computadores



**INTEGRACIÓN DE SISTEMAS PARA LA
SUPERVISIÓN, CONTROL Y OPTIMIZACIÓN
DE REDES DE ABASTECIMIENTO DE AGUA**
(Aplicación a la ciudad de Granada)

TESIS DOCTORAL

Miguel Damas Hermoso

Granada, Julio 2000

UNIVERSIDAD DE GRANADA



INTEGRACIÓN DE SISTEMAS
PARA LA SUPERVISIÓN, CONTROL Y OPTIMIZACIÓN
DE REDES DE ABASTECIMIENTO DE AGUA
(*APLICACIÓN A LA CIUDAD DE GRANADA*)

Miguel Damas Hermoso

TESIS DOCTORAL

DIRECTORES:

Gonzalo Olivares Ruiz
Francisco Gómez Mula

2000

Departamento de Arquitectura y Tecnología de Computadores

D. Gonzalo Olivares Ruiz y D. Francisco Gómez Mula, Profesores Titulares de Universidad del Departamento de Arquitectura y Tecnología de Computadores

CERTIFICAN

Que la memoria titulada: "*Integración de sistemas para la supervisión, control y optimización de redes de abastecimiento de agua (Aplicación a la ciudad de Granada)*" ha sido realizada por **D. Miguel Damas Hermoso** bajo nuestra dirección en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada para optar al grado de Doctor en Informática.

Granada, a 22 de Mayo del 2000

Fdo. Gonzalo Olivares Ruiz
Director de la Tesis

Fdo. Francisco Gómez Mula
Director de la Tesis

A mi familia,
y amigos.

Agradecimientos

Son muchas las personas a las que quiero reconocer su ayuda durante la realización de todo este trabajo y memoria pero, entre ellas, hay algunas que me gustaría destacar.

En primer lugar a los doctores *D. Gonzalo Olivares Ruiz* y *D. Francisco Gómez Mula*, mis directores, cuyo apoyo nunca me ha faltado, ni en el ámbito laboral ni en el personal.

A todos los miembros del *Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada*, y en especial a *D. Julio Ortega Lopera*, *D. Francisco Pelayo Valle* y *D. Alberto Prieto Espinosa* por su inestimable ayuda y por la confianza depositada en mí. En particular, me gustaría también agradecer especialmente la colaboración desinteresada de *D. José Manuel Domingo Bravo* y *D. Héctor Pomares Cintas*.

A todo mis antiguos compañeros de *Ingeniería y Control Remoto S.A.*, empresa donde comencé todo este trabajo como becario del programa de ayudas al intercambio de personal investigador entre industrias y organismos públicos de investigación del Ministerio de Educación y Ciencia (entre 1991 y 1995). Especialmente, a *D. José Luis Correa Cabrera* por su aportación en cuanto al diseño de las aplicaciones SCADA.

A la *Empresa Municipal de Abastecimiento y Saneamiento de Agua de Granada*, por su cooperación y confianza al permitir las pruebas de todos los sistemas desarrollados y descritos en esta memoria. Más concretamente, a *D. Miguel Rodríguez*, *D. Ramón Carlos Valor*, *D. Andrés Hernández* y *D. José Correa* por compartir conmigo sus valiosos conocimientos en torno a la explotación del agua, así como a la *Unidad de Desarrollo Tecnológico (UDT)* del Grupo AGBAR (Aguas de Barcelona) por sus recomendaciones referentes al gestor de comunicaciones distribuidas y funcionalidades del software de supervisión, control y explotación de redes de agua.

Además, he de indicar que el trabajo realizado ha sido financiado en parte por los siguientes proyectos de I+D:

- "*Realización de una red industrial reconfigurable de telemedida y telecontrol*". CICYT (PTR90-0040).
- "*Red de Telecontrol Distribuido*". Consejería Industria. Junta de Andalucía (1994).
- "*Un sistema de control distribuido neuro-difuso con capacidad de aprendizaje y de adaptación continua*". CICYT (TAP97-1166).
- "*Algoritmos masivamente paralelos para problemas de optimización combinatoria en microcomputadores y redes de computadores*". CICYT (TIC97-1149).

Por último, se lo agradezco y se lo dedico a una buena amiga que me ha apoyado siempre, incluso en estos últimos meses de estrés, y a toda mi familia, en especial a mis padres, que "ya tenían ganas de que terminase todo esto y me dedicara más a ellos".

Resumen

El trabajo de tesis doctoral presentado en esta memoria es una contribución al *diseño*, *integración* y *optimización* de sistemas de control distribuido aplicados a grandes redes de abastecimiento de agua.

En cuanto al *diseño*, se describe la metodología que se ha seguido para implantar los sistemas propuestos en la supervisión y control de la red de abastecimiento de agua de la ciudad de Granada. En concreto, se ha dedicado un capítulo completo a mostrar cómo usar un entorno abierto y a la vez unificado, que reúne aplicaciones con conectividad a través de normalizaciones internacionalmente reconocidas.

En cuanto a la *integración*, se han desarrollado nuevos elementos necesarios para el control distribuido de procesos de explotación del agua, y que representan una clara innovación frente a los sistemas de control industrial existentes.

Con respecto a la *optimización*, se han propuesto e introducido un conjunto de herramientas y procedimientos para mejorar la explotación en tiempo real de los sistemas de abastecimiento de agua. Además, se ha mostrado cómo abordar ciertos problemas (predicción y distribución automática) utilizando metodologías de computación basadas en redes neuronales y computación evolutiva. Por último, se han abierto nuevas líneas de investigación y desarrollo en el campo de la optimización aplicada a la industria del agua, utilizando dichos paradigmas de computación con los datos y la operatividad del sistema creado.

La metodología propuesta ha sido validada así mismo en otros campos de la industria relacionados o no con el ámbito del agua, y ha sentado las bases para el desarrollo y expansión de la compañía Ingeniería y Control Remoto S.A. participada por la Universidad de Granada.

Indice

CAPÍTULO 1: INTRODUCCIÓN	1
1.1 Planteamiento del problema y necesidades	2
1.2 Objetivos	4
1.3 Estructura de esta memoria	5
CAPÍTULO 2: FUNDAMENTOS DE LA AUTOMATIZACIÓN DE PROCESOS.....	7
2.1 Fabricación integrada por computador. El concepto CIM	9
2.2 Sistemas de automatización	12
2.2.1 Introducción a los PLC.....	13
2.2.2 Programación de PLC	14
2.3 Software para la supervisión y el control de procesos	16
2.3.1 Introducción	16
2.3.2 Conceptos generales sobre SCADA	18
2.3.2.1 Entorno de desarrollo	19
2.3.2.2 Adquisición de datos del proceso.....	20
2.3.2.3 Gestión de sinópticos	21
2.3.2.4 Gestión de archivos históricos.....	22
2.3.2.5 Las alarmas.....	23
2.3.2.6 La seguridad	24
2.3.2.7 Comunicaciones	25
2.3.2.8 Paquetes SCADA	26
2.4 Técnicas modernas aplicadas a la optimización de procesos.....	28
2.4.1 Introducción	28
2.4.2 Redes Neuronales	30
2.4.3 Neuro-programación dinámica.....	33
2.4.4 Algoritmos Genéticos.....	36
2.4.5 Lógica difusa	39
CAPÍTULO 3: EL CICLO INTEGRAL DEL AGUA EN LA CIUDAD DE GRANADA.....	43
3.1 Introducción.....	44
3.2 Redes de distribución de agua	45
3.3 El ciclo integral del agua en la ciudad de Granada.....	46

3.3.1	Estructura general	47
3.3.2	Captaciones.....	48
3.3.3	Estación de tratamiento de agua potable	48
3.3.4	Red de alta	50
3.3.5	Red de baja	51
3.3.6	Red de saneamiento.....	52
3.3.7	Estaciones depuradoras de aguas residuales.....	53
3.4	Tablas de sistemas y esquema general del abastecimiento	55
CAPÍTULO 4: SISTEMA DE CONTROL PARA LA RED DE DISTRIBUCIÓN DE AGUA DE LA CIUDAD DE GRANADA		59
4.1	Pasado, presente y futuro.....	60
4.2	Descripción general del sistema.....	62
4.3	Estación central	70
4.3.1	Sistema de transmisión de mensajes por telefonía móvil	73
4.3.1.1	Introducción.....	73
4.3.1.2	Descripción general.....	74
4.3.1.3	Características funcionales	75
4.3.1.4	Detalles de programación y ejecución.....	76
4.4	Estaciones remotas	78
4.4.1	Estaciones remotas con PLC	78
4.4.1.1	Descripción del hardware.....	80
4.4.1.2	Estructura del software.....	80
4.4.2	Estaciones remotas con SP-5000.....	84
4.4.2.1	Descripción del hardware.....	85
4.4.2.2	Estructura del software.....	86
4.5	Comunicaciones	89
4.5.1	Sistemas de telecontrol vía radio	89
4.5.2	El sistema SPIRAL.....	89
4.5.3	Modelo de capas del sistema de comunicaciones utilizado.....	90
4.6	Conclusiones.....	93
CAPÍTULO 5: GESTOR DE COMUNICACIONES DISTRIBUIDAS.....		95
5.1	Descripción general	96
5.2	Arquitectura del sistema	97
5.3	Descripción de los módulos.....	100
5.3.1	Núcleo coordinador	100
5.3.2	Módulo interfaz enlace radio (SMS y RTC)	101
5.3.2.1	Estructuras y gestión de memoria.....	101
5.3.2.2	Características funcionales	102
5.3.2.3	Protocolo y Tramas Spiral.....	103

5.3.3	Módulos de interfaz para clientes, servidores y operaciones	107
5.3.4	Módulos de interfaz de aplicación.....	109
5.3.4.1	DDE.....	109
5.3.4.2	OPC.....	111
5.3.5	Módulo de operaciones	112
5.3.5.1	Configuración y estado de las comunicaciones.....	113
5.3.5.2	Visualización de tramas	114
5.3.5.3	Generación de estadísticas de comunicaciones	115
5.3.5.4	Gestión de variables	116
5.3.5.5	Registro y control de acceso.....	116
5.3.5.6	Generación de ficheros de parametrización	116
5.4	Aplicación al abastecimiento de agua de la ciudad de Granada	118
5.5	Medidas y pruebas experimentales.....	119
5.5.1	Resultados de simulaciones.....	119
5.5.2	Resultados en aplicaciones reales.....	120
5.5.2.1	Red de alta.....	120
5.5.2.2	Red de baja.....	122
5.5.2.3	Sondeos de emergencia	123
5.6	Conclusiones	124
 CAPÍTULO 6: CONCEPCIÓN Y DISEÑO DEL SISTEMA SPIRAL-5000		125
6.1	Introducción.....	126
6.2	Descripción general del sistema.	127
6.3	Módulo central SP-5000.....	129
6.3.1	Consideraciones de diseño.	129
6.3.2	Conexionado y configuraciones hardware	130
6.4	Módulos básicos de ampliación.....	132
6.4.1	Consideraciones generales	132
6.4.2	SP-16ED: 16 Entradas Digitales Optoacopladas.....	134
6.4.3	SP-8REL: 8 Salidas de Relés	135
6.4.4	SP-DA: 8 Salidas Analógicas.....	136
6.4.5	Comunicación <i>peer-to-peer</i> entre módulos	138
6.5	Software de control de la SP-5000	138
6.5.1	Esquema de funcionamiento	139
6.5.2	La clase SP5000	140
6.5.3	La organización de la memoria	142
6.5.4	Gestión de las comunicaciones.....	142
6.5.5	Manipulación de los elementos de E/S.....	143
6.5.6	Los manipuladores de palabras	144
6.5.7	Programación de aplicaciones	145
6.6	Conclusiones	147

CAPÍTULO 7: DISEÑO DE SOFTWARE PARA CONTROL Y SUPERVISIÓN	149
7.1 Introducción.....	150
7.2 Consideraciones generales sobre el SCADA utilizado	151
7.3 Diseño de una aplicación SCADA. Caso de la gestión del agua en la ciudad de Granada... 152	
7.3.1 La programación dirigida a eventos	152
7.3.2 Comunicación con el exterior.....	155
7.3.3 Diseño gráfico de la aplicación	156
7.3.4 Tratamiento de las alarmas	160
7.3.5 Los históricos.....	162
7.3.6 Generación de informes.....	163
7.3.7 Nivel de configuración	165
7.3.8 Control de acceso	166
7.3.9 Ayudas al operador.....	167
7.3.10 La aplicación en números	169
7.4 Aplicaciones externas	172
7.5 Conclusiones.....	173
CAPÍTULO 8: OPTIMIZACIONES EN LA RED DE DISTRIBUCIÓN DE AGUA	175
8.1 Introducción.....	176
8.2 Objetivos del control óptimo	177
8.3 Consumo de energía en grupos de bombeos..... 178	
8.3.1 Introducción y objetivos	178
8.3.2 Criterios de optimización	179
8.3.3 Pantallas y controles diseñados	181
8.3.4 Consideraciones sobre la implementación	183
8.3.5 Resultados obtenidos	185
8.4 Sectorización de la ciudad..... 187	
8.4.1 Introducción y objetivos	187
8.4.2 Pantallas y controles diseñados	187
8.4.3 Resultados obtenidos	190
8.4.4 Líneas actuales y futuras derivadas del sistema	191
8.5 Control de reductoras. 193	
8.5.1 Introducción y objetivos	193
8.5.2 Pantallas y controles diseñados	194
8.5.3 Consideraciones sobre la implementación	196
8.5.4 Resultados obtenidos	198
8.6 Predicción del volumen de agua diario..... 199	
8.6.1 Introducción y objetivos	199
8.6.2 Análisis de la demanda y selección de las variables.....	199
8.6.3 Elección de la herramienta de cálculo	202
8.6.4 Resultados obtenidos	204

8.7	Regulación automática de los depósitos de la red de alta.	205
8.7.1	Introducción y objetivos	205
8.7.2	Descripción del procedimiento NPD propuesto	206
8.7.2.1	Descripción del algoritmo genético utilizado	208
8.7.2.2	La función coste futuro óptimo (<i>cost-to-go</i>)	210
8.7.3	Aplicación al control de la RAD	211
8.7.3.1	Función de coste para el algoritmo genético	213
8.7.3.2	Solución al problema de optimización	215
8.7.4	Resultados Obtenidos	216
8.7.5	Líneas actuales y futuras de investigación	219
8.8	Conclusiones	222
 CAPÍTULO 9: CONCLUSIONES Y LÍNEAS DE INVESTIGACIÓN FUTURAS		225
 APÉNDICE A: LISTADOS DE PROYECTOS		229
 APÉNDICE B: GLOSARIO DE TÉRMINOS		235
 REFERENCIAS BIBLIOGRÁFICAS		

Capítulo 1:

Introducción

Según informe de la ONU la escasez de agua es, junto a la contaminación, el problema medioambiental más grave. Los caprichos de la meteorología, la degradación de los recursos y el crecimiento demográfico han convertido al agua en un bien muy preciado. Lo que tiende a considerarse sólo una emergencia estival, podría convertirse en una catástrofe para el futuro del planeta. Y según los últimos datos, más de 200 millones de personas están seriamente afectadas por este problema, que no sólo afecta a los países en desarrollo, ya que más del 60% de las grandes ciudades europeas desaprovechan sus recursos.

Comenzamos este capítulo hablando sobre este problema y sobre las necesidades que se quieren satisfacer con este trabajo (sección 1.1). A continuación se hace una descripción de los objetivos a conseguir (sección 1.2). Y por último, se explica la organización que se ha seguido en la realización de esta memoria (sección 1.3), que pretende reflejar el esfuerzo que se ha realizado durante los años que hemos trabajado tanto en la Universidad como en la empresa privada, y que ha servido no sólo para obtener los resultados que se muestran en esta memoria, sino también, para intentar aportar un grano de arena en el equilibrio entre la capacidad investigadora demostrada en nuestro país (en cuanto a volumen de publicaciones científicas) y el desarrollo tecnológico del mismo, y en concreto de nuestro entorno local.

El capítulo en general versa sobre dos ideas fundamentales: la importancia de un recurso natural como es el agua y la integración vertical como metodología global de automatización que se ha de aplicar como solución a una demanda tan compleja, y a la vez tan desconocida, como es el abastecimiento de agua en grandes ciudades.

1.1 Planteamiento del problema y necesidades

El desarrollo de un país, y su grado de modernidad, depende de una manera directa de las infraestructuras de que dispone. Y de entre el colectivo de servicios urbanos e interurbanos que se agrupan bajo el genérico nombre de “infraestructuras”, la parte referente al ciclo integral del agua (ver Capítulo 3 de esta memoria), constituye sin duda uno de los bloques más importantes.

Aunque la inmensa mayoría de la población española dispone de servicios de agua potable en su vivienda, no es menos cierto que con relativa frecuencia aparecen en los medios de comunicación referencias a la necesidad de modernizar estos servicios. Averías que dejan sin suministro a más de 500.000 personas durante períodos superiores a las 24 horas, reclamaciones por insuficientes niveles de presión en las redes de distribución, protestas por la poca calidad del agua, etc., son noticias que a veces aparecen en los medios de comunicación de nuestro país.

Por otra parte, el problema de la sequía, cuyo dramatismo máximo se alcanza cuando el problema puede llegar a afectar al suministro de agua potable, se manifiesta periódicamente en muchas áreas geográficas de España.

Por tanto, está claro que el agua es un bien escaso que es necesario explotar de forma correcta. *“Las temperaturas, la explosión demográfica, la contaminación y un incorrecto uso han conducido al agua, el más imprescindible de nuestros recursos, a una crisis tan importante que es el problema más grave con el que se enfrentará el hombre en el segundo milenio”*.

Así pues, y según los comentarios precedentes, consideramos que es esencial obtener un sistema automático de operación en tiempo real para redes de abastecimiento y saneamiento de agua, que busque la optimización de la explotación de este recurso en cuanto a funcionalidad, calidad de servicio y costes.

En este sector y sobre todo en la parte correspondiente a las redes de distribución de agua, así como en otros campos científicos, técnicos e industriales, es necesario realizar medidas periódicas en tiempo real de ciertos parámetros y actuar sobre elementos de control. Esto no siempre es inmediato, especialmente cuando el control y la observación es distante o de difícil acceso. La tecnología actual permite el control y medida de procesos a distancia mediante sistemas enlazados a través de una red de comunicaciones con una estación central bajo supervisión o no de un operador humano.

Con este tipo de sistemas y desde un punto funcional se pretende conseguir:

- Un aumento de los márgenes de seguridad de la instalación.
- Un ahorro en los costes de producción, optimizando el consumo de energía y disminuyendo los gastos de personal.
- La mejora de la calidad del producto.
- Una mayor transparencia en la gestión.
- Una mayor flexibilidad en el diseño, para futuras ampliaciones y modificaciones de los elementos que intervienen en los sistemas de gestión, planificación y control de los procesos.

Nuestro objetivo no se limita a diseñar un simple sistema de supervisión y transferencia de información, sino que se pretende conseguir una herramienta imprescindible para lograr una explotación óptima. En las redes de abastecimiento de agua se presentan ciertos problemas comunes que son necesarios resolver u optimizar: ahorro energético, utilización de los recursos humanos, detección de fugas, control automático de las presiones de inyección, distribución automática del suministro, predicciones de consumo, etc. Como es lógico, para poder afrontar estas necesidades se necesita un alto nivel de integración que contemple soluciones a dichos objetivos.

Se propone una nueva metodología para el desarrollo de técnicas y elementos que contribuyan al diseño, integración y optimización de sistemas de control distribuido de redes de abastecimiento de agua en grandes ciudades. Los resultados concretos de la materialización del trabajo realizado corresponden a la automatización completa de todo el ciclo integral del agua de la ciudad de Granada.

1.2 Objetivos

El objetivo principal de esta tesis es el diseño de los diferentes niveles de integración de un sistema completo para la gestión del agua de grandes ciudades, para conseguir una explotación óptima de este recurso natural, tal y como se ha planteado en el apartado anterior.

En primer lugar se estudiarán las especificaciones funcionales de los sistemas de abastecimiento y saneamiento de agua, para determinar las necesidades reales de control, con el objetivo de conseguir una arquitectura de red de comunicaciones y dispositivos de control adecuada, y a la vez completamente flexible y extensible a otros procesos de telemedida y telecontrol de características similares.

Para ello tendremos que diseñar nuevas soluciones y elementos de hardware, software y comunicaciones e integrarlas con otras ya existentes.

Además, se hará un estudio para abordar problemas tradicionales en este sector con las nuevas herramientas disponibles hoy día para el tratamiento de la información (redes neuronales, algoritmos genéticos, lógica borrosa, etc.), con los objetivos de conseguir mejores resultados y/o facilitar su programación, y dejar una puerta abierta para futuras líneas de investigación.

Para el desarrollo del trabajo hemos establecido una estrecha cooperación con la empresa ICR (Ingeniería y Control Remoto S.A.) y la empresa responsable de la gestión del agua de la ciudad de Granada, EMASAGRA (Empresa Municipal de Abastecimiento y Saneamiento de Agua de la ciudad de Granada), integrada en el grupo AGBAR (Sociedad General de Aguas de Barcelona).

1.3 Estructura de esta memoria

En esta última sección se hace una breve descripción de cada uno de los capítulos de los que consta la presente memoria, así como de su estructuración.

Los fundamentos y la descripción del problema se tratan en el Capítulo 2 (**Fundamentos de la automatización de procesos**) y en el Capítulo 3 (**El ciclo integral del agua**), respectivamente. En el primero se da una visión general de los fundamentos y herramientas utilizadas: modelo de fabricación integrada por computador (*CIM*), sistemas de automatización (*PLC*), software de supervisión y control de procesos industriales (*SCADA*), y una breve descripción de las nuevas tendencias y herramientas aplicadas a la optimización de procesos (*AGs*, *RNAs*, *lógica difusa*, *programación dinámica*). En el segundo capítulo de esta primera parte, se hace una descripción del Ciclo Integral de Agua en general, y en particular para la ciudad de Granada, con el fin de dar una visión más amplia del problema que se pretende resolver.

En el Capítulo 4 (**Sistema de control para la red de distribución de agua de la ciudad de Granada**), se muestra en conjunto todo el trabajo realizado, y en especial los aspectos relacionados con el equipamiento de control y comunicaciones.

A continuación, el Capítulos 5 (**Gestor de comunicaciones distribuidas**), el Capítulo 6 (**Concepción y diseño del sistema Spiral-5000**), el Capítulo 7 (**Diseño de software para supervisión y control**) y el Capítulo 8 (**Optimizaciones en la red de distribución de agua**), están dedicados a profundizar en las principales aportaciones que se presentan en esta memoria, como son la gestión de comunicaciones, el desarrollo de un nuevo sistema de control distribuido, el diseño de una metodología *SCADA*, y un conjunto de herramientas y procedimientos de optimización para la explotación de redes de abastecimiento de agua potable, respectivamente.

Y finalmente, en el Capítulo 9 (**Conclusiones y líneas de investigación futuras**), se resumen cuales son las principales aportaciones de la tesis, indicando las conclusiones finales que pueden extraerse como síntesis de los resultados obtenidos. Incluimos además las líneas futuras de investigación a desarrollar en el Departamento de Arquitectura y Tecnología de Computadores de la Universidad de Granada como continuación de los resultados conseguidos en este trabajo.

En la Figura 1.1 se muestra un esquema descriptivo de los contenidos de estos capítulos y su ubicación en cuanto a los objetivos que pretenden cumplir.

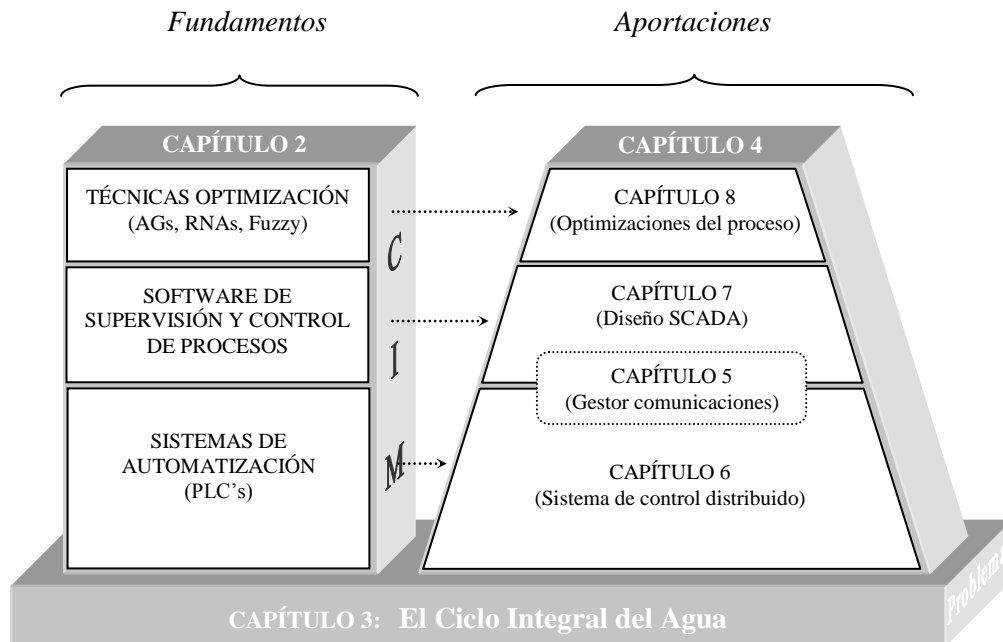


Figura 1.1: Cumplimiento de los objetivos a lo largo de los capítulos de esta memoria

El Apéndice A proporciona una lista completa de los proyectos en los que se ha utilizado la metodología y los sistemas descritos en esta memoria. Y el Apéndice B, donde se recopilan los términos o acrónimos utilizados, se ha introducido con la finalidad de facilitar la lectura de la memoria.

Por último se incluyen las referencias bibliográficas utilizadas para la realización de este trabajo. Para ello se han utilizado “formas resumidas” consistentes en los tres primeros caracteres del nombre del autor seguidos de las dos últimas cifras del año de publicación del trabajo.

Capítulo 2:

Fundamentos de la automatización de procesos

El objetivo principal de la automatización industrial consiste en gobernar la actividad y la evolución de los procesos sin intervención continua del operador humano. Este objetivo, en procesos de poca variación en el tiempo o de carácter autónomo, se cumple programando las secuencias de control deseadas sobre los controles locales de planta y cerrando los lazos de regulación necesarios para mantener los valores de variables en los rangos fijados por las consignas.

No obstante, la mayor parte de procesos industriales no cumplen estas condiciones, sino más bien las contrarias: han de ser *flexibles* y muy *interrelacionados* entre sí, por exigencias de factores no sólo específicos de la producción, sino también por otros como la minimización de costes, la calidad integral del producto y el impacto sobre el medio ambiente.

Estas necesidades obligan a disponer de sistemas automatizados de control de procesos industriales con un alto grado de complejidad y con funciones adicionales a las básicas de monitorización y control del proceso. Aspectos como la generación de históricos, gestión de alarmas, control de calidad y mantenimiento, etc., así como la toma automatizada de decisiones y optimizaciones (que se trata con más detalle en la sección 2.4 de este capítulo), quedan cubiertos en los niveles de control de planta y de supervisión del modelo jerárquico de integración (ver *Pirámide CIM* en la siguiente sección).

En un proceso típico, como es el caso de la red de abastecimiento de agua de la ciudad de Granada, el control directo de planta se realiza mediante sistemas de control distribuido (SCD), como por ejemplo SP-5000 (ver Capítulo 6), y/o por los controladores lógicos programables (PLC), sistemas que se introducirán en la sección 2.2, mientras que el computador, conectado con ellos, realiza las funciones de diálogo con el operador, tratamiento de la información del proceso, control de producción, y algoritmos de optimización.

Los programas necesarios, y en su caso el hardware adicional que necesiten, se denominan en general sistemas SCADA (*Supervisory Control And Data Acquisition*). Con esta herramienta se pueden desarrollar aplicaciones con captura de datos, análisis de señales, presentaciones en pantalla, envío de resultados a disco o impresora, control de actuadores, etc. Una introducción general y una descripción más completa sobre los sistemas SCADA se tratan en la sección 2.3 de este capítulo.

La metodología de integración vertical de todas estas tecnologías - controladores lógicos programables, sistemas de adquisición y control basados en microcontroladores, interfaces de comunicaciones, sistema SCADA y módulos para optimizaciones y toma de decisiones en tiempo real - es uno de los objetivos de este trabajo, y se ha implantado en el caso particular de la red de abastecimiento de agua de la ciudad de Granada.

2.1 Fabricación integrada por computador. El concepto CIM

Los procesos de producción requieren, cada vez más, un elevado nivel de automatización e integración con los sistemas de optimización y gestión, en lo que se conoce como Fabricación Integrada por Computador, *CIM (Computer Integrated Manufacturing)* [Sch-92, Bla-90], o *CIME (Computer Integrated Manufacturing & Engineering)* [Deb-93].

La automatización integral es el objetivo final a conseguir por cualquier compañía. Para ello, es imprescindible un sistema que permita la conversión de la gran cantidad de datos procedentes de las plantas de fabricación en una fuente de información más rica, con el fin de optimizar los procedimientos asociados a la producción.

Para aumentar la efectividad de los operadores de planta y del personal de gestión, esta fuente potencial de datos se debe interpretar, organizar y gestionar desde un *Sistema de Control de Procesos* [Bol-88]. Estos sistemas de explotación y control de procesos hacen uso de datos tanto capturados como generados en todo el proceso, y sustituyen al operador humano en la toma de datos y elaboración de decisiones en aquellas circunstancias, sin contar criterios económicos, en las que por las características del proceso ocurra que:

- La acción del hombre no sea lo suficientemente rápida o segura.
- La apreciación humana no sea suficiente para obtener datos precisos.
- La capacidad de procesamiento de información humana sea insuficiente.

Los sistemas de control industrial evolucionaron de los sistemas analógicos, pasando por el control centralizado por computador, hasta los sistemas de control distribuido (*SCD*).

Los *SCD* se basan en la incorporación de módulos independientes para la adquisición de datos y control local, de forma que el fallo en un determinado módulo no tiene necesariamente que anular la operatividad del sistema, mientras que los sistemas de control jerárquicos se desarrollan de forma natural a partir de las estructuras de dirección típicas de las empresas en las que cada elemento de decisión recibe órdenes desde el nivel que tiene por encima.

En los esquemas de sistemas de control actuales se utiliza una aproximación mixta entre los sistemas jerárquicos y los sistemas distribuidos. Las ventajas que caracterizan a los sistemas jerárquicos de control distribuido se pueden resumir en los siguientes puntos:

- Instalación y montaje simple.
- Operación más sencilla y segura.
- Distribución funcional y espacial de prestaciones.
- Se opera de acuerdo a una estructura jerarquizada.
- Permiten la implementación del concepto de control integral *CIM*.
- Admiten una mayor flexibilidad en configuraciones.
- Capacidad de autodiagnóstico y localización de averías.
- Mejor respuesta a alteraciones en el proceso.
- Fácil actualización de prestaciones mediante *software*.

Los requerimientos de comunicación pueden diferir sensiblemente dependiendo de la aplicación de los elementos que se han de comunicar, lo que obliga a la existencia de vías de comunicación horizontales a distintos niveles organizativos, que se conectan entre sí por medio de ciertos medios de interconexión de redes: puentes (*bridges*) y pasarelas (*gateways*).

En una aplicación típica de proceso productivo que siga la metodología *CIM*, se puede establecer un sistema jerárquico de comunicaciones [Sch-92], representado en la pirámide de integración a 4 niveles (Figura 2.1, en la cual se muestra además un ejemplo de control distribuido jerarquizado).

En determinadas circunstancias¹, para la comunicación entre los dos niveles inferiores de la pirámide de integración es aconsejable una conexión vía radio, y es en este contexto donde se enmarca el sistema SPIRAL [Gom-94]: definición de una arquitectura de red que permite el telecontrol vía radio de diferentes estaciones remotas desde una estación central. Este sistema, mejorado, y que se describe con más detalles en el Capítulo 4, es el que se ha utilizado en este trabajo para realizar el telecontrol de la red de abastecimiento de agua de la ciudad de Granada, debido fundamentalmente a las condiciones adversas en la ubicación de las estaciones remotas que presenta dicha aplicación.

¹ Distancias grandes, del orden de decenas o cientos de Km., puntos de difícil acceso para el tendido de cable, o cuando interesa libertad de movimientos entre el computador de supervisión y el sistema a controlar.

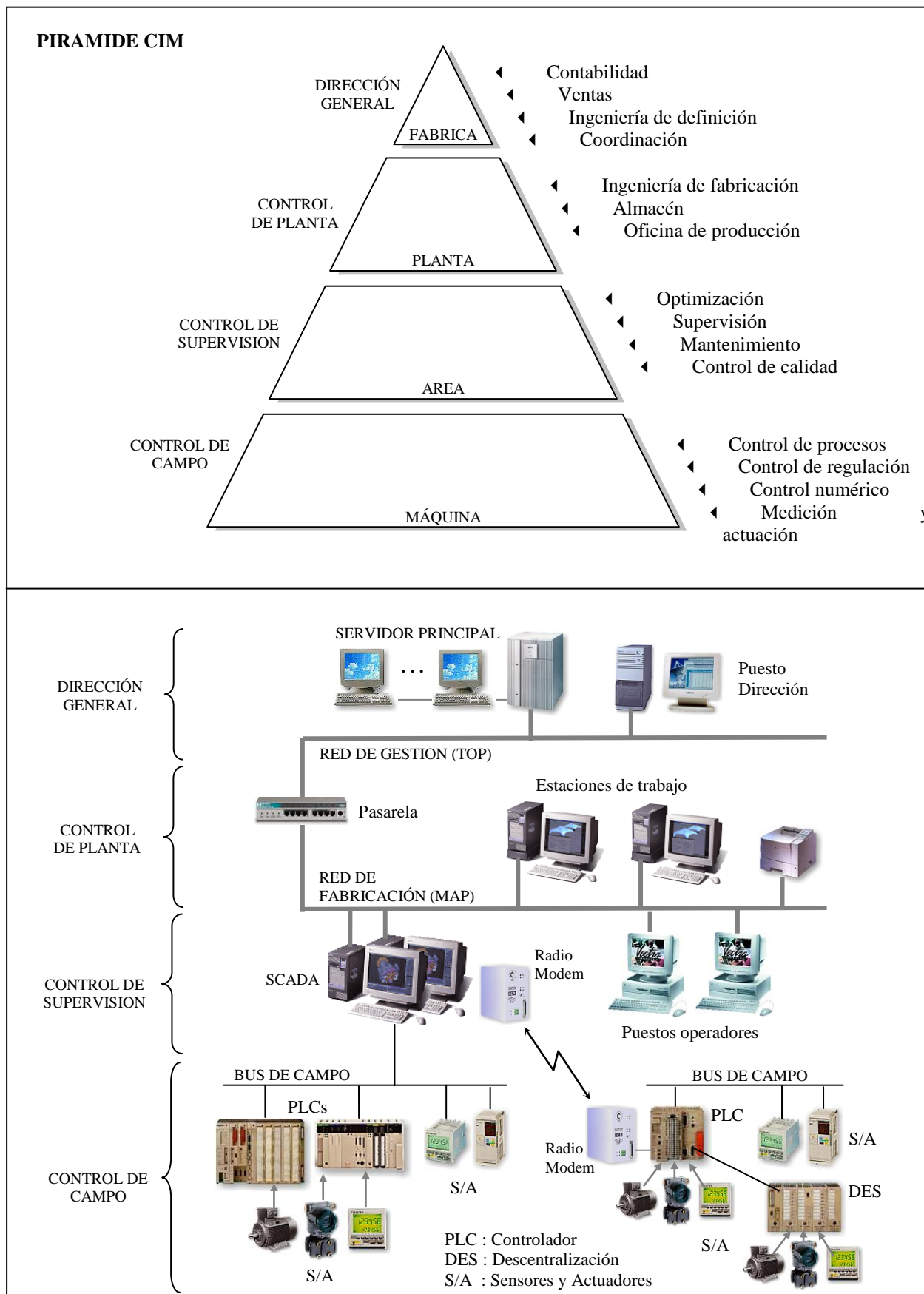


Figura 2.1: Pirámide CIM y ejemplo de la estructura de red de un proceso industrial

2.2 Sistemas de automatización

La aparición de nuevas tecnologías, sobre todo en los campos de la electrónica, informática y comunicaciones, constituye un constante motor en el desarrollo y sofisticación de los sistemas automáticos de control, provocando profundos cambios en la concepción y diseño de los procesos y en sus métodos de control.

Dichos cambios se suceden con tal rapidez que, incluso para los especialistas en ingeniería de control, requieren un continuo replanteamiento de los métodos de diseño empleados, a medida que se encuentran disponibles nuevos equipos y se amplían las posibilidades de interconexión entre ellos.

En concreto, respecto a la unidad de control, la disponibilidad de los autómatas programables o PLC² [Man-92, Bal-97, Cri-97], con una configuración adaptable prácticamente a cualquier necesidad y tamaño de instalación, hace que haya quedado descartado el empleo de sistemas lógicos o analógicos dedicados a funciones específicas. Esta evolución es fruto de la aparición de los circuitos integrados programables de uso general (microprocesadores, PLD, circuitos semipersonalizados y ASIC), que ofrecen un hardware estándar utilizable para una gran diversidad de aplicaciones.

Como alternativa a los controladores lógicos programables también se puede utilizar como unidad de control cualquier sistema basado en microcontrolador diseñado a medida para una aplicación o proceso dado. Este es el caso del sistema SP-5000, que se ha utilizado, además de los autómatas programables, como unidad de adquisición y control en ciertas estaciones remotas en el sistema de abastecimiento de agua de la ciudad de Granada. Por tanto, en esta memoria y con relación a los sistemas de automatización, se han cubierto las dos alternativas principales existentes hoy en día para controlar un proceso de forma automática. La concepción y el diseño del sistema SP-5000 se trata con detalle en el Capítulo 6 de esta memoria.

² PLC = *Programmable Logic Controller*. Controlador Lógico Programable.

2.2.1 Introducción a los PLC

Un controlador lógico programable (PLC) o autómatas industrial (AI) es un equipo electrónico de control con un cableado interno (*hardware*) independiente del proceso a controlar, que se adapta a dicho proceso mediante un programa específico (*software*) que contiene la secuencia de operaciones a realizar.

Esta secuencia de operaciones se define sobre señales de entrada y salida al proceso, cableadas directamente a los bornes de conexión del PLC. En la Figura 2.2 se muestran los bloques esenciales de un autómatas programable, donde la unidad central, basada en un sistema microprocesador, es la encargada de ejecutar el programa de usuario almacenado en la memoria de programa, y ordenar las transferencias de información entre las interfaces de entrada/salida y la memoria imagen o de trabajo.

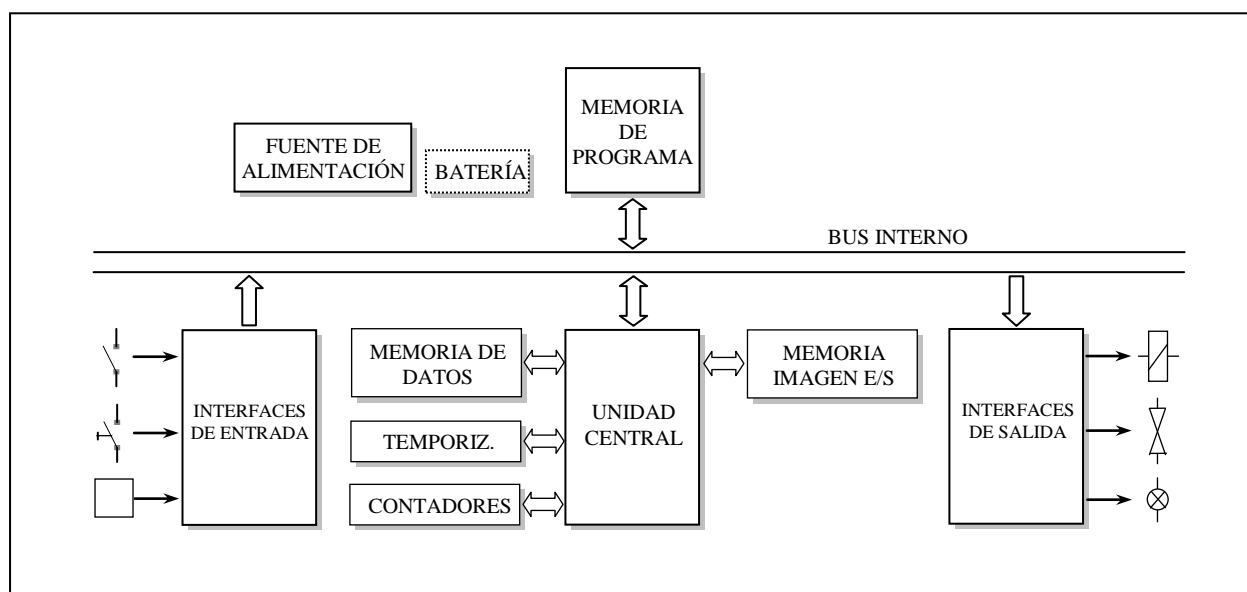


Figura 2.2: Diagrama de bloques de un controlador lógico programable

En cuanto al funcionamiento de un PLC, salvo el proceso inicial que sigue a un *reset*, es de tipo secuencial y cíclico, es decir, las operaciones tienen lugar una tras otra y se van repitiendo continuamente mientras el sistema disponga de alimentación eléctrica (ver Figura 2.3).

El tiempo total que el PLC emplea para realizar un ciclo de operación se llama tiempo de ciclo (“*Scan Time*”), y depende del número de entradas/salidas involucradas, la longitud del programa de usuario, y el número y tipo de periféricos conectados al PLC. Este tiempo habrá que

tenerlo en cuenta, ya que dependiendo de los tiempos de reacción del proceso a controlar será o no aconsejable utilizar un PLC.

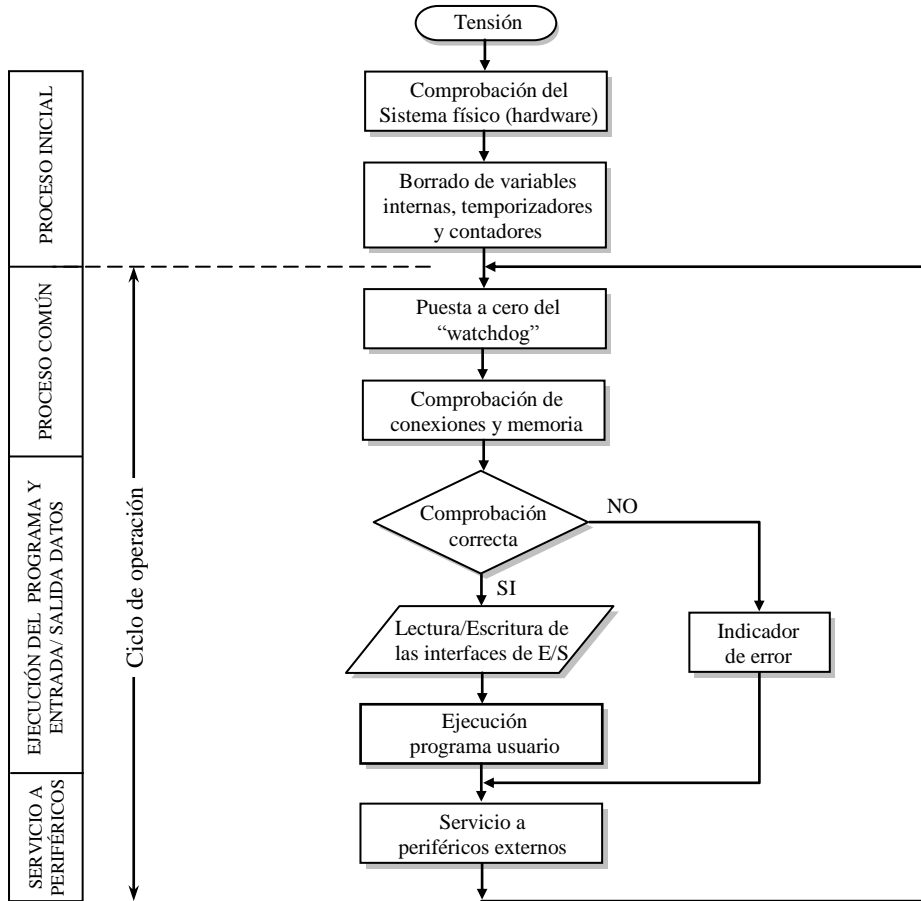


Figura 2.3: Ciclo de funcionamiento de un controlador lógico programable

2.2.2 Programación de PLC

Una de las ventajas de la utilización de los PLC está relacionada con su programación. Se puede hacer una traducción directa del lenguaje de contactores al lenguaje del controlador, y así facilitar la implementación de automatismos a la comunidad de técnicos provenientes del campo de la lógica de contactores y electrónica. O utilizar otro tipo de lenguajes provenientes del propio campo teórico de los PLC y definidos tradicionalmente mediante grafos, como por ejemplo el GRAFCET³.

³ GRAFCET = GRÁfico Funcional de Control de Etapas y Transiciones, es una herramienta de diseño para procesos secuenciales, de relevante difusión, que se inspira en un tipo de modelo matemático de autómatas denominado Redes de Petri.

Una interesante iniciativa de normalización en la programación de PLCs es la especificación IEC 1131-3 [Iec-93], donde se describe la sintaxis y la semántica de una serie de lenguajes de programación, tanto literales como gráficos, para estos dispositivos.

Como ejemplo de lenguaje de programación utilizado por los PLC, a continuación se describe de forma muy resumida la estructura de programación modular que propuso Siemens⁴ con su lenguaje STEP5 [Sie-91]; software que ha impulsado a nivel mundial la programación estructurada de autómatas programables, y que ha jugado un papel importante en la creación de la norma IEC 1131-3.

La Figura 2.4 muestra como STEP 5 utiliza una estructura modular abierta, que permite enlazar distintos tipos de módulos:

- *Módulos de organización (OB)*: fijan la estructura de ejecución del programa, y son llamados automáticamente por el monitor del sistema.
- *Módulos de programa (PB)*: contienen las instrucciones del programa de mando, son llamados desde los módulos de organización mediante instrucciones de salto condicional o incondicional, y permiten la anidación entre sí.
- *Módulos funcionales (FB)*: o subrutinas parametrizables que constituyen funciones cerradas del programa de aplicación. Pueden ser programados por el fabricante, que los oferta como productos software, o por el usuario, que los define para su aplicación y los almacena en bibliotecas.
- *Módulos de datos (DB)*: contienen constantes alfanuméricas, con posibilidad de almacenarlas en distintos formatos, que se pueden utilizar en el programa.

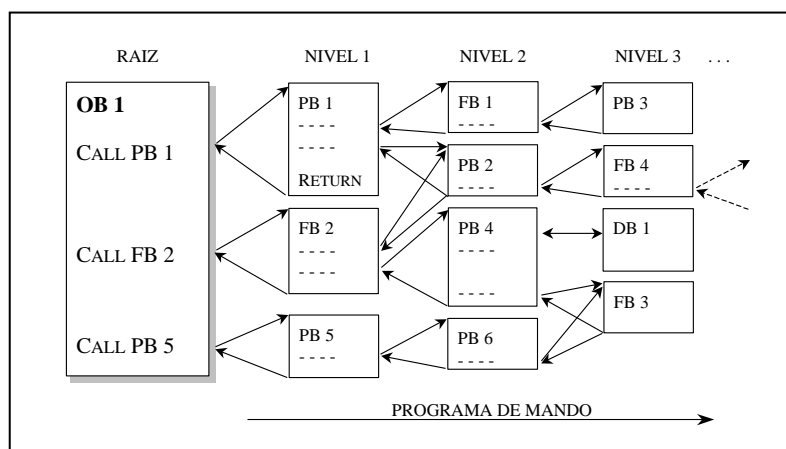


Figura 2.4: Ejemplo de programación estructurada con PLC

⁴ Uno de los fabricantes más importantes y reconocidos de autómatas programables.

2.3 Software para la supervisión y el control de procesos

2.3.1 Introducción

En la mayoría de los sistemas de automatización intervienen diferentes tipos de elementos, tales como actuadores y sensores, controladores, computadores o software, por citar los más habituales. La efectividad de una solución viene dada por el resultado de la integración de todos ellos y, por lo tanto, cada elemento contribuye a la funcionalidad global y condiciona la efectividad del conjunto; de ahí la importancia de utilizar también un software adecuado.

Las aplicaciones SCADA constituyen una de las soluciones más ampliamente adoptadas para la gestión y control de los procesos industriales. El acrónimo “SCADA” se corresponde con *Supervision, Control And Data Acquisition* [Wil-92, Ayz-98, Ayz-99], y se refiere a los sistemas que realizan la supervisión y control de procesos industriales, a la vez que capturan y almacenan los valores que van tomando en el tiempo todas las variables de estado del proceso.

Estas son las funciones básicas de tales sistemas, aunque habitualmente suelen incorporar más funciones accesorias, pero no menos útiles o importantes. Estas funciones accesorias incluyen entre otras cosas: generación de informes a partir de datos históricos, generación de gráficos de evaluación del proceso (también llamados sinópticos), gestión de copias de seguridad, control de acceso de usuarios, etc.

Frecuentemente se asocia al término SCADA la parte de software del sistema, pero en realidad es un concepto más amplio que comprende también al equipo físico y a la red de comunicaciones que permite la transmisión de la información en ambas direcciones.

Se entenderá como *aplicación SCADA* el diseño de los elementos gráficos, su interacción con el sistema y el usuario, las variables monitorizadas, las alarmas y su gestión, los históricos, y la configuración de todas aquellas otras posibilidades que el software permita, aplicadas al sistema en concreto. Es decir, la aplicación SCADA es lo que los usuarios del sistema verán, puesto que se definen los sinópticos y los elementos visibles que permiten a los operarios interactuar con el proceso.

El *software SCADA* es el conjunto de software que soporta la aplicación, y que realiza las tareas de gestión de los gráficos, datos, mensajes y órdenes del usuario.

Los paquetes de software SCADA suelen estar formados básicamente por dos programas: Editor y Ejecutor (*Run-Time*). Con el primero se generan las aplicaciones descritas, aprovechando los editores, macros, lenguajes y ayudas disponibles, y con el segundo se compilan para obtener el fichero de ejecución continua. En la Figura 2.5, se puede observar el diagrama de contexto general para este tipo de software.

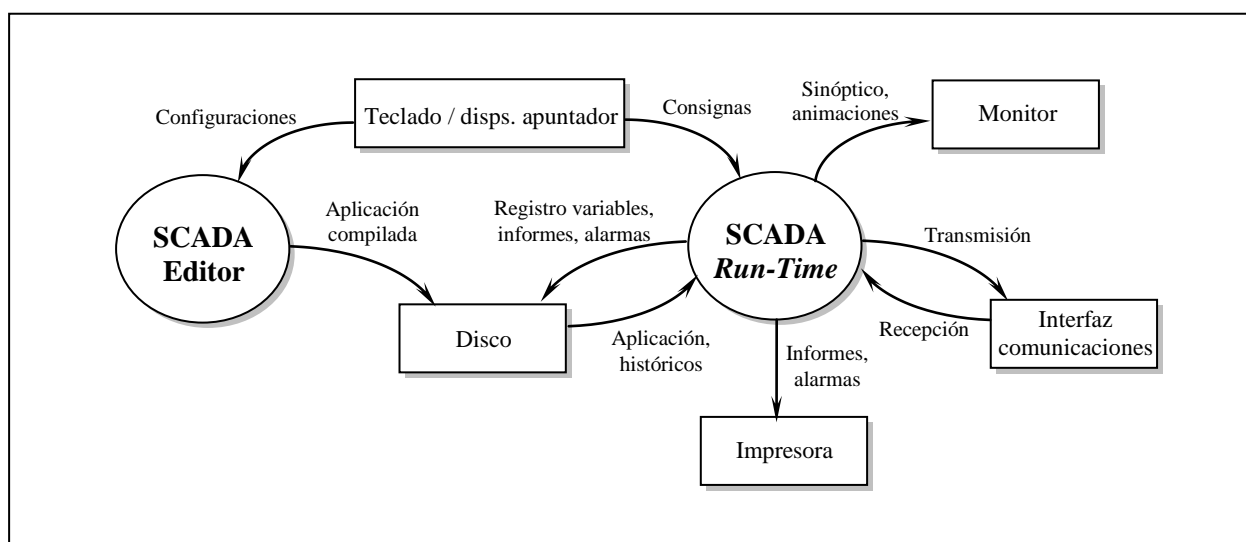


Figura 2.5: Diagrama de contexto general de un SCADA

El *sistema SCADA* es todo lo anterior (aplicación + software) junto con el equipo físico (hardware = computadores + sistemas de comunicación) que sirve para cumplir las funciones de supervisión, control y adquisición de datos.

Estas definiciones se conservan en todos los casos, aunque en ocasiones resulta complicado definir la línea que separa a la *aplicación* de este *software SCADA*, ya que dicha línea ni siquiera existe puesto que los programas que constituyen el software se crearon para esa aplicación. Sin embargo, lo habitual es utilizar un paquete de programas de uso genérico y diseñar una aplicación basada en dicho software configurable, por sus ventajas en cuanto a escalabilidad, modularidad y autonomía.

La metodología de integración de procesos que se describe a lo largo de esta memoria utiliza una herramienta SCADA comercial. En el Capítulo 7, como ejemplo de aplicación SCADA, se describe con detalle el diseño del sistema implementado para la red de abastecimiento de agua de la ciudad de Granada.

2.3.2 Conceptos generales sobre SCADA

Además de las características básicas ya descritas en la introducción anterior, para poder hacer un buen diseño del sistema, hay que tener claros una serie de conceptos y procedimientos que pueden estar o no soportados por el paquete de software utilizado.

Un SCADA debe cumplir varios objetivos para que su instalación sea perfectamente aprovechada:

- Debe ser un sistema abierto, capaz de crecer o adaptarse según las necesidades cambiantes de la empresa.
- Debe comunicarse con total facilidad y de forma transparente para el usuario con el equipo de planta (*driver* de comunicación con PLC, PID, CN, etc.), y con el resto de la empresa (acceso a redes locales y de gestión).

De esta forma el SCADA se convierte en una herramienta esencial para organizar el proceso productivo, permitiendo establecer estrategias avanzadas de gestión con respuestas inmediatas de la producción. Además, la aplicación constituye por sí misma una excelente herramienta de integración entre los diferentes departamentos involucrados en la fabricación, desde producción a gestión pasando por calidad, mantenimiento e ingeniería [Ayz-99].

Como soporte hardware de estos programas se utilizan normalmente computadores convencionales, desde computadores personales (PC) hasta estaciones de trabajo (en ocasiones pueden disponer de alguna protección adicional para ambientes industriales). Aunque pueden emplearse arquitecturas basadas en PC con sistema operativo Windows, la disponibilidad de máquinas con sistemas operativos más completos (Unix, Windows NT, VAX/VMS, ...) y arquitecturas cliente-servidor que comparten recursos informáticos (aplicaciones y datos), permite utilizar programas que atiendan varios servicios a la vez. Así, un operador puede estar monitorizando la situación en la planta desde un terminal PC, otro viendo informes de incidencias desde un computador industrial con sistema operativo Unix, o modificando la evolución del proceso desde una estación de trabajo, etc. En aplicaciones muy grandes se utilizan estas arquitecturas cliente-servidor para distribuir los datos procesados entre diferentes computadores y así reducir la carga de cada uno de ellos. Por tanto, en la elección del sistema, un factor clave a considerar es la capacidad del sistema operativo sobre el que se ejecuta la aplicación para soportar multitarea y/o multiusuario.

En la actualidad, para aplicaciones de mediana y baja complejidad sigue siendo mayoritario el empleo de equipos basados en PC, con una CPU tanto más potente cuanto más críticos sean los tiempos de respuesta necesarios, y configuraciones tanto más robustas cuanto más duras sean las condiciones ambientales de aplicación.

La comunicación con los elementos de campo se lleva a cabo mediante interfaces serie normalizados, tipo RS-232, RS-422 o RS-485, utilizando los protocolos adecuados incluidos en el propio SCADA. De hecho, el desconocimiento del protocolo de un determinado equipo, o su incompatibilidad con las normas aceptadas, puede impedir su conexión al sistema.

Otro dato importante a considerar es la cantidad de sinópticos (pantallas gráficas de representación) que el sistema puede soportar, así como el número máximo de variables que puede manipular.

También son factores a considerar la capacidad de intercambio de datos con otros programas o entornos, para integrar las funciones de éstos dentro de la aplicación (por ejemplo cálculos estadísticos, presentaciones y optimizaciones más complejas), y la posibilidad de programación de funciones a medida mediante la inclusión en el SCADA de rutinas y ficheros escritos en lenguajes de uso general, como C o Pascal.

2.3.2.1 Entorno de desarrollo

Todos los paquetes SCADA son programas que tienen una doble finalidad. Por un lado, son generadores de aplicaciones de supervisión y control, y son utilizados por ingenierías y departamentos de desarrollo para generar la aplicación concreta a cada proceso. Por el otro, son aplicaciones que trabajan para realizar las funciones de supervisión y control del proceso, y donde los usuarios son los operadores y supervisores.

Tanto para las ingenierías como para los departamentos de desarrollo o de mantenimiento de aplicaciones que deben editar las aplicaciones, es importante analizar las utilidades que ofrecen los entornos de desarrollo de cada software SCADA.

Un primer aspecto a considerar son los lenguajes de programación disponibles. Muchos programas incorporan un conjunto de instrucciones específico (lenguaje propietario), mientras que otros disponen de interfaces de programación de aplicaciones (API), para poder diseñar funciones adicionales programadas con lenguajes estándar como C++, Visual Basic, Delphi, etc.

Dependiendo del software SCADA, es posible ejecutar conjuntos de órdenes o comandos (*scripts*) que realizan ciertas tareas relacionadas con los diversos elementos del SCADA, tales como manipulación de objetos gráficos de un sinóptico, cálculos, generación de gráficos o informes, gestión de alarmas, realización de copias de seguridad, etc.

Dentro del módulo de desarrollo o de configuración el usuario también define las pantallas gráficas o de texto que va a visualizar, importándolas desde otra aplicación o generándolas desde el propio SCADA. Para ello, éste incorpora un editor gráfico que permite dibujar a nivel de pixel o utilizar elementos ya disponibles como líneas, círculos, textos o figuras, con funciones de edición típicas como copiar, mover, borrar, etc. Dibujadas o seleccionadas las pantallas se definen las relaciones entre ellas, que determinan el orden de aparición y el enlace entre unas y otras, su accesibilidad a operarios generales o particulares, etc. El mantenimiento de las pantallas es sencillo dado que cada una lleva asociadas sus propiedades configurables.

También durante la configuración se seleccionan las interfaces de comunicaciones (*drivers*) que permitirán el enlace con los elementos de campo y la conexión o no en red de estos últimos, y se selecciona el puerto de comunicación sobre el computador y los parámetros de la misma. En algunos sistemas es también en la configuración donde se indican en forma de lista o tabla las variables que después se van a visualizar, procesar o controlar.

Otro aspecto importante consiste en la disponibilidad de utilidades de simulación y depuración cómodas, potentes y eficaces, que permitan desarrollar y hacer la puesta a punto de una aplicación en poco tiempo y con seguridad.

Y una característica de interés para el mantenimiento de aplicaciones es lo que se conoce como la capacidad de configuración en línea, que consiste en la posibilidad de realizar modificaciones en una aplicación activa, e incorporarlas directamente sin necesidad de detener la aplicación y relanzarla posteriormente.

2.3.2.2 Adquisición de datos del proceso

Cuando se necesita leer el valor de una magnitud de una estación remota es necesario indicar donde poder encontrar dicho valor numérico.

Para ello, al configurar la aplicación SCADA, se han de definir los nombres, tipos, canal y parámetros de comunicación concernientes a dicha estación remota, entre otras cosas. Si se

supone una comunicación vía radio con las estaciones remotas, el computador deberá generar la trama con la dirección del PLC y los comandos pertinentes, enviarlo al radiomodem, y después el PLC cuya dirección coincida con la de la trama enviada, recogerla, procesarla, reunir la información necesaria, generar su trama, y enviarla de vuelta.

Otro parámetro que se debe poder configurar en la aplicación es el periodo de comunicación con las estaciones remotas. Existen varios modos de muestreo, en relación con el momento y su frecuencia, que dependerán del software SCADA con que estemos diseñando la aplicación. Este es un punto importante a la hora de evaluar y decidir la herramienta de software SCADA a seleccionar, aunque hay que tener presente que en muchos casos estos modos de lectura dependen más del *driver* de comunicaciones utilizado que del propio SCADA.

Muestreo temporal. Este es el tipo de muestreo más común. Se fija un tiempo de muestreo (o una frecuencia); es decir, cuánto tiempo hay entre lectura y escritura. Este tiempo o frecuencia de muestreo puede variar en tiempo de ejecución, si el software (*driver* o SCADA) lo permite, adaptándose a las necesidades de tiempo según la información que se esté visualizando o que se necesite almacenar en cada momento.

Muestreo por interrupción. Como las comunicaciones en modo de muestreo temporal pueden ser completamente ineficientes para ciertos casos, se dispone del muestreo por interrupción en el cual la estación remota es la que toma la iniciativa de la comunicación, transmitiendo al puesto de control central una señal de interrupción para que este establezca contacto de modo inmediato con la estación remota y le transfiera la información que se considera de alta prioridad. De esta forma, no es necesario esperar a que el SCADA establezca contacto con la remota para que esta le pueda mandar la información.

Muestreo por evento: En este caso la comunicación se establece cuando sucede algo: alguna variable alcanza un cierto valor fijado, un conjunto de variables satisfacen alguna condición, el operador realiza alguna acción, se produce alguna alarma, etc.

2.3.2.3 Gestión de sinópticos

La gestión de sinópticos comprende todas aquellas operaciones relacionadas con los objetos gráficos que representan el estado de un proceso. Todos estos objetos gráficos forman un

sinóptico, o representación esquemática más o menos detallada, que tiene como finalidad proporcionar al operador un conjunto de ventanas de visualización con unos elementos y un modo de operación que le resulten lo más familiares e intuitivos posible, y que al mismo tiempo proporcionen un entorno de trabajo robusto, ágil y seguro.

En muchos casos el software SCADA viene acompañado de librerías gráficas genéricas predefinidas, que permiten la composición de sinópticos rápidamente. No obstante, ello no es un impedimento para que el desarrollador de la aplicación pueda definir otros objetos con el propósito de conseguir adaptarlos al estilo general de la aplicación, facilitar al operador su utilización y dotarlos de mayor funcionalidad.

Estas librerías gráficas también incluyen elementos de mando tales como botoneras, pulsadores, selectores rotativos y deslizantes, visualizadores analógicos de aguja y barra, y algunos otros más, lo que permite implementar auténticos paneles de mando virtuales con los que manipular el proceso sin requerir el montaje de un panel de mandos físico, con la ventaja que esto supone en cuanto a facilidad y bajo coste de modificación y/o ampliación.

Los objetos gráficos que se utilizan en los sinópticos pueden ser vectoriales, es decir definidos a partir de líneas, polígonos y circunferencias, o bien imágenes en mapa de bit (“*raster*”). En cualquiera de los casos se pueden realizar ciertas acciones sobre estos objetos, tales como su superposición, deformación, rotación, visibilidad, cambio de color, relleno progresivo, etc., aplicables o no dependiendo de si el objeto es vectorial o “*raster*”.

La forma habitual de estructurar una aplicación SCADA consiste en presentar un esquema general del proceso. Cada uno de los elementos del esquema general es sensible a la pulsación del dispositivo apuntador (ratón generalmente), de manera que cuando se selecciona uno de estos elementos se pasa a otra pantalla en donde se representa un esquema más o menos general según la complejidad de dicho elemento seleccionado.

2.3.2.4 Gestión de archivos históricos

Los archivos históricos son los que almacenan toda la información de funcionamiento en el pasado del proceso controlado. Sólo se almacenan aquellas variables que se configuran para ello, de manera que se suelen monitorizar muchas más variables de las que se almacenan.

Estos archivos históricos, al ser un registro del funcionamiento del sistema contienen la respuesta de muchos posibles fallos, y además pueden ser utilizados con los siguientes propósitos:

- Elaboración de informes de gestión.
- Elaboración de estadísticas de funcionamiento.
- Realización de estudios sobre los datos recogidos para predicciones y optimizaciones en el funcionamiento.
- Búsqueda de posibles anomalías en el control.

De todo esto se puede deducir la importancia de los archivos históricos. Se necesita analizar los datos almacenados y realizar estudios sobre ellos para extraer la información útil; para ello se han de facilitar las herramientas necesarias, bien usando el propio SCADA si lo permite, o mediante programas externos (hojas de cálculo, bases de datos, etc.). Usualmente serán las herramientas hechas a medida las más apropiadas para realizar este tipo de análisis puesto que tratan de manera más específica cada tipo de dato.

2.3.2.5 Las alarmas

Las alarmas son mensajes que se generan a partir de los datos que envían las estaciones remotas, para indicar que se ha producido alguna situación límite que debe ser corregida lo antes posible. Ejemplos típicos de alarmas son: depósitos que alcanzan un nivel excesivo o demasiado bajo, presiones que alcanzan valores fuera de lo admisible, elementos de seguridad que actúan, etc.

Las alarmas son la manifestación de una de las funciones básicas de los sistemas SCADA: la supervisión. El operador, cuando recibe una alarma, debe realizar las acciones oportunas para corregir dicha situación y reconocer la alarma (decirle al sistema que se da por enterado). En ocasiones, la situación de alarma se corrige sin que el operario de supervisión haga o tenga que hacer algo para corregirla, en cuyo caso, la situación de la alarma pasa a ser de alarma corregida, aunque debe ser igualmente reconocida. En todo caso, las alarmas quedan registradas en un fichero tal como van sucediendo.

Se entiende como gestión de alarmas a todas aquellas acciones realizadas sobre las alarmas: presentación en pantalla o sinóptico, envío hacia los centros de control, almacenamiento en ficheros históricos de eventos, o impresión en el dispositivo de salida correspondiente.

La estructura de una alarma es normalmente la siguiente: fecha y hora en que se produce, estación remota donde se produce, descripción, estado actual, fecha y hora en la que se corrige, y fecha y hora en la que se reconoce la alarma. Estos datos son necesarios para su almacenamiento y posterior análisis, con el objetivo de facilitar la resolución de posibles problemas. Resulta especialmente útil conocer el momento en que se producen y los momentos en los que se reconocen y solucionan.

Los mensajes de alarmas deben ser dirigidos al puesto de control permanente para que sean detectados por personas con capacidad de resolver la situación. Es práctica común la utilización de la pantalla del computador para la comunicación de las alarmas a los operarios supervisores, que son quienes deben reconocerlas. Pero es también una práctica generalizada la utilización de dispositivos de salida tales como impresoras (de línea y no de página) para que quede constancia escrita del suceso, o de un modem (telefónico o vía radio) para poder notificar las alarmas a operarios desplazados del puesto de control permanente.

La activación de las alarmas puede ser normalmente, o bien por niveles (muy alto, alto, bajo o muy bajo), o bien cuando el valor de la derivada con respecto al tiempo supera un cierto valor. Pueden existir otras condiciones de activación, pero dependerá del software SCADA que se utilice.

2.3.2.6 La seguridad

La potencia y flexibilidad que ofrece un sistema SCADA facilita el control además de la posibilidad del manejo remoto, pero también abre la posibilidad de que personal no cualificado realice acciones sobre el proceso que puedan perjudicar a la calidad del servicio o provocar graves averías. Luego es necesario que el sistema limite las posibilidades de control según el tipo de usuario.

Generalmente se definen los siguientes niveles de usuario: diseñador, gerente, supervisor, operador, y el resto de usuarios. Al nombre de un usuario se le asigna una clave de identificación para que solamente él pueda introducirse en el sistema con su nombre. Esto es importante ya que así cada usuario es responsable de las acciones y alarmas que se producen durante su turno, puesto que estas quedan registradas bajo su nombre.

2.3.2.7 Comunicaciones

Las comunicaciones se encargan de la transferencia de información entre la planta y la arquitectura hardware que soporta al SCADA, y entre ésta y el resto de elementos informáticos de gestión.

El módulo de comunicaciones contiene los *drivers* de conexión con el resto de elementos digitales conectados, entendiendo por *driver* un programa encargado de la inicialización del enlace, aplicación de los formatos, ordenación de las transferencias, etc., es decir, de la gestión del protocolo de comunicación. En el Capítulo 5 de esta memoria se describe con detalle el *driver* diseñado para comunicaciones por radio.

En aplicaciones de instrumentación existen *drivers* para el bus GPIB (IEEE-488) y para el bus VXI (extensión del VME para instrumentación), principalmente. Y en aplicaciones industriales orientadas a dispositivos de planta existen *drivers* para las principales redes industriales o buses de campo tales como Profibus, Fieldbus, CANbus, etc., así como buses propietarios de los principales fabricantes de autómatas programables del mercado: Siemens, Allen-Bradley, Modicom, y Omron, entre otros.

Aunque no todos los productos SCADA presentes en el mercado lo soportan, algunos son capaces de realizar comunicaciones entre aplicaciones residentes en distintos computadores conectados en red. De esta forma, cada uno de estos computadores puede estar dedicado a una parte del proceso, con una parte de las estaciones remotas conectadas a cada uno de ellos, pudiéndose, desde cualquier computador del sistema, monitorizar y/o controlar variables de las estaciones remotas conectadas a otros computadores. Las posibilidades de este modo de funcionamiento son muy amplias, pero también dependen mucho de las características del sistema operativo que da soporte al SCADA (Windows NT / 98, OS/2, Unix, etc.). El sistema operativo que requiere una aplicación típica es un sistema multitarea, con una gestión avanzada de errores, que proporcione control de acceso y con soporte para comunicaciones multiprotocolo (TCP/IP, NetBEUI, NetBIOS, IPX/SPX, etc.).

Los SCADA suelen incorporar las normas del mercado para las comunicaciones entre procesos y con periféricos. Por ejemplo, en Windows se ha venido utilizando el DDE (intercambio dinámico de datos), y las DLL (librerías dinámicas de datos), para implementar los *drivers* de comunicaciones para los dispositivos industriales de control y adquisición (autómatas

programables, dispositivos específicos de control distribuido, tarjetas de adquisición de datos, etc.).

En la misma línea, es importante también el soporte al estándar de conectividad entre bases de datos relacionales, ya que ello permite acceder desde la aplicación, de forma transparente y bidireccional, a datos recogidos en tiempo real y a datos de tipo relacional de forma simultánea mediante consultas SQL. Normalmente este tipo de soporte cubre a las bases de datos más extendidas: dBase, Access, Informix, Oracle, Sybase, etc.

2.3.2.8 Paquetes SCADA

Tradicionalmente, teniendo en cuenta el tipo de funcionalidad, las aplicaciones de adquisición de datos y control de procesos se clasifican en tres grandes grupos: las orientadas al control de procesos (actúan directa y continuamente sobre las variables de control), las dedicadas a monitorización y registro de datos (sin responsabilidad sobre el proceso), y las de control de producción (o gestión de la producción).

La evolución tecnológica ha llevado a unificar las plataformas hardware de estos tres tipos de aplicaciones y a avanzar en la integración de aplicaciones. Actualmente se encuentran aplicaciones SCADA que incorporan funcionalidades de los tres tipos de aplicaciones en un único entorno, con uno o varios módulos, normalmente desarrollados o suministrados por un mismo fabricante.

En todos los casos se trata de aplicaciones de integración que interrelacionan dispositivos de campo, computadores personales, bases de datos y diferentes aplicaciones. Por esta razón, el usuario desea que le ofrezcan un sistema abierto que responda a estándares ampliamente difundidos y, por tanto, que le condicione lo menos posible en la selección de los demás elementos del sistema informático, le proporcione un entorno de trabajo familiar, y reduzca la necesidad de formación y el tiempo de implantación efectiva.

Durante estos últimos años han aparecido tres líneas de evolución principales: la intercomunicabilidad entre aplicaciones, la normalización de las comunicaciones con los dispositivos de campo, y la adopción del entorno de comunicaciones Internet. Todas ellas con un denominador común: “la integración del sistema informático de planta con el sistema global de la empresa”.

Entre los elementos que caracterizan la oferta de software SCADA destacan la programación en VBA (*Visual Basic for Applications*), servidor "web" para acceso vía Internet (tecnología basada en la arquitectura DNA⁵ de *Microsoft*), utilización creciente del OPC (*OLE for Process Control*) para la comunicación entre aplicaciones y con dispositivos (MES⁶, ERP⁷, comunicaciones con buses de PLC), uso creciente de la tecnología ActiveX, y en general compatibilidad con el entorno y tecnología de *Microsoft*.

En la Figura 2.6 se pueden observar una tabla con las características más importantes de algunos SCADA comerciales representativos.

Producto	Fabricante	Plataforma hardware	Sistema Operativo	Drivers	Enlaces y protocolos
Factory Suite 2000 / InTouch	Wonderware (USA)	PC	W-3x, W-98, W-NT	CAN, Profibus, Ethernet, Interbus-S, etc.	ActiveX, DDE, OCX, ODBC, OPC.
LabVIEW	National Instruments (USA)	PC, HP, PowerMac, SUN	W-3x, W-98, W-NT, UNIX.	Allen Bradley, Fieldbus, Omron, Siemens, Profibus, VXI, etc.	TCP/IP, NetDDE, UDP, etc.
RSView 32	Rockwell Automation	PC	W-98, W-NT	DH 485, DeviceNet, Ethernet, etc.	DDE, ODBC, OLE, OPC, ActiveX, DCOM, COM.
FIX Dynamics	Intellution (USA)	PC	W-NT	Profibus, Ethernet, de diversos PLCs y CNs, RS232/422/485	ActiveX, ODBC, OLE, etc.
SattGraph 5000	Alfa Laval Automation	PC	W-NT	Sattbus, Comli, Siemens, DataLink, DataHigh, SNP-X, J-Bus, etc.	Los soportados por W-98 y W-NT.
SCAN 3000	Honeywell	PC, Digital	W-NT, UNIX	Modbus, Profibus, Honeywell, etc.	TCP/IP.
Supervisor	Adasa Sistemas Compax	Sun, DEC Alpha, VAX	UNIX, Open VMS	---	Decnet, TCP/IP.
WinCC	Siemens	PC	W-98, W-NT	Profibus, OPC, S5, Mitsubishi, etc.	Los soportados por W-98 y W-NT.
WizFactory	PC Soft International	PC	W-98, W-NT, OS2	Todos los estándares del mercado	DDE, ODBC, SQL, Interfaces.
Cube	ORSI (I)	PC	W-NT	FMS, Profibus, Ethernet, Sinec H1, 3964-R, Modbus, AB	DDE, OPC, ODBC, ficheros.
Genesis32	Iconics (USA)	PC	W-98, W-NT	Los principales y cualquiera con servidor OPC	DDE, OPC, ODBC

Figura 2.6: Paquetes SCADA comerciales representativos (revista Automática e Instrumentación, año 1998/99)

⁵ DNA = *Distributed interNet Application*

⁶ MES = *Manufacturing Execution System*

⁷ ERP = *Enterprise Resource Planning*

La estructura global de un SCADA queda bien recogida en la Figura 2.7 correspondiente al software CUBE de la firma Orsi. El conjunto de servidores constituye el núcleo del SCADA y este núcleo está rodeado de las interfaces con dispositivos, bases de datos, sistemas de comunicaciones y otras aplicaciones propias y de terceros.

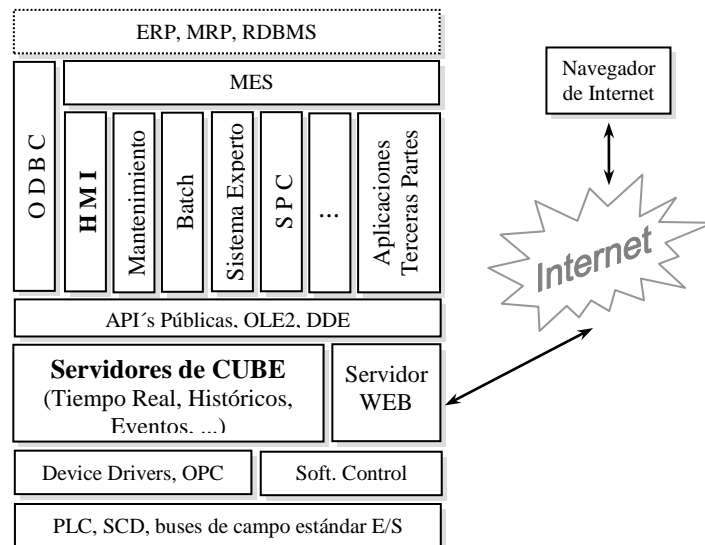


Figura 2.7: Estructura global de un SCADA (Cube)

2.4 Técnicas modernas aplicadas a la optimización de procesos

2.4.1 Introducción

La automatización de procesos es una tarea que viene realizándose desde hace ya muchos años, y mediante la cual se ha llegado a unos niveles de explotación altamente mecanizados. Sin embargo, y ante una evolución futura de los sistemas hacia mayores cotas de productividad y competitividad, se está trabajando en la actualidad en el diseño de sistemas automáticos de operación en tiempo real que busquen la optimización de los costes de explotación y de la calidad de servicio.

El objetivo de la optimización es encontrar una solución que satisfaga un conjunto de restricciones, de forma que una función objetivo sea maximizada o minimizada. La consecución del óptimo en un problema conlleva dos aspectos fundamentales: la correcta formulación del mismo y el uso de un algoritmo de cálculo apropiado. El primer aspecto se tratará para cada caso en el Capítulo 8, y a continuación se van a tratar los procedimientos de cálculo más habituales.

Los métodos clásicos de optimización tratan de llegar al óptimo matemático, a base de algoritmos más o menos sofisticados, cuyo objetivo principal consiste en alcanzar dicho óptimo con el mínimo número de cálculos, y con la máxima garantía en cuanto a que el punto

alcanzado corresponda realmente al óptimo global del problema formulado. Ambos aspectos caracterizan lo que actualmente se conoce como *robustez* de un algoritmo de búsqueda.

Las dificultades encontradas por los métodos clásicos para resolver el problema de diseño óptimo son grandes, por lo que no debe extrañar que habitualmente se recurra a simplificaciones. Una de las primeras simplificaciones aplicadas ha sido la de linealizar el problema. En tal caso, la robustez del algoritmo de cálculo es máxima, por cuanto el mínimo se encuentra en un número finito de pasos si el problema está correctamente formulado, por ejemplo mediante el algoritmo SIMPLEX [Pre-92]. El inconveniente principal es que la linealización puede desvirtuar el problema original. No obstante, la técnica de linealización puede combinarse con otras más generales para llegar al óptimo a través de un proceso iterativo.

En general, existen otros muchos métodos que tratan de llegar al óptimo de distintas formas y con el fin de cumplir con los objetivos de rapidez y fiabilidad.

Sin embargo, en determinadas situaciones, para resolver un problema de optimización en el funcionamiento de un proceso pueden admitirse también métodos de cálculo no excesivamente precisos, pero sí lo suficientemente robustos; es decir, que permitan garantizar la consecución de una solución más o menos próxima a la óptima.

A continuación se van a revisar, bajo esta óptica, otras técnicas de optimización de tipo heurístico, menos exactas, pero que están teniendo cada vez más aceptación. Estas metodologías de computación (denominadas *soft computing* [Zad-96]) aplicadas a la resolución de problemas de optimización hacen uso de nuevas teorías más abiertas a la casuística del mundo real. La probabilidad de un suceso desempeña cada vez un papel más importante en la toma de decisiones.

En la actualidad dentro de este concepto se incluyen clases de metodologías tales como la neurocomputación, la computación evolutiva, y la lógica difusa, cada una con unas capacidades propias que se describen de forma muy resumida en las siguientes secciones.

2.4.2 Redes Neuronales

Históricamente se puede considerar que la *neurocomputación* o *computación neuronal* nació en 1943 con un trabajo de McCulloch y Pitts [McC-43], con el que usando la lógica formal desarrollaron sencillos modelos de redes neuronales basándose en conocimientos neurobiológicos.

Las redes neuronales artificiales (RNAs) [Koh-88, Hil-95, Pri-96] siguen un proceso de cálculo que trata de emular al cerebro humano. Las salidas se relacionan con las entradas mediante un proceso de intercambio de información entre los núcleos básicos de información, que se corresponderían con las neuronas del cerebro. Las diferentes neuronas se disponen por capas y se comunican entre sí a través de las dendritas o líneas de enlace, las cuales están afectadas por un factor de ponderación que determina la capacidad de transmisión de dicha dendrita o proceso de sinopsis. Asignando valores adecuados a dichos factores es posible conducir el razonamiento en un sentido u otro. En este caso el proceso de aprendizaje se lleva a cabo revisando los factores de ponderación de las dendritas en función de los hechos observados y de las repercusiones sobre el sistema de decisiones anteriores.

Las RNAs suponen un paso adelante entre los sistemas de ayuda a la toma de decisiones, ya que son capaces de incorporar la experiencia pasada ante la toma de una futura decisión, pero no por los métodos de inferencia clásicos basados en series temporales o análisis espectrales, sino ponderando cada hecho observado sobre los factores internos que condicionan el proceso intuitivo.

Desde un punto de vista formal, la contribución de una sinapsis en una neurona se puede representar por un valor o peso (w), con lo que para una neurona con n entradas el potencial de membrana X_T sería (Figura 2.8):

$$X_T = \sum_i w_i \cdot x_i + I \quad ; i = 1, \dots, n$$

donde w_i es el peso (*sinapsis*) asociado a la entrada x_i procedente de la salida de otra neurona o de una de las entradas a la red, e I es una entrada externa de control. Pueden emularse conexiones excitatorias o inhibitorias sin más que suponer pesos positivos o negativos, respectivamente.

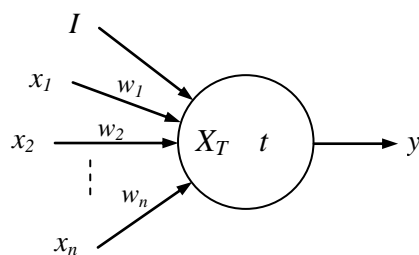


Figura 2.8: Esquema del modelo simplificado del comportamiento de una neurona

Si el potencial de membrana, X_T , es mayor que un determinado *umbral*, t , la neurona se disparará, y si es igual o está por debajo de él no se disparará. Matemáticamente se puede representar este hecho diciendo que la *salida*, $y=f(X_T - t)=f(u)$, donde $u=X_T - t$ es el potencial de activación, y f es una función no lineal denominada *función de activación* (que dependiendo del modelo o arquitectura de red, será una función umbral, una función rampa, una función sigmoideal, o una función de base radial).

En cuanto a la estructura de las RNAs, en la Figura 2.9 se representan los factores que se pueden modificar, y que dan lugar a los muchos tipos diferentes de redes neuronales existentes, cada uno de los cuales tienen unas aplicaciones particulares más apropiadas.

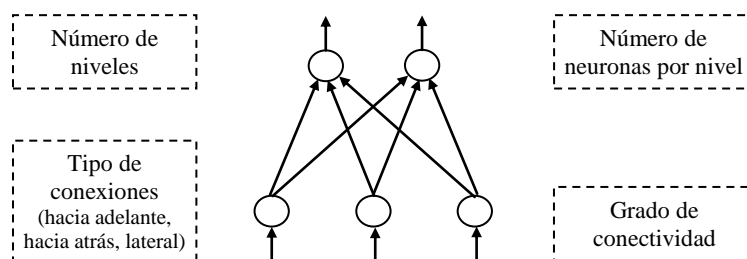


Figura 2.9: Factores modificables en una red neuronal en cuanto a su estructura

Debido a su constitución y a sus fundamentos, las RNAs presentan un gran número de características semejantes a las del cerebro. Por ejemplo, son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan información irrelevante, etc. Esto hace que ofrezcan numerosas ventajas y que este tipo de tecnología se esté aplicando en múltiples áreas. Estas ventajas incluyen [Mar-90]:

- *Aprendizaje adaptativo:* capacidad de aprender a realizar tareas basadas en un entrenamiento o una experiencia inicial.

- *Autoorganización*: una red neuronal puede crear su propia organización o representación de la información que recibe mediante una etapa de aprendizaje.
- *Tolerancia a fallos*: la destrucción parcial de una red conduce a una degradación de su estructura; sin embargo, algunas capacidades de la red se pueden retener, incluso sufriendo un gran daño.
- *Operación en tiempo real*: los computadores neuronales pueden realizar grandes procesos con datos de forma muy rápida, debido a su implementación paralela; y se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.
- *Fácil inserción dentro de la tecnología existente*: se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas, lo cual facilita la integración modular en los sistemas existentes.

Por lo tanto la forma de utilización de las RNAs es diferente a la de los computadores convencionales: no es necesario conocer el algoritmo correspondiente a la tarea a realizar. El aprendizaje permite a la red neuronal descubrir patrones y sus relaciones, y organizarse a sí misma para efectuar asociaciones. Esta capacidad tiene dos implicaciones de suma importancia: la habilidad de resolver problemas cuyas reglas son difíciles de formular, y la habilidad de extraer de grandes conjuntos de datos con numerosas variables tanto modelos estadísticos como reglas basadas en el conocimiento.

Hay gran diversidad de aplicaciones prácticas en las que las RNAs complementan o superan a otras tecnologías, en los campos que se indican a continuación: procesamiento de patrones, agrupación/categorización, optimización y planificación, filtrado de señales, segmentación de datos, aproximación de funciones, predicción/pronóstico, extracciones en bases de datos y búsquedas asociativas, control adaptativo, interfaces adaptativas hombre/máquina, etc. Las RNA han dado lugar a gran cantidad de productos comerciales y aplicaciones prácticas, como por ejemplo: la retina desarrollada por Synaptics para reconocimiento de direcciones postales de sobres, los sistemas de control de plantas industriales ideados por Fujitsu, Jawasaki y Nippon Steel, el sistema para determinar la calidad del material de asientos de vehículos (Citroën), los chips neuronales utilizados para ahorro de combustible y predicción de fallos en automóviles (Mercedes Benz y Ford), y el sistema neuronal de ayuda al piloto en el control del avión en caso de alcance por el fuego enemigo (incluido en el avión de combate F-15), entre otras.

En esta memoria, concretamente en el Capítulo 8, se describe el uso de una de las RNAs más utilizadas (*Backpropagation*) para predecir el consumo diario de agua de la ciudad de Granada.

2.4.3 Neuro-programación dinámica

La programación dinámica [Ber-95b] es una técnica que permite resolver problemas de optimización en aplicaciones de planificación de tareas, gestión de inventarios, asignación de canales de comunicación, y en muchas otras en los ámbitos de la investigación operativa, el control de procesos, las comunicaciones, la biología, etc. Se trata de problemas en los que hay que tomar decisiones en etapas (por ejemplo, hay que determinar qué señales de control se aplican a un sistema cada cierto intervalo de tiempo) y en los que el efecto de cada decisión no se puede conocer con total precisión antes de tener que tomar la siguiente (ver Figura 2.10). Cada decisión tiene un coste inmediato asociado, pero también afecta al coste de las etapas futuras al determinar los estados siguientes, en los que se han de tomar las siguientes decisiones. Por lo tanto, la obtención de una solución de bajo coste en la etapa presente debe contrapesarse con la posibilidad de obtener costes elevados en un futuro.

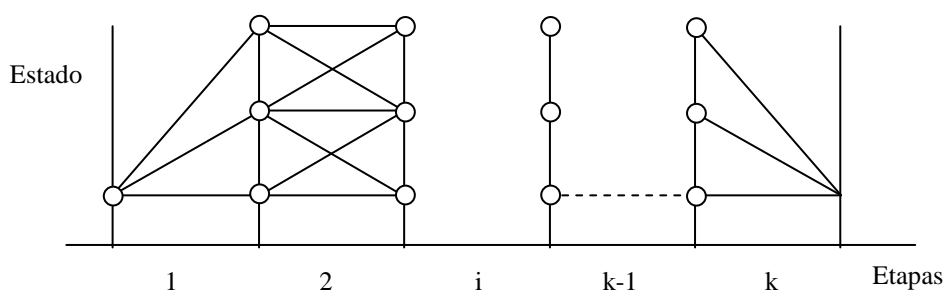


Figura 2.10: Proceso de búsqueda del óptimo por programación dinámica

La programación dinámica proporciona la metodología necesaria para establecer el compromiso entre *coste inmediato* y *coste futuro*. Para ello se considera un sistema dinámico discreto en el tiempo, cuyos estados cambian según una probabilidad de transición que depende de la señal de control aplicada (la decisión tomada), $Y(i^*t)$. Si el sistema está en el estado $X(i^*t)=a$, y se aplica $Y(i^*t)=u$, el paso al siguiente estado $X((i+1)^*t)=b$ se produce con una probabilidad $p_{ab}(u)$. La transición desde a hasta b cuando se aplica u tiene un coste inmediato asociado, $g(a,u,b)$, aunque también es necesario tener en cuenta el efecto del estado b al que se pasa. Para ordenar los estados siguientes se utiliza el coste mínimo, $J^*(b)$, de cada estado b

considerando todas las etapas restantes. La función $J^*(b)$ se denomina *coste futuro óptimo* (*optimal cost-to-go*) del estado b y satisface el principio de Bellman [Bel-62]:

$$J^*(a) = \min_u E[g(a,u,b) + J^*(b) | a,u], \text{ para todo } a$$

donde $E[g(a,u,b) + J^*(b) | a,u]$ es el valor esperado con respecto a b , dadas a y u . Es decir, para cada estado a , la señal de control óptima, u , es la que minimiza la expresión previa. El objetivo de la programación dinámica es determinar, para cada estado a , la función de coste óptimo J^* y el valor de u para ese coste óptimo.

En las aplicaciones realistas, el coste computacional asociado a la programación dinámica suele ser demasiado elevado dado el gran número de estados y señales de control posibles. Los problemas presentan espacios de estados de muchas dimensiones y de variables continuas cuya discretización da lugar a problemas de gran tamaño. Además, para determinar el valor esperado asociado a cada estado, se necesita conocer las probabilidades de transición entre estados (a,b) para cada u . Este carácter estocástico del problema surge de la incertidumbre que existe en las transiciones de estado que finalmente se producen, dado que los valores de las entradas externas y las perturbaciones no se conocen con anterioridad y son incontrolables.

En los últimos años se está construyendo una nueva metodología [Ber-96] para afrontar los problemas que limitan la aplicabilidad de la programación dinámica a situaciones realistas:

- (1) El problema de la complejidad (*curse of dimensionality*), que crece exponencialmente con el tamaño del mismo, y
- (2) El problema de la construcción del modelo (*curse of modeling*) explícito del sistema, la determinación de las probabilidades de transición, p_{ab} , y de los costes asociados, $g(a,u,b)$.

Esa metodología recibe el nombre de *neuro-programación dinámica*, e incorpora a la programación dinámica el uso de las redes neuronales, la teoría de la aproximación funcional, y la optimización iterativa.

La neuro-programación dinámica pretende desarrollar procedimientos de optimización basados en la determinación aproximada de la *función de coste futuro*, J^* . Usualmente esto se hace mediante simulación de un modelo aproximado del sistema. Así, las probabilidades de transición no tendrían que determinarse explícitamente, sino que la función de coste futuro para una determinada secuencia de decisiones se calcula progresivamente mediante la simulación

Monte Carlo de varias trayectorias posibles de funcionamiento del sistema y la evaluación de sus respectivos costes. Con esta alternativa, la función $J^*(b)$ se sustituye por una lista constituida por el valor de J^* para cada estado b , o bien por una aproximación $J^*(b,r)$, donde r es un vector de parámetros (pesos en el caso de un aproximador neuronal), y la secuencia de decisiones óptima (sub-óptima, puesto que se utiliza un procedimiento aproximado), μ , se determina a partir de:

$$\mu(a) = \arg \min_u E[g(a,u,b) + J^*(b,r) \mid a,u]$$

La evaluación de $E[g(a,u,b) + J^*(b,r) \mid a,u]$ para cada u es bastante complicada usualmente y, aún si el cálculo de la función $J^*(b,r)$ es simple, precisa un tiempo de procesamiento elevado, inaceptable si existen ciertas restricciones de tiempo real. Una alternativa a esta situación parte de la definición de los factores-Q [Wat-92]:

$$Q^*(a,u) = E[g(a,u,b) + J^*(b,r) \mid a,u]$$

que se aproximan mediante la función $Q^*(a,u,r)$, de forma que el siguiente estado se obtiene a partir de:

$$\mu(a) = \arg \min_u Q^*(a,u,r)$$

Para construir las aproximaciones de los factores-Q existen básicamente dos procedimientos, la iteración de políticas aproximadas (*approximate policy iteration*) [Ber-96], y el aprendizaje-Q [Wat-92]. La iteración de políticas aproximadas necesita un modelo del sistema (las probabilidades de transición $p_{ab}(u)$), mientras que el aprendizaje-Q no.

Ejemplos de aplicaciones reales donde se ha utilizado la neuro-programación dinámica son: asignación de canales en sistemas celulares [Ber-96], control de inventarios [Ber-95b], estrategias óptimas para juegos como el *tetris* o el *backgammon* [Tes-95], planificación de varios ascensores en un edificio [Cri-96], y problemas de planificación en las misiones del transbordador de la NASA [Zha-95]. En el Capítulo 8, que trata sobre optimizaciones aplicadas a la red de distribución de agua de la ciudad de Granada, se propone un nuevo procedimiento dentro de este ámbito de la neuro-programación dinámica, y se aplica al problema de distribuir un caudal prefijado entre una serie de depósitos de forma que se satisfaga la demanda existente sin que se derramen ni se vacíen los depósitos.

2.4.4 Algoritmos Genéticos

La Computación Evolutiva⁸ (CE) [Fog-95, Bac-97, Fog-00] surge a finales de los años 60 cuando John Holland se planteó la posibilidad de incorporar los mecanismos naturales de selección y supervivencia para resolver una gran cantidad de problemas que ya habían sido "resueltos" muy eficientemente por la Naturaleza. Pero no es hasta mediados de los 80, con la aparición de los computadores de altas prestaciones y bajo coste, cuando el desarrollo de la CE crece de forma espectacular.

Los algoritmos genéticos (AGs) [Gol-89, Gor-93, Mic-94, Hei-96] son, con diferencia, el paradigma más usado de entre los de la CE para la búsqueda de soluciones en espacios complejos. Los AGs hacen uso de las reglas de selección natural que guían el proceso evolutivo de los seres vivos, las cuales conducen como sabemos a una mejora de la especie. Estas técnicas tratan por tanto de emular dicho proceso para conseguir una solución cada vez mejor que la actual en su búsqueda del óptimo.

El método está especialmente indicado para todos aquellos problemas en los que otros métodos de búsqueda no resultan convenientes, bien sea porque el espacio de soluciones no es conexo, por la existencia de varios óptimos locales, o por la imposibilidad de obtener derivadas de la función objetivo.

La mayoría de las técnicas de búsqueda convencionales parten de una solución factible que se pretende mejorar aplicando criterios de muy diverso tipo, pero cuya característica general es el desplazamiento en la dirección que indica una posible mejora de la función objetivo. Cuando la complejidad del problema aumenta, la estructura topológica del espacio de soluciones presenta peculiaridades cada vez más insalvables, que dificultan de modo considerable la exploración de dicho espacio.

Por el contrario, los AGs están basados en la exploración del espacio de soluciones a partir de un conjunto de puntos aleatoriamente dispuestos (población), de forma que mediante unas sencillas reglas de transición probabilística se generan sucesivas poblaciones que mejoran el valor (aptitud) obtenido en las anteriores.

⁸ Un concepto más amplio que engloba a una serie de técnicas de resolución de problemas complejos basadas en la emulación de los procesos naturales de evolución.

Puesto que las reglas de transición que permiten construir una población de soluciones a partir de otra son de naturaleza probabilística, no aseguran que se alcance la solución óptima, pero sí garantizan una exploración más exhaustiva del espacio de soluciones.

Según Goldberg (1987), los AGs responden a unas expectativas para las que otros métodos de búsqueda tradicionales resultan limitados:

- Se trata de un procedimiento general e independiente de la estructura del problema.
- Requiere un mínimo de información auxiliar para guiar la búsqueda.
- Presentan una perspectiva más global que el resto de técnicas, puesto que exploran el espacio de soluciones a partir de un conjunto de puntos.

En los AGs, para conseguir una “evolución” de la población inicial hacia nuevas poblaciones cuya aptitud promedio sea cada vez superior, se hace uso de tres procesos básicos: selección, cruzamiento y mutación. Dependiendo de cómo se representen los individuos de la población, cadenas de bits o números reales, estos tres procesos de evolución se programarán de una manera o de otra.

Aunque la intervención del azar es fundamental en la realización de estos procesos, la búsqueda no es aleatoria, al menos en su totalidad, puesto que está dirigida por procesos y parámetros bien definidos. Los AGs pueden calificarse como métodos heurísticos, puesto que es el propio algoritmo quien “aprende” a orientar el proceso de búsqueda hacia poblaciones con una mayor aptitud promedio.

Aunque no puede demostrarse formalmente que los AGs poseen una efectividad mayor que otros métodos de búsqueda, ni tan siquiera que sean más efectivos que una simple búsqueda aleatoria, su aplicación práctica pone de manifiesto que la aptitud promedio de la población aumenta como consecuencia de su aplicación sucesiva.

La Figura 2.11 muestra el esquema del proceso de optimización empleando un algoritmo de tipo genético.

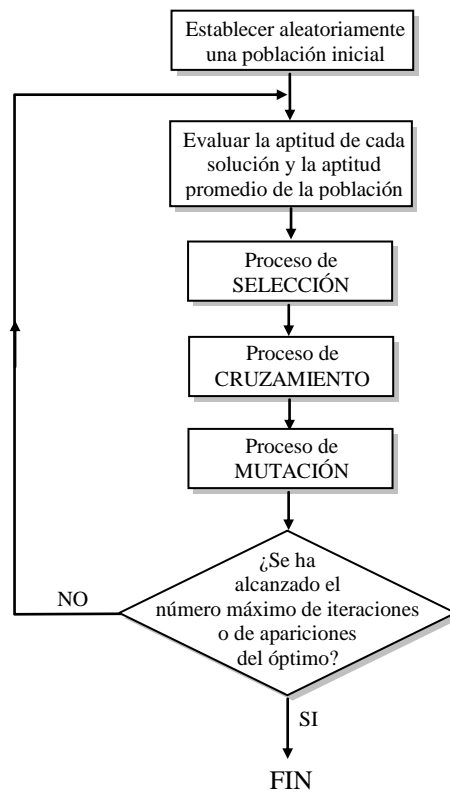


Figura 2.11: Esquema de funcionamiento de un algoritmo genético

Frente a la aplicación de otros posibles métodos, podemos destacar las siguientes ventajas:

- El algoritmo es completamente general e independiente del problema concreto que se pretende solucionar. Los únicos vínculos que hay que establecer entre ambos son por un lado, la representación de las variables de decisión y por otro, la función objetivo adecuadamente formulada para evaluar la aptitud de las posibles soluciones.
- Aunque un AG no permite identificar si la solución óptima obtenida es el óptimo global, sí proporciona un conjunto de soluciones cuya aptitud promedio mejora durante el proceso.
- La intervención del azar en las etapas de reproducción, cruzamiento y mutación, aseguran una exploración exhaustiva del espacio de soluciones mediante una población de pequeño tamaño, con el ahorro consiguiente de tiempo de cálculo y memoria del computador.

Y los inconvenientes principales que presentan son:

- No permite identificar si realmente se ha alcanzado la solución óptima global.
- Derivado del anterior inconveniente, no es posible establecer un criterio de convergencia directamente relacionado con la aptitud de la solución.
- Coste computacional elevado.

En la actualidad funcionan con éxito aplicaciones industriales de los algoritmos genéticos en campos tan dispares como el diseño de turbinas multietapa, el control de redes de distribución de energía eléctrica o de gas, la planificación de tareas, el cálculo de estrategias de mercado o el cortado de piezas con mínimo desperdicio, por citar algunas.

En el Capítulo 8 se dan más detalles de implementación, ya que se utiliza un algoritmo genético en el contexto de la neuro-programación dinámica, descrita en la sección anterior, para resolver el problema de optimizar el reparto de agua entre los depósitos principales existentes en la red de abastecimiento de agua de la ciudad de Granada.

2.4.5 Lógica difusa

En 1965, L. Zadeh [Zad-65] aplicó la lógica multivaluada, desarrollada anteriormente por J. Lukasiewicz, a la teoría de conjuntos, estableciendo la posibilidad de que los elementos pudieran tener diferentes grados de pertenencia a un conjunto. Zadeh introdujo el término difuso (*fuzzy*) y desarrolló un álgebra completa para los conjuntos difusos o borrosos (*fuzzy*), aunque estos conjuntos no tuvieron aplicación práctica hasta mediados de los años setenta, cuando E. Mamdani diseñó un controlador difuso para un motor de vapor [Mam-74].

En lógica difusa se trabaja con conjuntos que se definen por sus funciones de pertenencia, que se denotan como $\mu_z(x)$ e indican el grado de pertenencia (entre 0 y 1) del elemento con valor x al conjunto Z .

La lógica difusa fue empleada originalmente para construir modelos de inferencia para tratar el conocimiento impreciso, básicamente en el contexto de la construcción de Sistemas Expertos. Sin embargo, actualmente, su mayor, más conocida y notable aplicación es en el control de procesos, un campo muy alejado de aquel que la impulsó en sus primeras etapas. Los fundamentos sobre la teoría de control difuso y cuestiones relacionadas con el diseño y construcción de controladores se pueden encontrar en [Yag-92, Tri-94].

La idea de base es que las relaciones de entrada-salida (reglas de control) de un sistema complejo suelen ser descritas por los expertos en forma de sentencias condicionales SI-ENTONCES, con antecedentes(s) y consecuente(s) dados en forma de proposiciones difusas. Y a partir de esta descripción y de la observación del comportamiento del sistema, se puede deducir la acción de control empleando las leyes de inferencia de la lógica difusa.

En la Figura 2.12 se muestra el funcionamiento de un sistema de control difuso, en el que se puede distinguir los siguientes componentes principales:

- Una primera etapa de conversión de entradas reales en un conjunto difuso. Es lo que se denomina proceso de "fuzzificación".
- Una base de conocimiento, consistente en la definición de los valores lingüísticos de cada una de las variables de entrada/salida del sistema y las reglas que lo conforman.
- Otra etapa de evaluación de reglas de control difuso con capacidad de simular el proceso de toma de decisiones de un operador humano. Dependiendo del diseño de los consecuentes de las reglas difusas, se pueden distinguir en la bibliografía tres modos fundamentales de realizar dicho proceso: modelo de Mamdani, modelo de Takagi-Sugeno-Kang, y modelo de Tsukamoto.
- Y una última de conversión del conjunto difuso obtenido en un valor numérico de salida que se pueda aplicar en las aplicaciones industriales. Es lo que se denomina proceso de "defuzzificación". En la bibliografía existen distintos métodos para realizar este proceso: centro de área, centro de sumas, centro de picos, centro de área mayor, primer máximo, media de máximos y método de calidad ξ , entre otros.

Como ejemplo, se indica la expresión del de centro de área (CDA), uno de los métodos más utilizados tanto en aplicaciones reales de control como en realizaciones electrónicas:

$$Z_{CDA} = \frac{\int_Z z \mu_Z(z) dz}{\int_Z \mu_Z(z) dz}$$

Caso de una distribución continua, donde Z' es el conjunto difuso final, resultado de combinar los diferentes valores lingüísticos de la variable de salida que son activados.

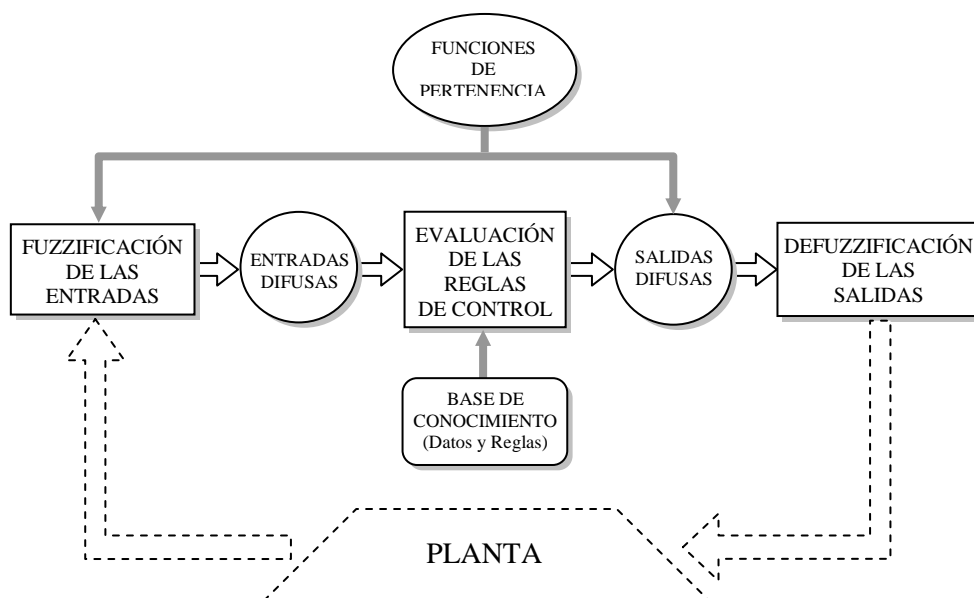


Figura 2.12: Funcionamiento de un sistema de control difuso

El actual interés en la teoría difusa es, en gran parte, debido a la gran cantidad de aplicaciones y procesos en que los controladores difusos han sabido operar con efectividad. Algunas de las razones para que esto haya sido así son [Wan-94]:

- Uso efectivo de toda la información disponible: en cualquier trabajo de ingeniería se debe intentar utilizar toda la información que esté disponible. A falta de un modelo matemático del proceso a controlar (modelo que generalmente no se tiene), la mayor parte de la información proviene de: (1) los sensores, que nos dan medidas numéricas de las variables externas del proceso y (2) descripciones generalmente lingüísticas de expertos familiarizados con el proceso. Contrariamente a otros tipos de controladores, los controladores difusos sí son capaces de procesar e incorporar este tipo de información lingüística.
- Control de procesos no lineales: los controladores difusos son aproximadores universales. Por tanto, siempre que se escoja un número de parámetros suficiente y se les dé un valor adecuado, podremos encontrar un controlador difuso apropiado para el control de cualquier sistema no lineal.
- El control difuso es fácil de comprender: debido a que un controlador difuso emula la estrategia de control de un ser humano, su funcionamiento es fácil de entender por personal no especialista en control. Esto permite construir algoritmos de una forma más sencilla y puede reducir el tiempo de desarrollo.
- Los controladores difusos son fáciles de implementar: gracias a la existencia de chips comerciales que pueden ser programados y reprogramados externamente y a que los sistemas difusos admiten un alto nivel de paralelismo, la implementación de un controlador difuso de altas prestaciones se puede realizar de una forma sencilla y rápida.
- Posible combinación con tecnologías clásicas ya establecidas y con otras más modernas, como las redes neuronales y los algoritmos genéticos.

En los últimos años, la lógica difusa ha sido un área de trabajo muy fructífera para el desarrollo de diferentes disciplinas (como pueden ser los problemas de interpolación, modelado e identificación de sistemas, reconocimiento y clasificación de patrones, etc.), aunque el área donde esta teoría ha proporcionado mayores resultados ha sido, como ya se ha comentado, en el campo del control de sistemas en general (especialmente en productos de electrónica de consumo) y en robótica [Yas-85, Sak-93, Kwo-95, Men-95, Yen-95]. En nuestro caso, y como se indica también en el Capítulo 8, se está utilizando actualmente para la obtención tanto de un modelo como de un controlador para la red de distribución de agua de la ciudad de Granada, con capacidad de aprendizaje y adaptación continua.

Capítulo 3:

El ciclo integral del agua en la ciudad de Granada

"La gestión del agua en Abastecimientos Urbanos no sólo implica la preocupación por un recurso escaso sino también por su calidad y distribución óptima ya que es el primer alimento para la humanidad".

Este capítulo, incluido dentro del bloque de fundamentos y descripción del problema, tiene como objetivo detallar la infraestructura y elementos que son necesarios en el proceso de explotación del agua en grandes ciudades.

En la primera sección de introducción se hace una descripción general del ciclo integral del agua, para pasar a la siguiente sección donde se dan detalles adicionales sobre la problemática que se pretende resolver de una parte concreta de este ciclo del agua: las redes de distribución.

En la sección 3.3 se da una visión particular del problema para la ciudad de Granada, con el propósito de que el lector tenga una visión más amplia y detallada del problema que se va a tratar a lo largo de esta memoria. Y al final del capítulo, sección 3.4, se muestran unas tablas de equipos instalados para cada sistema y un esquema general del abastecimiento de agua en la ciudad de Granada que dan idea de la magnitud y complejidad de este proceso.

3.1 Introducción

Pensemos en nuestro planeta sin habitantes y cómo es su ciclo del agua: “*Se evapora del océano, debido a la energía solar, se forman las nubes que, en parte, emigran hacia la tierra; allí precipitan en forma de agua (o nieve), se forman arroyos, ríos, lago, corrientes de agua subterránea hasta que, por fin, el agua vuelve al océano y comienza otra vez el ciclo*”.

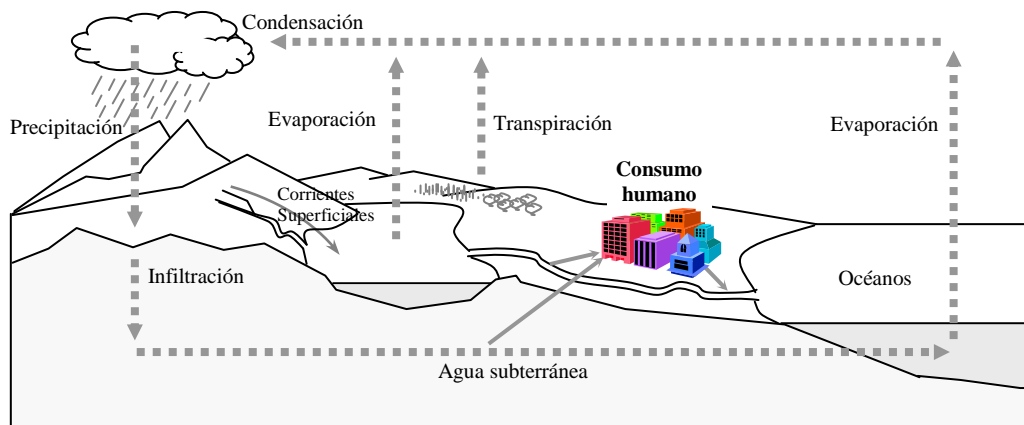


Figura 3.1: El ciclo integral del agua

Pero en este escenario (ver Figura 3.1) aparece el hombre, que influye sobre el ciclo del agua de dos formas distintas, bien directamente mediante extracción de las mismas y posterior vertido de las aguas residuales (tratadas o no) o bien alterando indirectamente la vegetación y la cobertura del suelo.

Según estimaciones del Plan Hidrológico Nacional, el consumo medio de agua para toda España se distribuye en más de un 80 por 100 para regadíos y el resto para abastecimientos urbanos e industriales. Esta proporción general se invierte en el caso de zonas turísticas y en las grandes urbes. Las aguas de regadío se devuelven en poca proporción al sistema natural, y las de abastecimientos urbanos cada día son menos reutilizables ya que la reducción de los caudales de los ríos, por efecto de los regadíos, embalses, etc., unido al vertido excesivo de aguas residuales, ha disminuido dramáticamente su capacidad de autodepuración. De ahí se deriva la necesidad creciente de construir y controlar plantas depuradoras de aguas residuales.

La situación de las aguas subterráneas no es diferente. Aunque la mayoría del agua dulce del planeta es subterránea, no toda ella es fácilmente utilizable, ni se encuentra en las zonas donde se precisa. La escasez del recurso superficial en muchas regiones ha llevado a la

sobreexplotación de los acuíferos, produciéndose casos de disminución de niveles freáticos que pueden llegar a ser alarmantes.

Por estos motivos, muchos países realizan grandes inversiones de dinero público para regular el agua (presas, plantas desalinizadoras, plantas de tratamiento y residuales, redes de distribución, sondeos de aguas subterráneas). Nuestro trabajo va encaminado a mejorar dicha regulación desde el punto de vista de la aplicación de los avances tecnológicos en instrumentación, control y comunicaciones, y de las últimas técnicas en optimización.

3.2 Redes de distribución de agua

Un factor que acentúa la carencia de agua es la concentración de las poblaciones humanas en determinadas zonas, de modo que actualmente unos 2.000 millones de personas viven en zonas de escasez crónica de agua. A medida que la población humana vaya creciendo la crisis se agravará, ya que su aumento es función exponencial del número de individuos y, por tanto, más grave en las zonas más densamente pobladas, con lo cual también irá creciendo la necesidad de ampliar y/o actualizar los sistemas de gestión actualmente instalados para hacer frente a este crecimiento.

Antes de continuar, es necesario aclarar dos definiciones relativas a las infraestructuras hidráulicas:

- **Sistema de abastecimiento:** conjunto conexo compuesto de instalaciones de producción (captaciones, pozos, plantas, conducciones de aducción), aportaciones de otros sistemas y red de distribución.
- **Red de distribución:** conjunto coherente de depósitos, instalaciones de elevación, tuberías de transporte y sectores (tuberías de distribución, acometidas, válvulas, elementos de regulación de presión, elementos de medición).

Aunque el ciclo integral del agua engloba al abastecimiento y al saneamiento, nos vamos a centrar en describir con más detalle y optimizar los aspectos relacionados con la red de distribución. Las captaciones, pozos y plantas de tratamiento también están automatizadas con la misma metodología de diseño presentada aquí, pero desde el punto de vista de la central sólo se tratan como estaciones remotas que aportan la información más relevante en cada caso. En relación al saneamiento, actualmente se está incorporando al sistema empleando también la misma metodología.

La supervisión y control de una red hidráulica implica el uso de varios instrumentos de medida, tales como transductores de presión, caudalímetros, medidores de nivel, medidores de apertura de válvulas, finales de carreras, etc. Se necesitan además dispositivos para operaciones remotas dirigidas desde el centro de control (apertura y cierre de válvulas, arranque y parada de bombas). También se necesitan instrumentos para mostrar los valores de las variables asociadas al estado de operación del equipo. En concreto, el conjunto mínimo de medidas y actuaciones a considerar en una red de distribución típica pueden agruparse de la siguiente manera:

Medidas: Nivel de depósitos y canales, presión en tuberías, caudal, cloro residual, turbidez, posición de apertura de válvulas y compuertas, intensidad de motores, temperatura, precipitación, etc.

Estados: Bomba (on/off), válvula y compuerta (cerrada/abierta), estación (remoto/local), fallo de tensión, seguridad, disponibilidad (válvulas, compuertas, motores), totalizadores, etc.

Actuaciones: Bomba (marcha/paro), y válvula o compuerta (cerrar/abrir) principalmente.

Hay gran variedad de instrumentos para medir las variables mencionadas y, en cada caso, el análisis de la instrumentación adecuada se asocia no sólo al coste, sino también a la calidad del instrumento y, particularmente, a la asistencia técnica necesaria para el mantenimiento y conservación del equipo.

3.3 El ciclo integral del agua en la ciudad de Granada

Dentro del área técnica de la Empresa Municipal de Abastecimiento y Saneamiento de Granada (en adelante EMASAGRA), los sistemas productivos se dividen en:

- Captaciones.
- Estación de tratamiento de agua potable.
- Explotación de la red de abastecimiento y distribución: red de alta y red de baja.
- Red de saneamiento.
- Estaciones depuradoras de agua residual.

Con esta infraestructura, la empresa Emasagra puede asegurar el suministro de los, aproximadamente, 300.000 habitantes de la ciudad, hasta en las épocas de más pertinaz sequía, así como la devolución del agua depurada y en condiciones óptimas de nuevo a la naturaleza.

3.3.1 Estructura general

Desde Emasagra se realiza la gestión integral del agua (ver Figura 3.2), aunque el recurso es aportado por la Confederación Hidrográfica del Guadalquivir (C.H.G.) mediante los canales de Quéntar, Pinos, o Loaysa. Las captaciones de los pozos, si bien son también de la C.H.G., en este caso son mantenidas desde Emasagra.

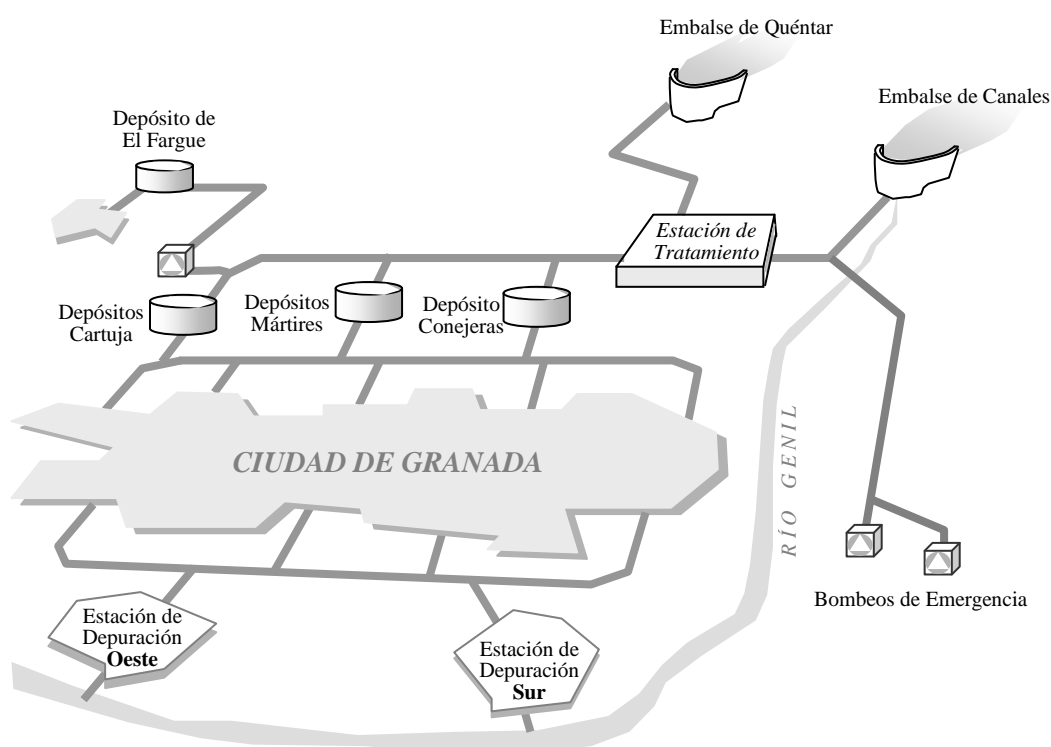


Figura 3.2: Estructura general del sistema de abastecimiento y saneamiento de la ciudad de Granada

El agua llega a la ETAP donde se realiza su tratamiento y potabilización, y es enviada hacia los depósitos de la red de distribución.

En la red de distribución se pueden diferenciar dos partes: red de alta y red de baja. La red de alta distribuye el agua en los depósitos de regulación, desde donde se envía hacia las reductoras y la arteria, o mediante bombeos a otros depósitos secundarios. La red de baja redistribuye el agua en sectores más pequeños desde donde llega a los abonados o usuarios finales.

La red de saneamiento recoge las aguas residuales que se canalizan hacia los diferentes colectores que terminan su recorrido en cabecera de las dos depuradoras de aguas residuales.

El ciclo termina con la devolución del agua depurada al río. Esta es la estructura general de la gestión integral del agua en grandes rasgos. A continuación se detallarán los diferentes procesos productivos.

3.3.2 Captaciones

La Ciudad de Granada abastece su red de agua de tres fuentes:

- Embalse de Canales (70 Hm³).
- Embalse de Quéntar (14 Hm³).
- Bombeos de Emergencia.

Los embalses, que recogen las aguas de los ríos Genil y Aguas blancas, se utilizan tanto para abastecimiento de agua potable como para regadío.

Los bombeos explotados están ubicados en la ronda sur de la ciudad y están compuestos por 10 pozos de 230 metros de profundidad con bombas de 540 CV de potencia y un caudal nominal de 130 l/s, lo que permite una capacidad de abastecimiento de 1.300 l/s.

Los diez pozos están conectados a una tubería común que permite elevar el agua hasta la cabecera de la estación de tratamiento de agua potable.

3.3.3 Estación de tratamiento de agua potable

La estación de tratamiento de agua potable (en adelante ETAP), ver Figura 3.3, recoge las aguas brutas provenientes tanto de los embalses como de los bombeos de emergencia y las somete a los tratamientos convencionales de preoxidación, decantación, filtración, y desinfección.

El conjunto de la ETAP no corresponde a un diseño único, como es normal en otras estaciones de tratamiento, sino que se compone de tres plantas de tratamiento diferentes, construidas cada una de ellas en épocas muy distantes en el tiempo, por lo que coinciden en el mismo recinto tres tecnologías diferentes respecto al tratamiento de agua potable.

En la Planta de Tratamiento se tienen diversos sistemas interrelacionados entre sí:

- Entrada y bombeos de aguas brutas.
- Reparto de aguas brutas.
- Planta de tratamiento I (800 l/s).
- Planta de tratamiento II (650 l/s).
- Planta de tratamiento III (1.000 l/s).
- Planta de cloración.
- Planta de reactivos.
- Servicios auxiliares: Agua sanitaria, aire a presión, generación energía de emergencia.
- Depósitos de almacenamiento.
- Salida de agua tratada.



Figura 3.3: Estación de tratamiento de agua potable (ETAP)

En conjunto, la planta potabilizadora (suma de las tres) es capaz de tratar 2.450 l/s.

Al margen de las tres plantas, y teniendo en cuenta que uno de los canales, concretamente el que viene del embalse de Quéntar, llega a la estación de tratamiento a una cota 104 metros superior a la de entrada a las plantas, se ha construido una central hidroeléctrica que permite convertir la energía hidráulica de los 1000 l/s que pueden llegar por dicho canal, en energía eléctrica, proporcionando hasta 905 Kw de potencia.

3.3.4 Red de alta

La red de alta (en adelante RA) es la parte de la infraestructura que permite la distribución del agua en los diferentes depósitos de almacenamiento de la ciudad (ver Figura 3.4). El agua se distribuye por dos canales de abastecimiento que, saliendo de la planta potabilizadora, alcanzan los depósitos principales o de cabecera:

- Conejeras.
- Mártires.
- Cartuja.

Desde estos depósitos de cabecera parten las arterias básicas (pasando por reductoras) que transportan los caudales demandados por los distintos sectores de la red. El agua se distribuye según el principio que rige los vasos comunicantes. Como a causa de la topografía del terreno es imposible desplazar el agua sólo en virtud de la fuerza de la gravedad, para asegurar el suministro se han instalado en puntos estratégicos las siguientes estaciones de bombeo que permiten el abastecimiento a las zonas más altas de la ciudad:

- Bombeo a depósito de Bola de Oro.
- Bombeo a depósito de Silla del Moro.
- Bombeo a depósito de San Miguel y a depósito de Beiro.
- Bombeo a depósito de Nuevo de San Miguel y a depósito de El Fargue.

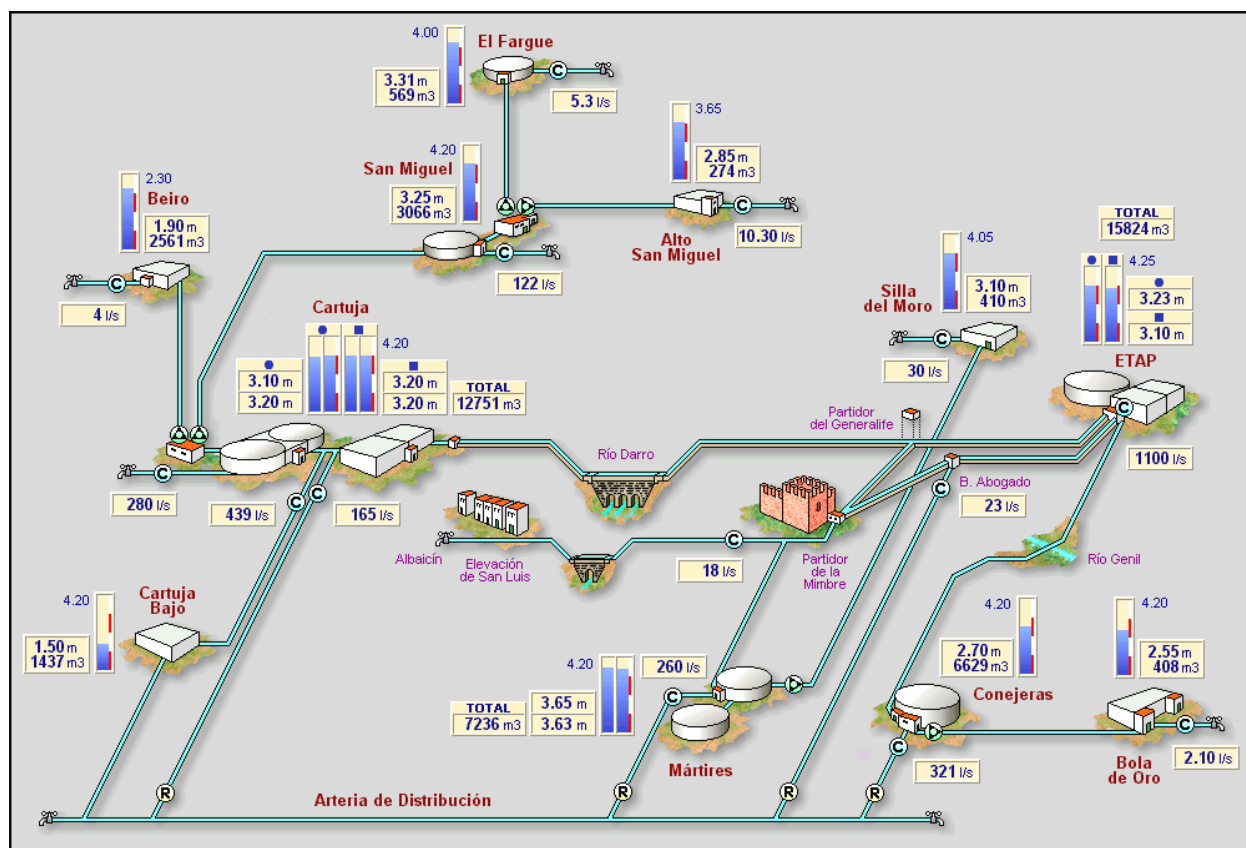


Figura 3.4: Red de alta en la distribución de agua

3.3.5 Red de baja

La red de baja (en adelante RB) empieza o bien a la salida de los mismos depósitos (para alimentación de sectores) o a través de las tres reductoras principales (Cartuja, Mártires, y Conejeras) hacia la arteria general de distribución, que en definitiva no es más que un anillo cerrado del cual derivan una gran parte de los sectores.

En esta parte de la red de transporte, según el tamaño y la función que cumplen, existen tres tipos de tuberías:

- Tuberías principales o arterias: a ellas no deben conectarse las derivaciones para los usuarios (acometidas) ni las bocas de riego.
- Tuberías secundarias: transportan el agua por las calles.
- Tuberías derivadas: transportan el agua al interior de los edificios.

La RB está dividida en sectores (ver Figura 3.5), de ahí que también se la denomine "sectorización". Cada sector tiene un único punto de entrada, lo que permite la lectura de diversas variables físicas (caudal y presión fundamentalmente, y cloro en algunos de ellos). Granada está dividida en la actualidad en 43 sectores individuales.

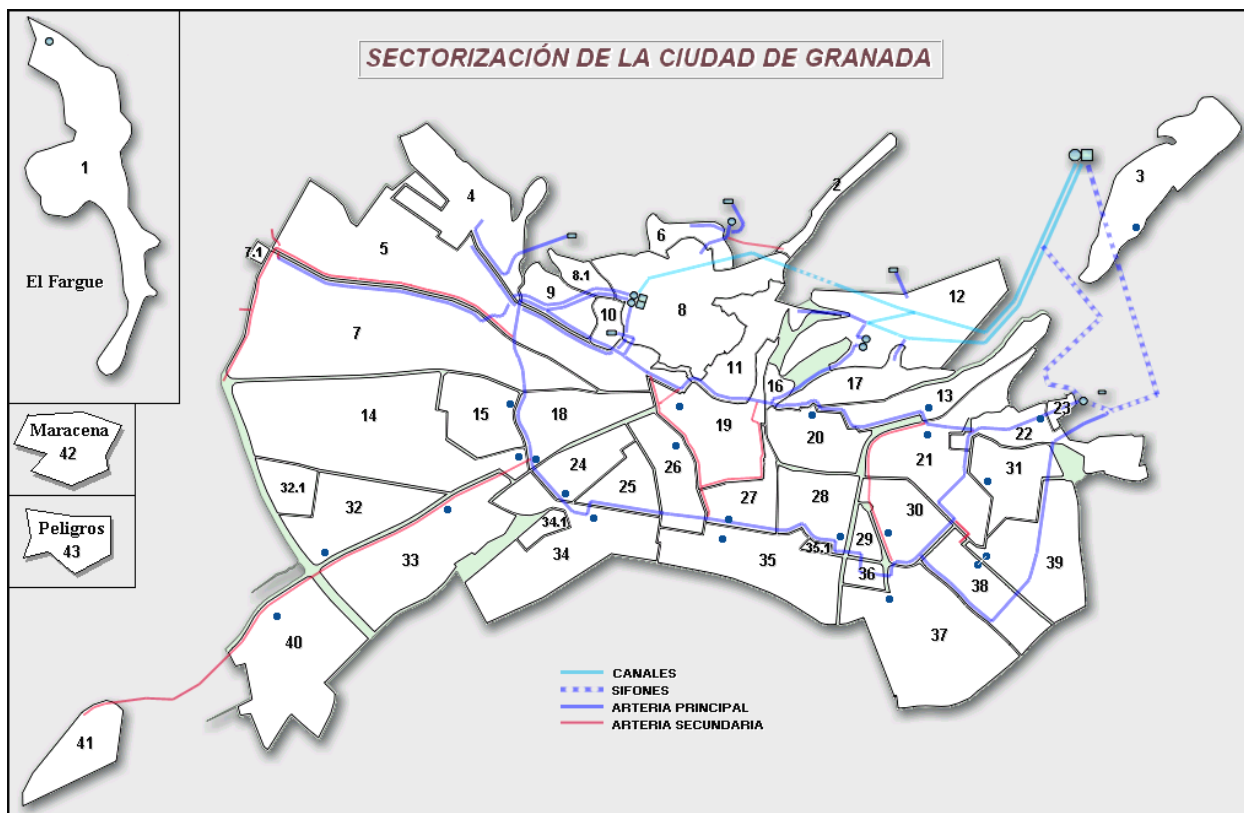


Figura 3.5: Red de baja (sectorización) en la distribución de agua

Existen además unas válvulas que permiten cortar el agua para aislar un sector que necesite ser reparado, y en algunos sectores se ha instalado además una reductora de presión que permite ajustar de forma individual la presión en el sector y bajarla (cuando haya poco consumo) para así reducir posibles fugas.

3.3.6 Red de saneamiento

La red de saneamiento (en adelante RS) recoge las aguas residuales urbanas y son conducidas hacia los colectores principales que desembocan en las dos depuradoras. Funciona por gravedad, salvo casos excepcionales donde se necesita un bombeo. Además, las tuberías de saneamiento están situadas por debajo de las de agua potables, para evitar que cualquier pérdida o fuga pueda contaminar el agua.

Esta red también tiene tuberías de diversos tamaños, que aumentan de grosor conforme recogen residuos de zonas más extensas. Este sistema recoge también el agua de lluvia.

La RS es controlada por varios caudalímetros situados estratégicamente que permiten conocer el nivel de carga de los diferentes colectores.

3.3.7 Estaciones depuradoras de aguas residuales

Y para terminar con las infraestructuras necesarias en el ciclo integral del agua de una ciudad, está claro que corresponde al hombre reparar el daño infligido a la naturaleza, y que para ello hay que crear también tecnología en el tratamiento y distribución del agua usada.

A una estación depuradora de aguas residuales (en adelante EDAR) le llega el agua ya utilizada en las casas, fábricas y comercios, que lleva gran cantidad de productos de desecho, y que por ello se le denomina *agua residual*: aguas domiciliarias (de 80 a 120 l/hab. día), aguas de limpieza pública y riego, aguas pluviales, etc.

Antes de devolverla al río es preciso depurarla para evitar desequilibrios ambientales. *Depurar* consiste en eliminar las materias orgánicas, inorgánicas y microorganismos patógenos que hay en el agua [Her-92].

En Granada existen dos cuencas en las que se vierte el agua residual de la ciudad:

- EDAR Sur (capacidad de 52.000 m³/día)
- EDAR Oeste (capacidad de 22.500 m³/día)

Para depurar el agua se siguen, habitualmente, cuatro tipos de procesos:

- Separación mecánica de sólidos de gran tamaño, objetos flotantes y grasas,
- Separación por medios físicos de partículas en suspensión (decantación),
- Tratamiento biológico,
- Desinfección.

En concreto, las dos depuradoras construidas en Granada integran los siguientes procesos: agua bruta y elevación, pretratamiento, decantación primaria, biológico y clarificación, cloración (opcionalmente), espesamiento de lodos primarios, flotación de lodos en exceso, digestión

primaria y secundaria, secado mecánico, y línea de gas. En la Figura 3.6 se puede observar una de estas EDARs.



Figura 3.6: Estación depuradora de aguas residuales (EDAR)

3.4 Tablas de sistemas y esquema general del abastecimiento

A continuación se presentan una serie de tablas, Figura 3.7 y Figura 3.8, que muestran la infraestructura y la instrumentación instalada para cada sistema de la red de distribución de agua de Granada.

Sistema: CAPTACIONES			
Subsistema	Capacidad	Tipo de Presa	
Embalse de Canales	73 Hm ³	Escollera	
Embalse de Quéntar	13,5 Hm ³	Bóveda-Hormigón	
Contraembalse de Canales	175.400 m ³	Hormigón-Gravedad	
Subsistema	Cantidad de sondeos	Caudal nominal (l/s)	Lugar de entrega
Bombes de Emergencia	10	1000	Llegada a ETAP
Bombeo de Sancho Panza	1	100	Depósito de Conejeras
Bombeo de Primavera	1	60	Inyección directa a la Red
Totales	12	1160	

Sistema: CANALES DE TRANSPORTE				
Subsistema	Origen	Entrega	Longitud (m)	Capacidad Transporte (l/s)
Canal de Pino	Salida Contraembalse	ETAP, Planta I	7.000	1.100
Canal de Loaysa	Salida Contraembalse	ETAP, Canal Quéntar	8.650	3.600
Canal de Quéntar	Pantano de Quéntar	ETAP, Planta II	14.000	2.000
Tubería de Bombes Emergencia	Sondeos Ronda Sur	ETAP, Canal Quéntar	4.870	1.300

Sistema: ETAP										
Subsistema	Capac. Total	Caudal medio	Población abastecida	Depósitos	Bombas	Compresores	Dosificadores	Transd. Caudal	Clorómetros	Detec. Cloro
General	2.450 l/s	1.014 l/s	≈ 300.000	2	5	3	12	1	4	2
Subsistema	Decantadores		Filtros	Válvulas	Bombas	Soplantes		Transduc. caudal		
Planta I	4		20	-	2	-		1		
Planta II	2		4	4	3	2		1		
Planta III	2		5	20	5	2		1		
Totales	8		29	24	10	4		3		

Sistema: EDAR									
Subsistema	Habitantes	Caudal nominal (m ³ /d)	Decantadores	Filtros	Digestores	Bombas	Compresores	Calderas	Intercambiadores
EDAR Sur	184.970	51.237	3	3	1	21	3	2	2
EDAR Oeste	80.970	22.416	2	3	1	21	5	1	1
Totales	265.940	73.653	5	6	2	42	8	3	3

Figura 3.7: Tabla-1 de infraestructuras e instrumentación del abastecimiento de agua de Granada

Las tablas de la Figura 3.7 corresponden a subestaciones con sus propias subcentrales de supervisión y control, pero que a efectos de la central global sólo actúan como estaciones remotas de las cuales sólo interesan para su supervisión ciertos estados y valores.

Sistema: RED DE ALTA											
Nº de Estación	Descripción	Depósitos	Capacidad Total (m ³)	Válvulas	Reductoras	Compuertas	Nº de bombas	Transduc. Nivel	Transduc. Caudal	Transduc. Presión	Sensores Presencia
2	Depósito Bajo de Cartuja	1	3.833	4	2			1			1
4	Depósito de Conejeras y Bombeo de Bola de Oro	1	9.820	5			2	1	1	1	1
5	Depósito Silla del Moro, partidor del Generalife y partidor de la Mimbre	1	535	3		8		6	1		3
8	Depósitos, bombeo y reductora de los Mártires	2	7.952	6	2		2	2	2	2	3
9	Depósito de San Miguel y nuevo Albayzín, y bombeos de El Fargue y nuevo Albayzín	2	4.123	6			4	2	3	2	3
10	Depósitos de Cartuja y Beiro, bombeos de San Miguel y Beiro, y partidor de Cartuja	5	19.166	17			5	7	4	2	4
15	Depósito de El Fargue	1	670	1				1	1		1
20	ETAP (como estación remota)	2	20.000	-	-	-	-	2	2	-	-
21	Reductora de Cartuja				2					1	1
23	Reductora de Conejeras				2					1	1
Totales		15	66.099	42	8	8	13	22	14	9	18

Sistema: RED DE BAJA						
Subsistema	Diámetro (mm)		Longitud (m)		Válvulas	
Arteria Principal Fibrocemento	800		8.464		11	
	600		1.000		4	
	500		1.857		8	
	400		7.600		3	
	300		960		6	
Arterias Secundarias Hormigón C. Chapa	1000		1.063		-	
	600		7.464		-	
	450		632		-	
Subsistema	Diámetro Max.(mm)	Diámetro Min.(mm)	Longitud Red (Km.)	Válvulas	Acometidas	Bocas de riego
Distribución	600	60	480	4.815	18.776	8.017
Subsistema	Nº de Sectores	Nº de Subsectores	Caudalím. Electromag.	Caudalím. de Pulsos	Transmisores de Presión	Sensores Presencia
Sectorización	43	5	24	22	22	34

Figura 3.8: Tabla-2 de infraestructuras e instrumentación del abastecimiento de agua en Granada

En las tablas de la Figura 3.8 se muestran las cifras aproximadas de aquellos equipos e infraestructuras que se han considerado más significativas, aunque en la realidad hay muchos más sistemas que también son supervisados y controlados de forma remota, como por ejemplo: fallos de alimentación, disponibilidad de los equipos, indicadores de funcionamiento (manual / automático, local / remoto), sondas de nivel todo/nada, detectores de flujo de caudal, medidores de intensidad, finales de carreras, fallos en los caudalímetros, medidores de cloro, y estados de electroválvulas, entre otros.

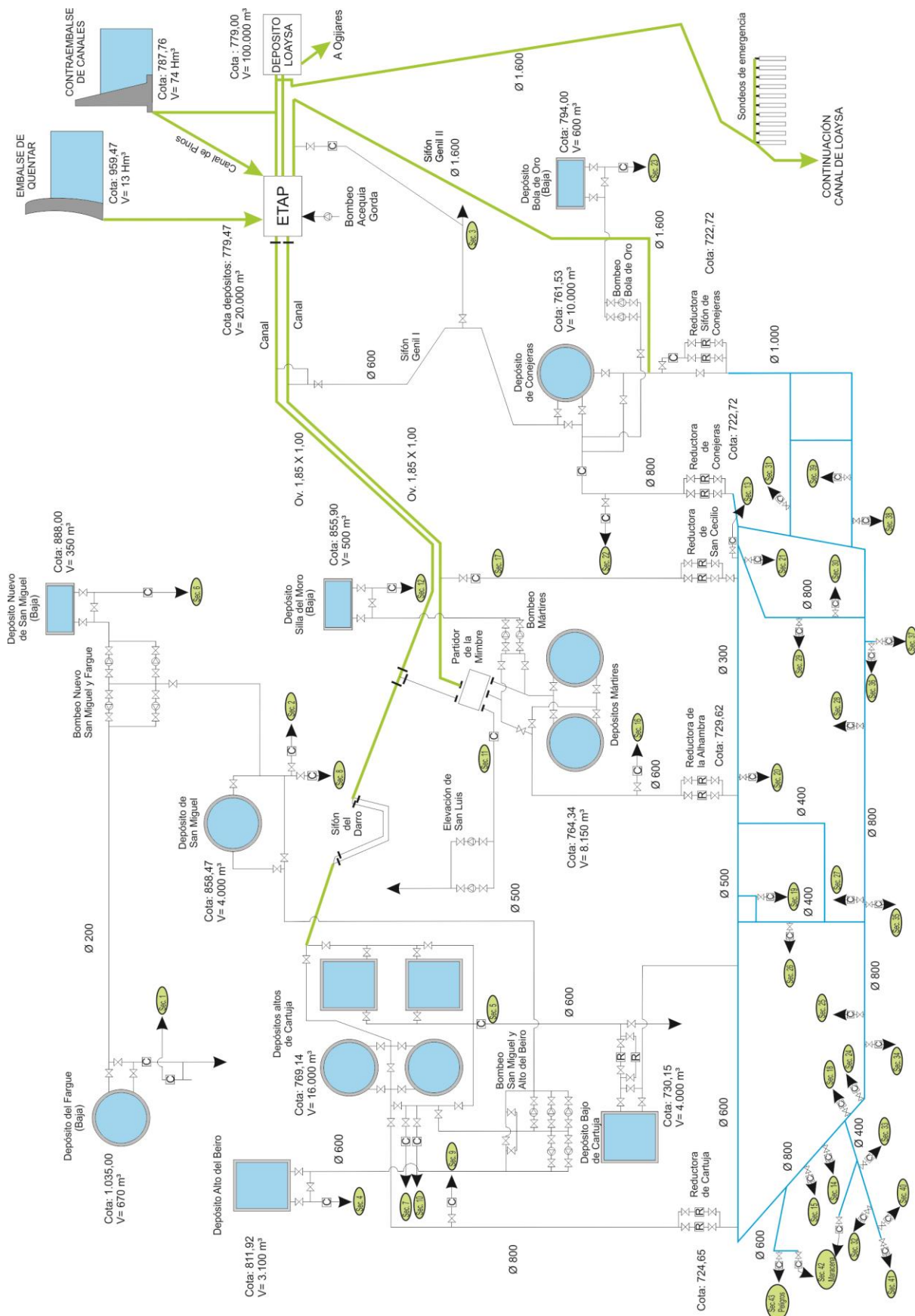


Figura 3.9: Esquema general del abastecimiento a Granada (cortesía de Emasagra)

Capítulo 4:

Sistema de control para la red de distribución de agua de la ciudad de Granada

"Todo el ciclo del agua controlado por ordenador" es el título de un artículo publicado recientemente en un periódico de la provincia de Granada, que describe de forma muy concisa el contenido de este capítulo, donde se muestra en conjunto todo el trabajo realizado, y en especial los aspectos relacionados con los equipos de adquisición, control y comunicaciones instalados en la ciudad de Granada para la explotación de un recurso imprescindible como es el agua.

Se comienza, sección 4.1, recordando cuando y cómo se introdujeron los sistemas de adquisición y control en la gestión del agua en esta ciudad, qué se ha conseguido mediante estos sistemas, y cómo se prevé el futuro.

A continuación, sección 4.2, se hace una descripción general y esquematizada del proceso en su totalidad y de los subprocesos en particular, para pasar ya en la sección 4.3 a mostrar los equipos y programas diseñados para la central, y en la sección 4.4, los utilizados en las estaciones remotas.

Por último, sección 4.5, se tratan los aspectos relacionados con el sistema de comunicaciones utilizado; en concreto se justifica el uso tanto de las comunicaciones por radio como del sistema SPIRAL.

4.1 Pasado, presente y futuro

Los sistemas de control modernos comenzaron a utilizarse en la gestión del agua de Granada a finales de los 80, surgiendo como necesidad de controlar una pequeña estación de tratamiento en una barriada alejada, El Fargue, que requería excesivos recursos humanos para una población pequeña.

En aquella ocasión se instaló un PLC local que gobernaba las válvulas de entrada y salida, el bombeo a dicha estación, y las lecturas de la turbidez del agua bruta y del cloro libre en agua tratada. Las señales de esta estación eran enviadas vía radio a una sala de control en la ETAP, desde la que se vigilaban los parámetros anteriores además del estado de las bombas y los posibles eventos que se produjeran [Gom-94].

Con esta instalación, se reducía considerablemente el tiempo de personal dedicado a esta pequeña población, y por otra parte el control pasaba a ser continuo.

Mientras esta experiencia daba sus frutos, tanto desde el área de tratamiento como desde el área de explotación, se comenzaban a realizar en paralelo las obras pertinentes encaminadas a:

- La modernización de la estación de tratamiento de agua potable (ETAP) y sus sistemas hidráulicos y eléctricos.
- El desarrollo de una reforma muy importante: la división en sectores de una red de distribución de agua extremadamente mallada.

La habilitación de estas infraestructuras era absolutamente necesaria para la futura incorporación de los sistemas de control en ambas áreas.

A partir del año 91 se inició la construcción de la tercera planta de tratamiento de agua potable y la integración de todos los sistemas existentes en las dos plantas anteriores en un sistema de control global para la ETAP; se comenzó con el desarrollo del telecontrol de la red de abastecimiento, y se finalizó la construcción de las dos depuradoras de agua residuales (EDAR Sur y EDAR Oeste), a las que posteriormente también se incorporaron sistemas de control.

Las sucesivas fases de control se desarrollaron como consecuencia evidente de los éxitos obtenidos en las fases anteriores.

En la actualidad, se controlan el 100% de los procesos productivos críticos de Emasagra; es decir, se puede afirmar que el 100% de los procesos relevantes son controlados de forma distribuida. Se incorporarán nuevas instalaciones en sucesivas fases, que se acometerán en los próximos ejercicios.

Hace 10 años era impensable hablar, por poner algunos ejemplos, de:

- Regulación automática del nivel de los depósitos de la red de alta.
- Regulación y conmutación automática en las reductoras principales de abastecimiento.
- Detección de fugas en los sectores de la ciudad.
- Reducción automática de los costes en el consumo eléctrico de los bombeos.
- Predicción de demandas de consumo diarias.
- Detección de averías en la red desde la sala de control.
- Dosificación de coagulante proporcional al caudal de agua bruta.
- Regulación del oxígeno disuelto a las balsas de aireación con un doble lazo de control en cascada.
- Generación de partes de explotación automáticos derivados de los resultados obtenidos y almacenados por las diferentes aplicaciones de control.

Actualmente estos problemas se han resuelto en gran parte con los sistemas desarrollados [Dam-98]; así, en breve pretendemos conseguir nuevos objetivos tales como:

- Integración de la cartografía para observar en tiempo real un mapa de presiones o concentraciones de cloro.
- Telelectura de contadores domésticos.
- Mayor nivel de optimización en costes y en funcionamiento automático.
- Comunicaciones más rápidas y seguras.
- Incluso, por que no, disponer de un sistema único y autónomo (sin necesidad de operadores) para la explotación global del ciclo integral del agua.

4.2 Descripción general del sistema

El sistema desarrollado permite disponer de un control centralizado y en tiempo real sobre el ciclo integral del agua en la ciudad de Granada. La sala de control central está ubicada en las oficinas principales de la empresa EMASAGRA, desde donde se establece comunicación vía radio (Sistema Spiral [Gom-94]) con un conjunto de estaciones remotas controladas por PLCs y otras basadas en el sistema de telecontrol distribuido SPIRAL-5000. El software de aplicación se ha desarrollado basándose en un paquete SCADA.

Los principales elementos del ciclo integral del agua de la ciudad de Granada que hasta el momento son supervisados en tiempo real son:

- **Red de alta** formada por 14 depósitos, 6 bombeos y 2 partidores o bifurcaciones, con: válvulas de apertura y cierre (control continuo o todo/nada), válvulas reguladoras, compuertas (control continuo o todo/nada), medidores de consumo de energía, caudalímetros, medidores de nivel, medidores de presión, sondas de máximo y mínimo, y detectores de presencia.
- **4 estaciones reguladoras de presión**, con dos reductoras de presión en cada una.
- **Red de baja** dividida en 45 sectores, con telemedida de caudal y presión (también cloro en algunos de ellos), y control de válvula (para la apertura/cierre del sector).
- **Red de Saneamiento** con medidas de nivel y turbidez (aprovecha las infraestructuras creadas para la red de alta y la red de baja).
- **Sondeos de emergencia** o bombeos de aguas subterráneas controlados vía radio desde la ETAP.
- Una **ETAP** (Lancha de Cenes) formada por tres plantas, de diferente época de construcción y por tanto de diferente tecnología, totalmente automatizadas mediante una red de 10 PLCs. Toda la supervisión (incluida la de una minicentral hidroeléctrica) se realiza desde la sala de control ubicada en la misma ETAP, que a su vez actúa como estación remota de la red de telecontrol principal.
- Dos **EDAR's** (Sur y Oeste) que se integrarán también en el sistema mediante líneas RDSI o enlaces microondas (*spread spectrum*).

Para describir el sistema de control que se ha desarrollado para todas estas instalaciones, se muestra a continuación una serie de esquemas gráficos y tablas de equipos de automatización, que nos muestran en primera aproximación, la infraestructura de hardware y comunicaciones empleada.

En primer lugar se proporciona información del sistema en conjunto mediante un diagrama de bloques (Figura 4.1), una tabla que presenta el sistema de control en cifras (Figura 4.2), y un esquema global más completo (Figura 4.3). A continuación, las siguientes figuras y tablas muestran una visión más detallada de cada subsistema.

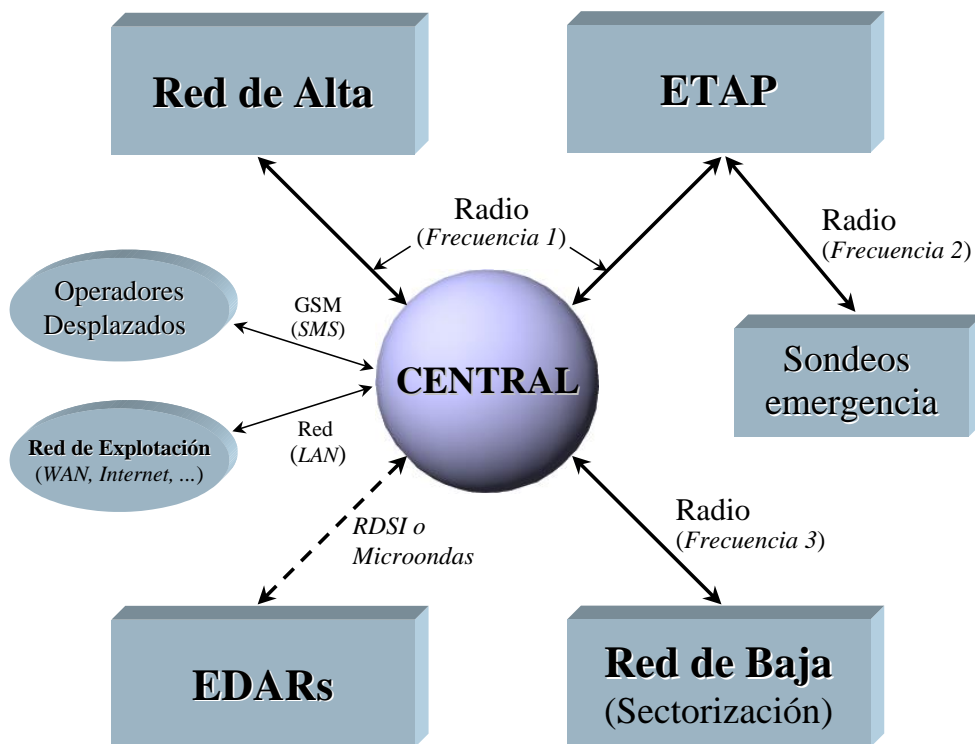


Figura 4.1: Diagrama de bloques para la integración de los elementos del Ciclo Integral del Agua en Granada

Procesos	Sistemas de control distribuido (PLCs y SP-5000)	E/S digitales	E/S analógicas
Red de Alta	15	1000	100
Sondeos emergencia	13	400	90
Red de Baja	24	200	100
ETAP	10	1500	150
EDARs	8	500	75
TOTAL	70	≈ 3600	≈ 515

Figura 4.2: Tabla del sistema de control en cifras para el Ciclo Integral del Agua en Granada

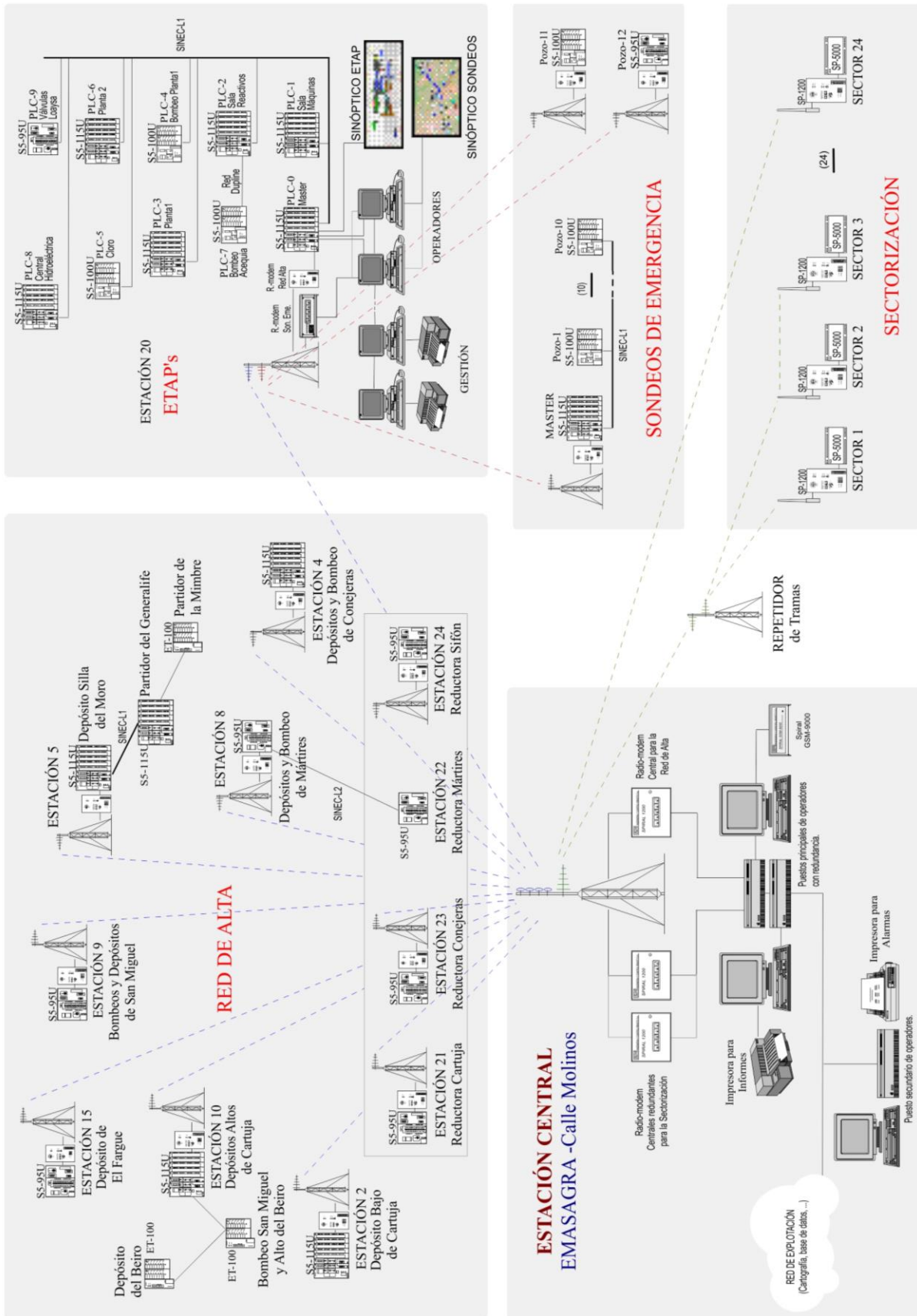


Figura 4.3: Sistema global de automatización y de comunicaciones en EMASAGRA

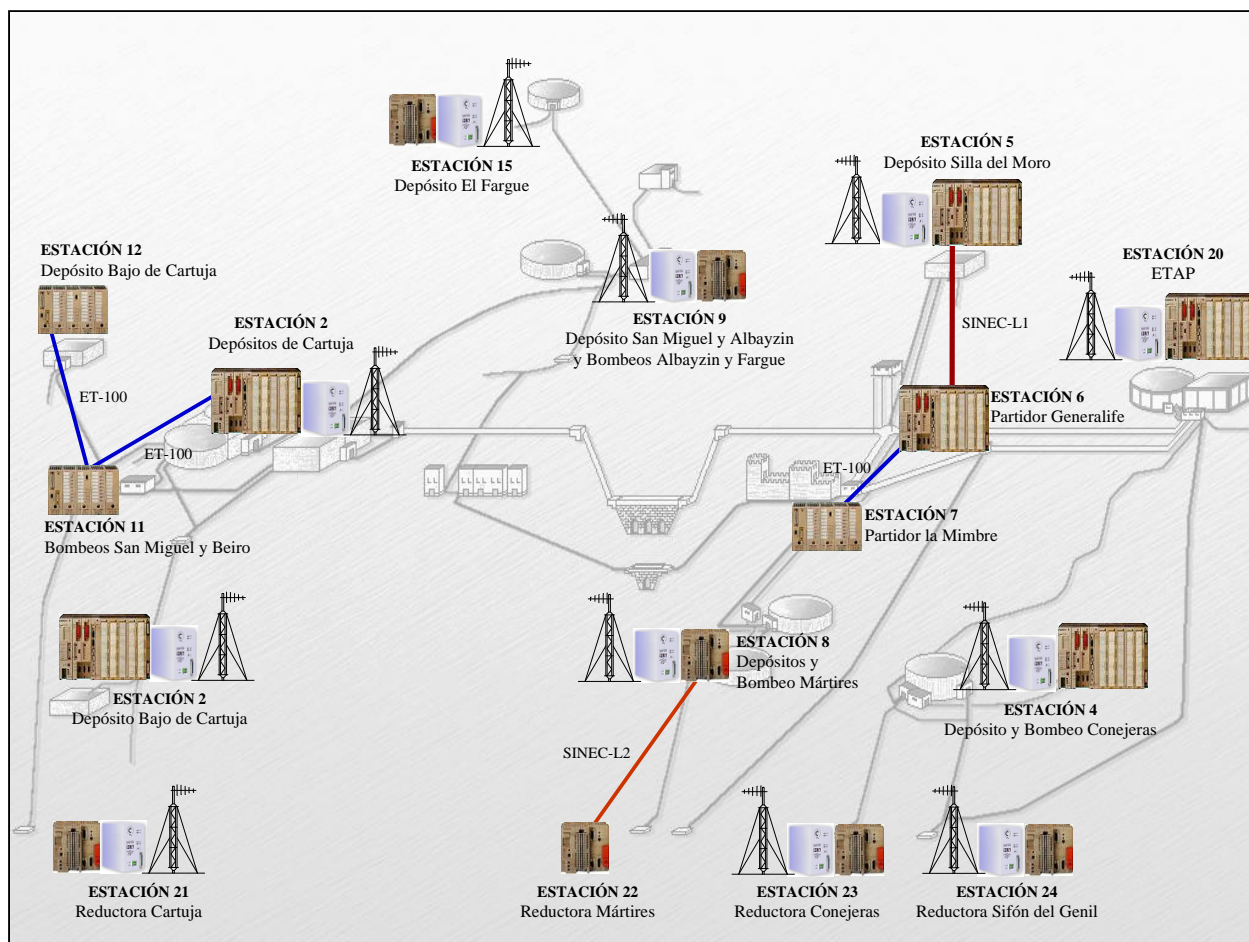


Figura 4.4: Sistema de telecontrol de la red de alta y reductoras

Como se puede observar en la Figura 4.4, en algunos puntos de la red cercanos a estaciones remotas enlazadas por radio se han empleado buses de campo (Siemens Sinec L1/L2) y periferia descentralizada (Siemens ET-100) aprovechando el cableado existente originalmente entre estas estaciones. Por tanto, las estaciones que comunican los datos por radio a la central, y que engloban a otras estaciones mediante comunicaciones por cable, necesitarán transmitir una trama de datos de mayor longitud que las restantes¹.

PLCs ²	Periferia descentralizada	E/S digitales	E/S analógicas
12	3	≈ 1000	≈ 100

¹ En el capítulo dedicado al gestor de comunicaciones distribuidas, se comprueba que este incremento en la trama provoca mayor cantidad de fallos de comunicaciones y se indica como se ha solucionado.

² No se considera el autómatas de la ETAP.

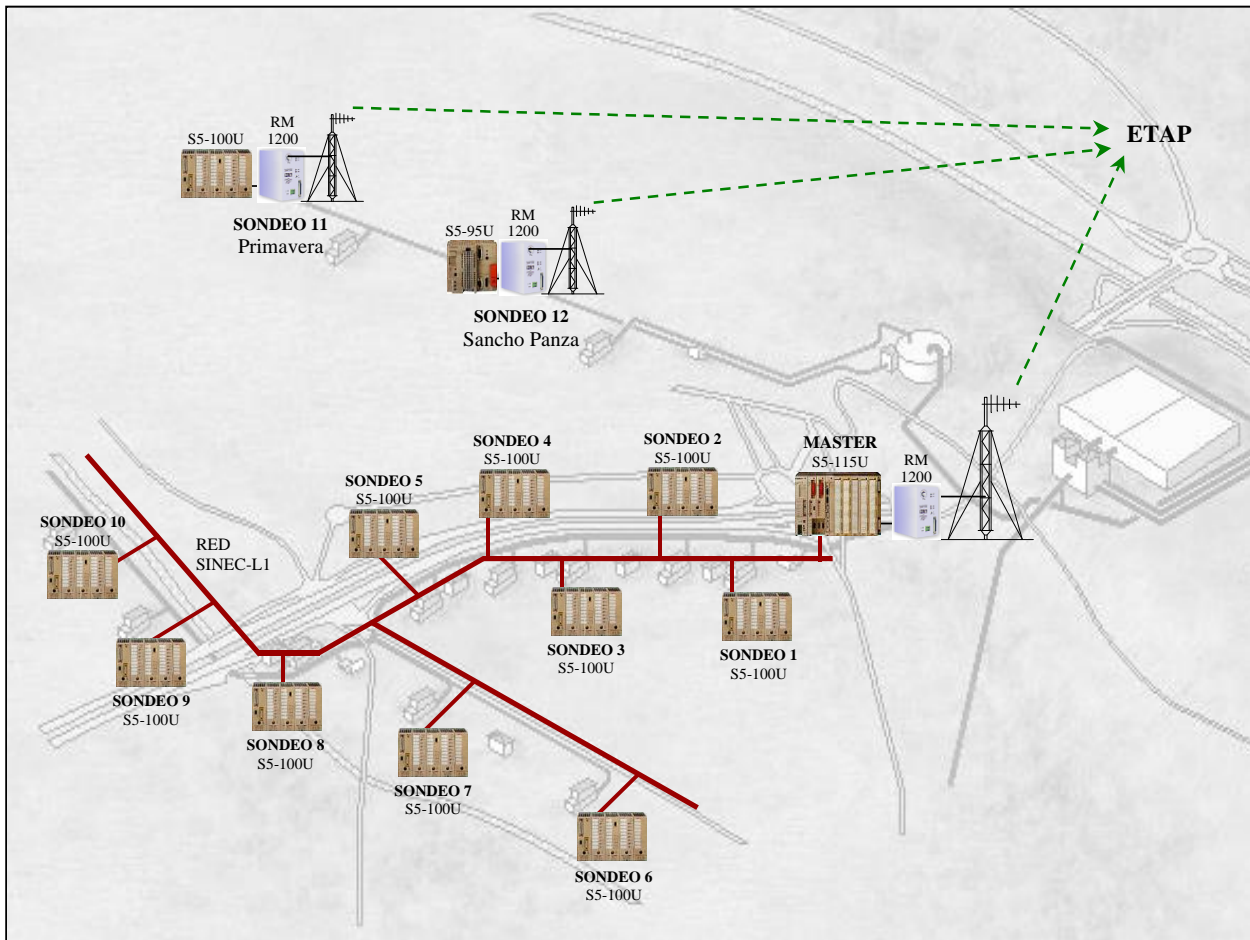


Figura 4.5: Sistema de telecontrol de los sondeos de emergencia

La Figura 4.5 muestra la red por cable (Siemens Sinec L1) utilizada para interconectar un PLC maestro con otros 10 esclavos encargados de realizar el control local de cada uno de los sondeos. El PLC maestro y otros dos, en sondeos más antiguos y alejados, se comunican vía radio con la central de supervisión y control situada en la ETAP, desde donde se configuran las condiciones de funcionamiento. También se puede realizar una supervisión de la información más relevante desde la central general de Emasagra.

PLCs	E/S digitales	E/S analógicas
13	≈ 400	≈ 90

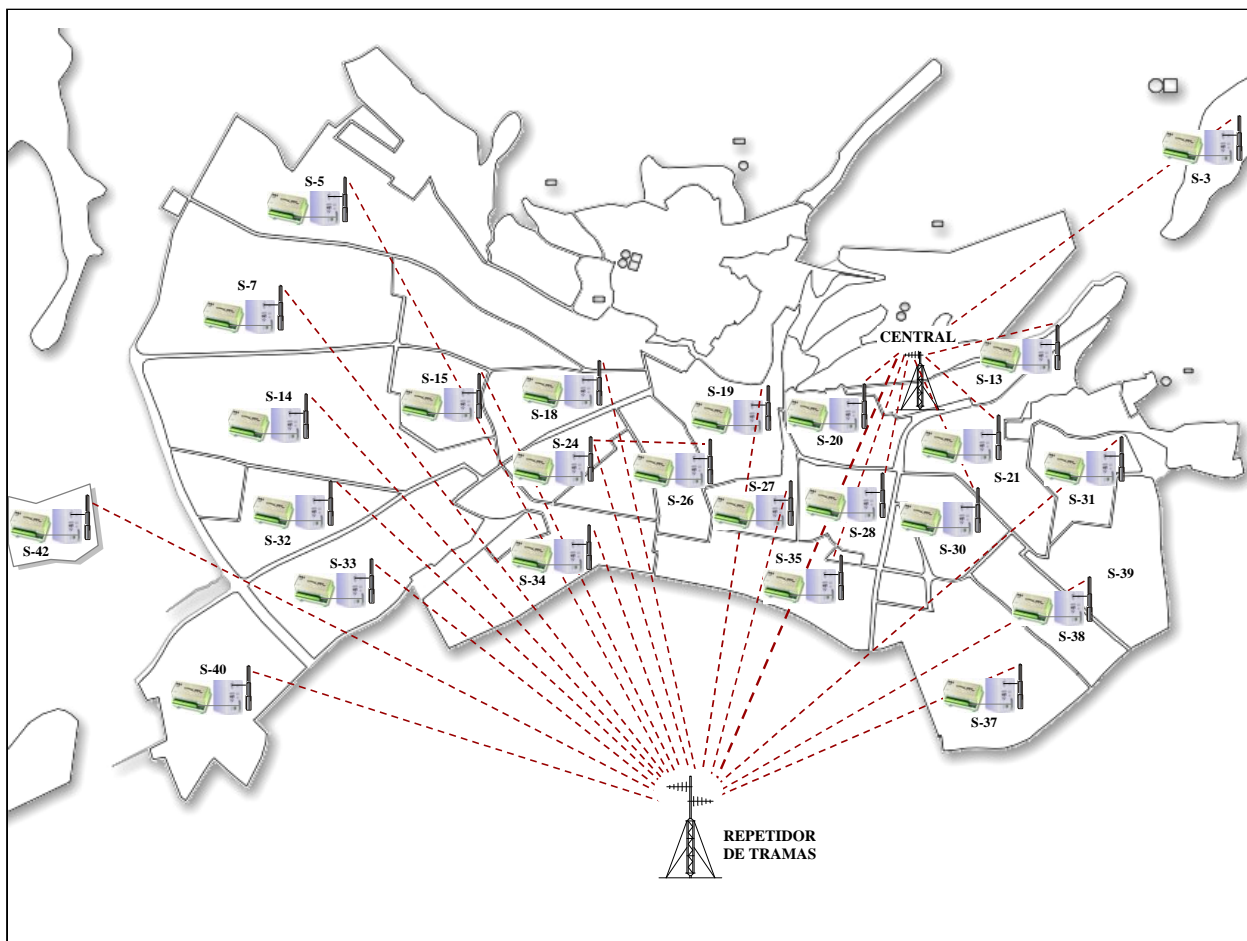


Figura 4.6: Sistema de telecontrol de la red de baja (sectorización)

En esta Figura 4.6 se puede observar la configuración de las comunicaciones vía radio que se ha realizado con el objetivo de adquirir los datos y operar en tiempo real. Se ha instalado un encaminador (*router*)³ para aquellas estaciones que presentaban problemas en las comunicaciones si se realizaba el enlace directamente con la central. También se muestra la posibilidad de realizar enrutamientos a través de otra estación en aquellos casos en los que no se puede enlazar ni directamente con la central ni con el repetidor (sector 26).

SP-5000	E/S digitales	E/S analógicas
24	≈ 200	≈ 100

³ Repetidor digital que redirige las tramas para aquellos sectores más alejados o con problemas en las comunicaciones vía radio.

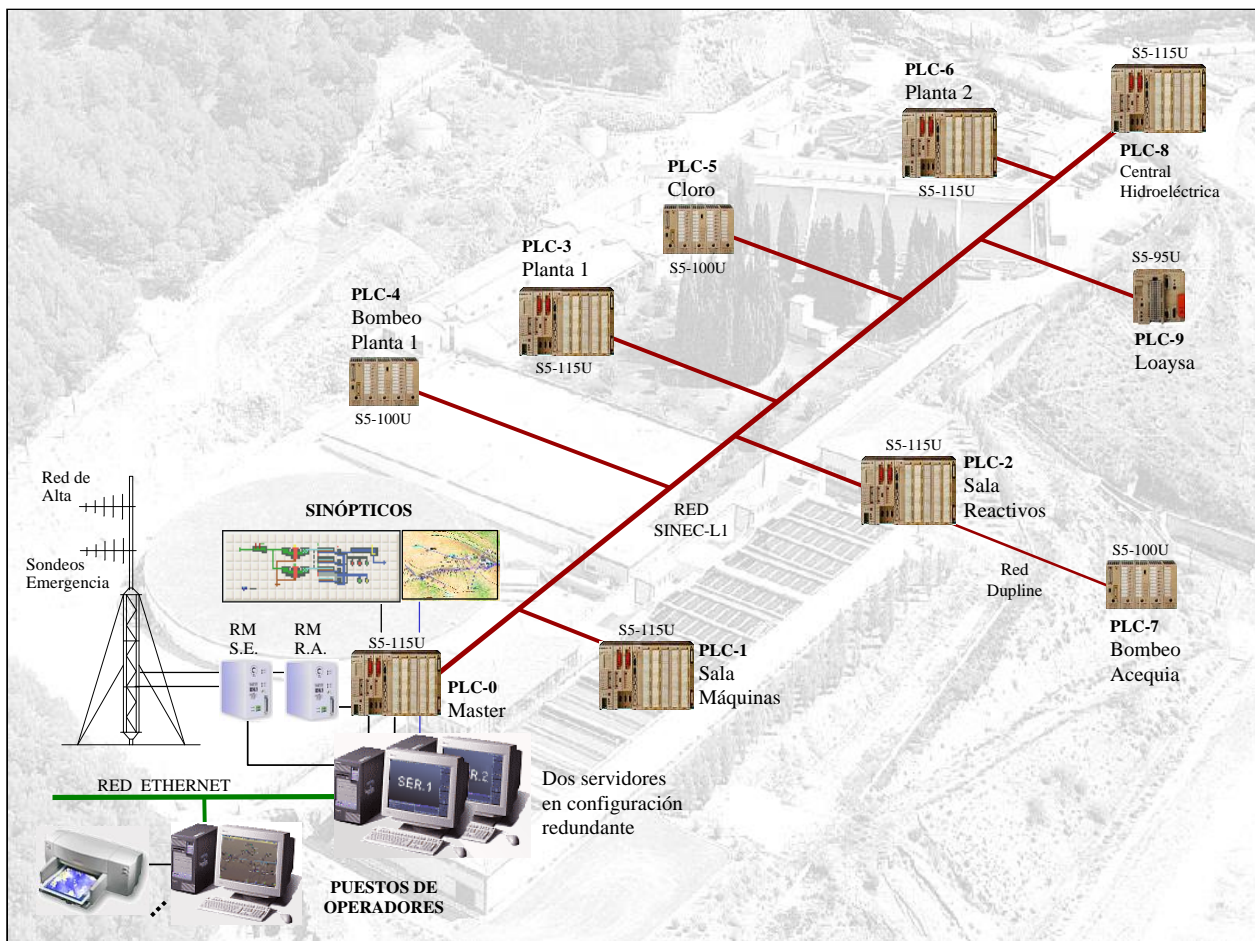


Figura 4.7: Sistema de control de la ETAP

La Figura 4.7 muestra la central de supervisión y control de la ETAP y de los sondeos de emergencia, así como los sistemas de automatización instalados para la explotación de las tres estaciones de tratamiento de agua de que dispone la ciudad de Granada. Como se puede observar, los dos ordenadores principales están conectados al PLC en configuración redundante, de forma que si uno de ellos falla se puede seguir operando con el otro.

Esta estación es a su vez una remota (estación 20) de la red de alta, para proporcionar a la central general de Emasagra la información más relevante del proceso de tratamiento, y para que en la ETAP a su vez se disponga de los niveles y caudales de la red de alta.

PLCs	E/S digitales	E/S analógicas
10	≈ 1500	≈ 150

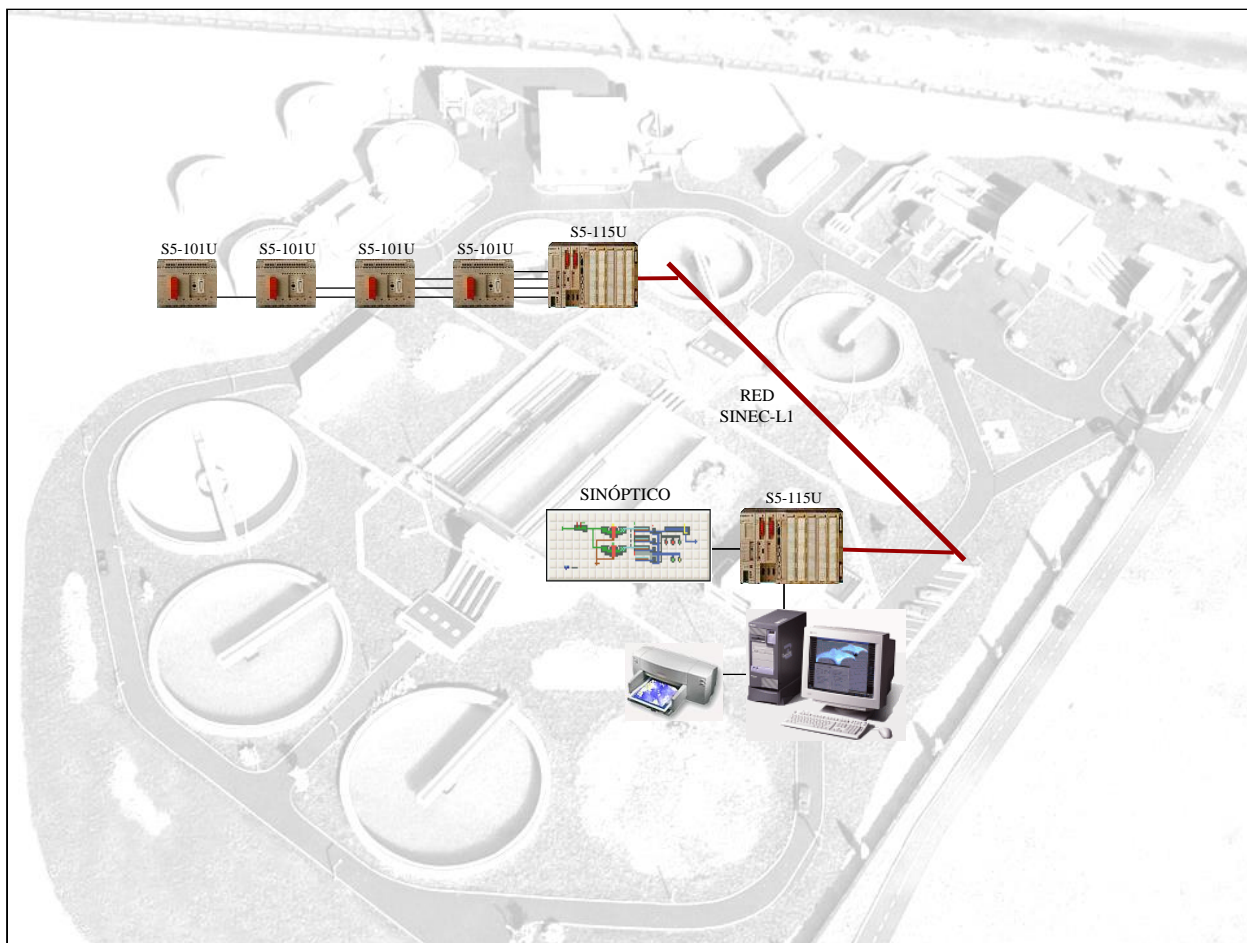


Figura 4.8: Sistema de telecontrol de una de las EDARs

En la Figura 4.8 se puede observar una de las dos EDAR (Sur de la ciudad) para la depuración de aguas residuales. La otra (Oeste de la ciudad) dispone de unas instalaciones similares, con dos PLCs, de ahí que no se muestren más detalles. Para transmitir la información a la central general de Emasgra se está pensando utilizar enlaces RDSI o microondas; en ambos casos se utilizaría una conexión TCP/IP para así acceder como un nodo cliente.

Los datos que se dan a continuación se refieren a los equipos de automatización de las dos EDARs en conjunto.

PLCs	E/S digitales	E/S analógicas
8	≈ 500	≈ 75

4.3 Estación central

La estación central, o centro de control, está compuesta por una sala principal con dos puestos de trabajo (REDES1 y REDES2), y una serie de puestos auxiliares distribuidos por el resto del edificio, denominados REDES3, REDES4, etc. En estos puestos de trabajo se puede operar simultáneamente, de manera que podrían estar distintos operadores trabajando con la aplicación al mismo tiempo. En general, el número de puestos de trabajo no está limitado, ya que el sistema se basa en una estructura cliente/servidor y la cantidad de clientes sólo depende de los puestos de operación que la empresa gestora necesite.

Uno de los dos puestos de trabajo de la sala principal actúa de servidor y por lo tanto está conectado a los radiomodem centrales Spiral-1200 [Gom-94] encargados de las comunicaciones por radio. Los otros puestos, clientes, reciben la información del proceso a través de la red local; de ahí que todos los puestos puedan visualizar y modificar la información independientemente.

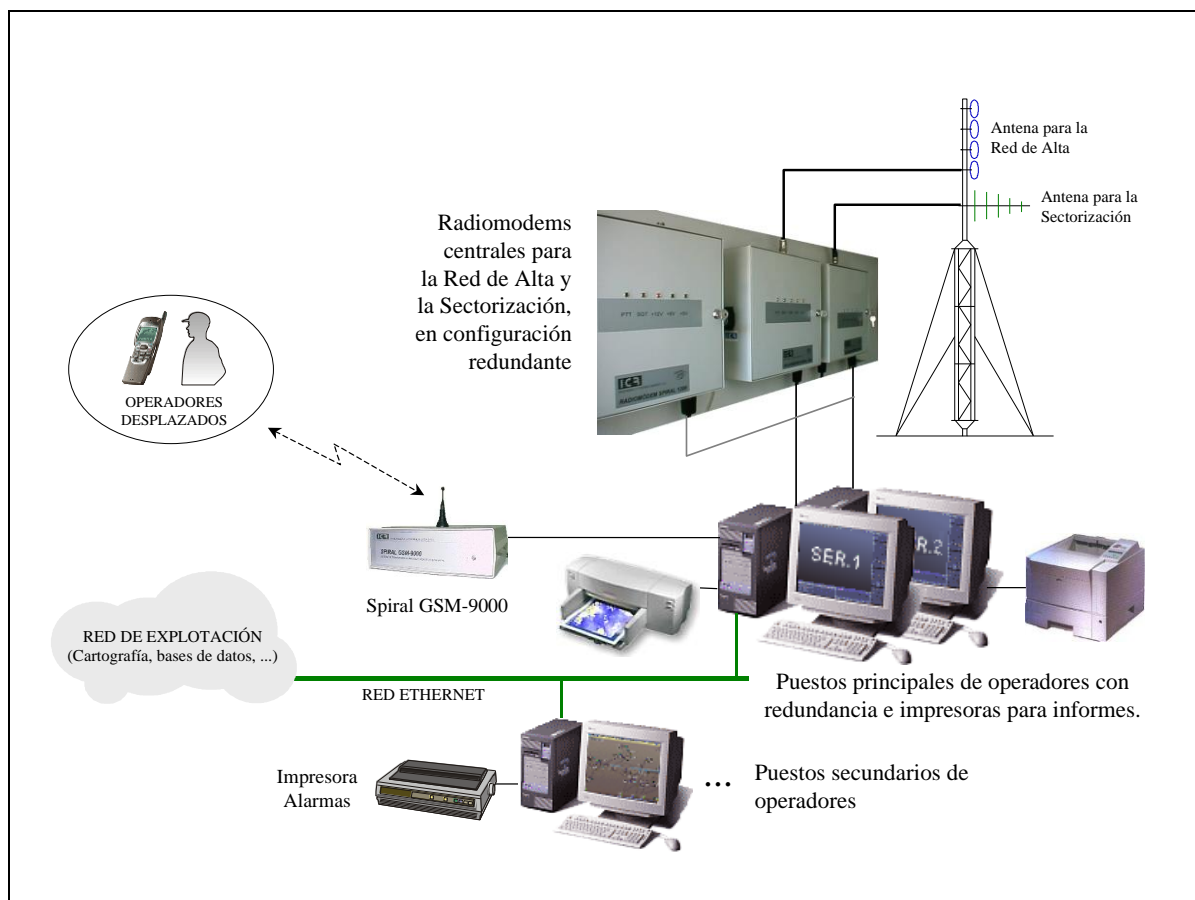


Figura 4.9: Estación central del sistema de control

Los computadores utilizados son PCs con la siguiente configuración: procesador Intel Pentium III, 64 MB de RAM, disco duro de 3.5 MB de capacidad, tarjeta gráfica de 64 bits con 2 MB SGRAM (o similar). De los dos puestos de la sala principal, uno tiene un lector de CD-ROM y otro de tipo magneto-óptico de 640 MB, así como una unidad ZIP de 100 MB.

Hay tres impresoras, una para cada puesto y con funciones diferentes:

1. Impresora de color de inyección de tinta (capaz de imprimir planos a A3) en el computador REDES1, para los informes diarios, semanales y mensuales, las gráficas de los históricos, los volcados de las pantallas gráficas de la aplicación, y los planos del sistema GIS.
2. Impresora Laser en REDES2, para imprimir informes varios desde programas de apoyo a la aplicación como Word, Excel, etc.
3. Impresora Matricial en el puesto REDES3 de la sala de operarios, para imprimir las alarmas y eventos que se producen en la aplicación.

Los puestos de la sala principal están configurados de forma redundante; es decir, si REDES1 (que actúa de servidor por defecto) tiene algún problema, se puede hacer que el puesto REDES2 (que por defecto actúa como cliente) actúe como servidor o puesto principal. Para esta operación se utiliza una de las funciones añadidas al gestor de comunicaciones desarrollado, y que se describe con más detalle en el capítulo siguiente.

El sistema operativo utilizado en los servidores es Windows NT versión 4 (*Service Packs* 6). En los clientes se puede utilizar tanto Windows NT como Windows 98. Se ha elegido esta plataforma por las siguientes razones:

- Su adopción por parte de los proveedores más importantes de software para adquisición de datos, supervisión y control de procesos (como se puede apreciar en la tabla sobre SCADA comerciales mostrada en el capítulo 2).
- Potencia gráfica, necesaria para la aplicación diseñada.
- Gran expansión, facilidad de utilización y disponibilidad de software.
- Permite trabajar con los nuevos estándares industriales ActiveX, OPC⁴, VBA y COM/DCOM⁵.

⁴ OPC = *OLE for Process Control*, norma que permite la integración de componentes de automatización de diversos fabricantes en las aplicaciones para PC.

⁵ COM/DCOM = *Component Object Model / Distributed Component Object Model*, tecnología de Microsoft para desarrollar aplicaciones distribuidas.

Para centralizar la información de todas las estaciones remotas se ha optado por las comunicaciones por radio y se han instalado tres radiomodems (ver Figura 4.10) para:

- Las comunicaciones con la red de alta o abastecimiento,
- Las comunicaciones con la red de baja o sectorización, y
- Un tercero en configuración redundante, de manera que si falla alguno de los anteriores se puede conmutar a este sin más que cambiar los cables de comunicación y de antena.



Figura 4.10: Instalación de radiomodem centrales

En la central también se ha instalado un sistema de alarmas vía GSM⁶ (sec. 4.3.1 para más detalles) que llama automáticamente al teléfono móvil configurado cuando se produce una alarma importante en la aplicación, y que permite realizar consultas o modificaciones de variables de la aplicación de forma remota.

El sistema de bombeos de emergencia se supervisa y controla desde una subcentral instalada en la ETAP por cuestiones de operatividad. Por lo tanto, el puesto de control se encuentra en estas instalaciones y en la central sólo se dispone de un resumen con la información más relevante.

⁶ GSM = *Global System for Mobile communications*.

4.3.1 Sistema de transmisión de mensajes por telefonía móvil

4.3.1.1 Introducción

Una de las necesidades de los sistemas de producción, como por ejemplo el agua en nuestro caso, es la de aprovechar los recursos humanos de forma eficiente y aumentar la operatividad del sistema, con el fin de ahorrar costes y al mismo tiempo dar mejor servicio. En este sentido, ampliar las posibilidades de acceso al sistema para aumentar la disponibilidad de los operadores y dotarles de mayor libertad de movimiento, es una forma de conseguir las prestaciones mencionadas.

La solución adoptada en nuestro diseño para esta necesidad, el sistema Spiral GSM-9000 (Figura 4.11), está basada en la transmisión de mensajes por telefonía celular digital, y en la utilización de modems y teléfonos móviles GSM.



Figura 4.11: Fotografía sistema Spiral GSM-9000

Dicho sistema permite, de forma manual o automática, generar una llamada al teléfono móvil especificado para avisar de cualquier avería, evento o aviso en general que se desee notificar a los operarios desplazados fuera de la central de control y supervisión, y además poder realizar, desde cualquier teléfono móvil autorizado, peticiones o actualizaciones de variables de una aplicación.

Concretamente, utiliza el servicio de mensajería (SMS⁷), de manera que se pueden enviar textos de hasta 160 caracteres que se visualizan en la pantalla de los teléfonos móviles. Dicho servicio de mensajería es un sistema bidireccional confirmado, que permite enviar y recibir mensajes alfanuméricos. El teléfono le avisará, además de con una señal sonora, con un icono en forma de sobre o con un texto en la pantalla. Además del contenido, también se recibirá el número de teléfono del remitente, y la fecha y hora en la que fue enviado el mensaje. Los mensajes que, por cualquier causa, no puedan ser entregados inmediatamente serán almacenados para su entrega inmediata en cuanto sea posible.

⁷ SMS = *Short Message Service*. Servicio de mensajes cortos punto a punto del sistema GSM.

4.3.1.2 Descripción general

Actualmente, el programa que se ha realizado, cuya ventana principal se muestra en la Figura 4.12, permite fundamentalmente:

- **Enviar un mensaje** escrito por el operador a cualquier teléfono GSM.
- **Conectar con cualquier aplicación** que actúe como servidor DDE, como por ejemplo el SCADA Intouch (Wonderware).
- **Generar una llamada de forma automática** a los teléfonos especificados cuando se activa una de las condiciones de transmisión definidas por el usuario.
- **Interactuar con el operador** mediante un protocolo de intercambio de comandos de fácil uso, entre el operador del teléfono móvil y la estación receptora, tanto para solicitar información como para actualizarla remotamente.

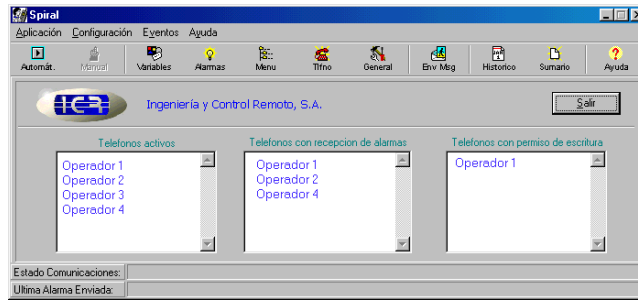


Figura 4.12: Pantalla principal programa Spiral GSM-9000

El sistema está compuesto por un programa, un módulo *SPIRAL GSM-9000* (ver fotografía), y un cable RS-232 para conectar al PC. El diagrama de contexto de la Figura 4.13 describe el funcionamiento de esta herramienta.

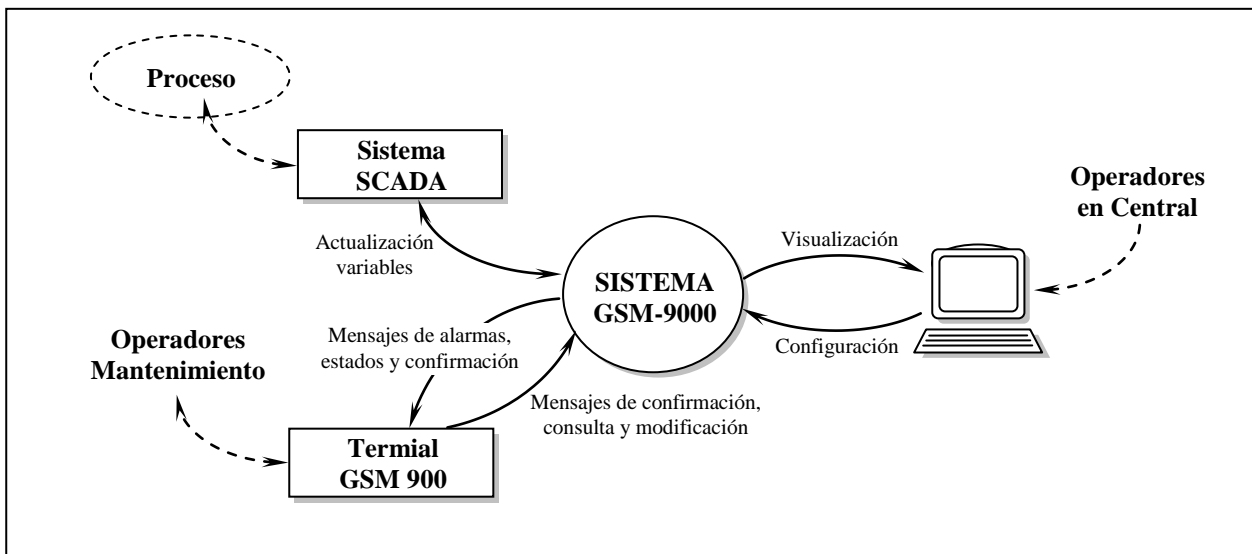


Figura 4.13: Diagrama de contexto del Sistema GSM-9000

4.3.1.3 Características funcionales

A continuación se describen las funciones concretas que se han incluido en este sistema de acceso a distancia al centro de supervisión y control de un proceso:

- Se pueden especificar dos modos de operación: manual, para enviar cualquier mensaje o aviso desde la central a un teléfono móvil, o automático, en el cual el sistema funciona de forma autónoma y notifica al operador o a los operadores las alarmas o eventos que se vayan activando.
- Configuración flexible y sencilla de las condiciones de alarma (Figura 4.14), basándose en operaciones relaciones (<, >, =) y lógicas (& - AND, | - OR, ! - NOT), y utilizando el nombre de las variables definidas previamente.

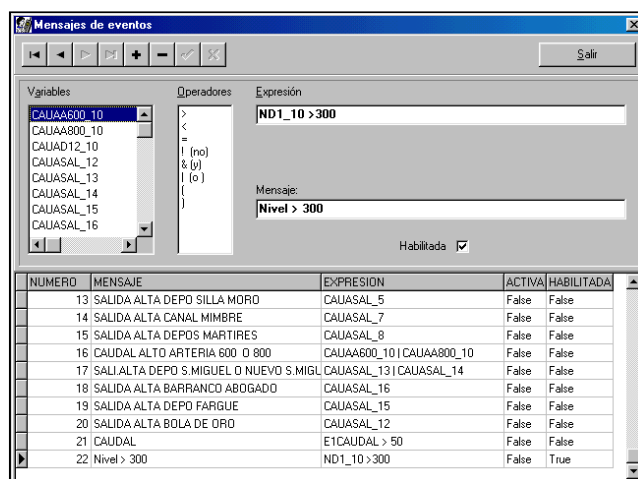


Figura 4.14: Definición de alarmas en Spiral GSM-9000

- Petición y actualización del valor de cualquiera de las variables de una aplicación de supervisión y control mediante un protocolo de intercambio de información fácil e intuitivo basado en menús.
- Control restringido en función del teléfono móvil que solicita el acceso. Se permiten, para cada uno de los teléfonos dados de alta en el programa, tres opciones no exclusivas entre sí: activo o autorizado para realizar consultas, recepción de alarmas, y permiso de escritura.
- Notificación de alarmas en función de distintos turnos horarios laborables o festivos configurables por el usuario.
- Registro automático de la evolución de los eventos del sistema tanto en un sumario de visualización rápida, como en una base de datos para búsquedas y tratamientos posteriores.

- Programa totalmente configurable y protegido con clave de acceso, que permite al operador autorizado adaptarlo a cualquier tipo de aplicación de supervisión y control. En la Figura 4.15, se muestran dos ejemplos de ventanas definidas para esta configuración.

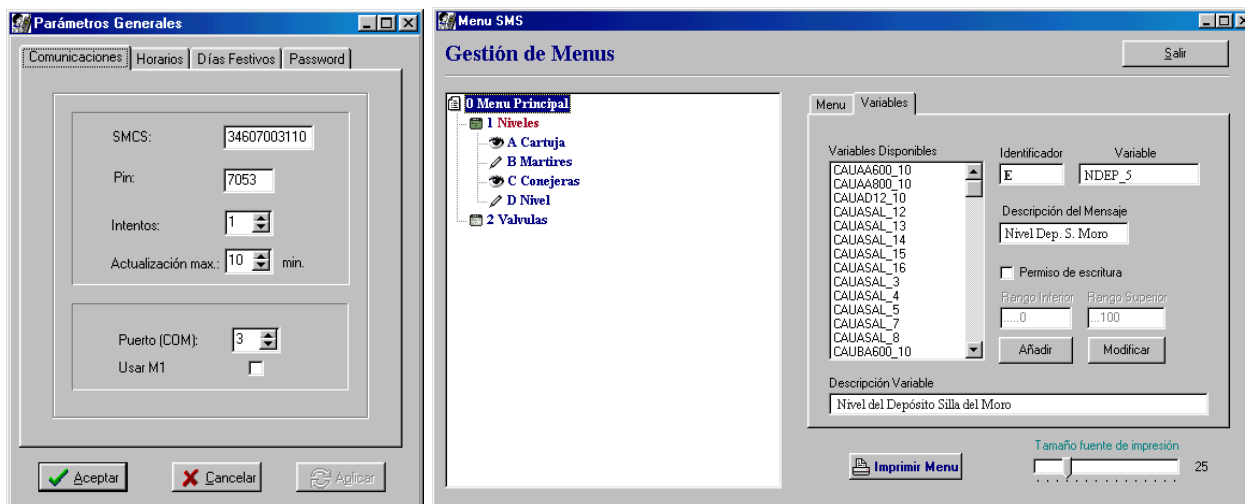


Figura 4.15: Configuración de parámetros generales y menús en Spiral GSM-9000

4.3.1.4 Detalles de programación y ejecución

El diagrama de flujo de la Figura 4.16 muestra el funcionamiento interno y los procesos más relevantes que se han desarrollado para este sistema. Actualmente está totalmente operativo y ha permitido que el operador de mantenimiento, que antes de la implantación de este sistema tenía que permanecer en la central de operaciones las 24 horas del día por si se producía alguna alarma, ahora pueda estar desplazado con un teléfono móvil y dedicarse a tareas de mantenimiento.

El programa se ha realizado para entorno Windows 9X/NT utilizando Delphi como herramienta de desarrollo. Una mejora prevista consiste en utilizar el estándar OPC, además del DDE, para aumentar la capacidad de integración con las aplicaciones existentes.

En la Figura 4.17 se observan algunos ejemplos de cómo es el protocolo de intercambio de comandos entre el teléfono móvil de un operador y la estación base en la central.

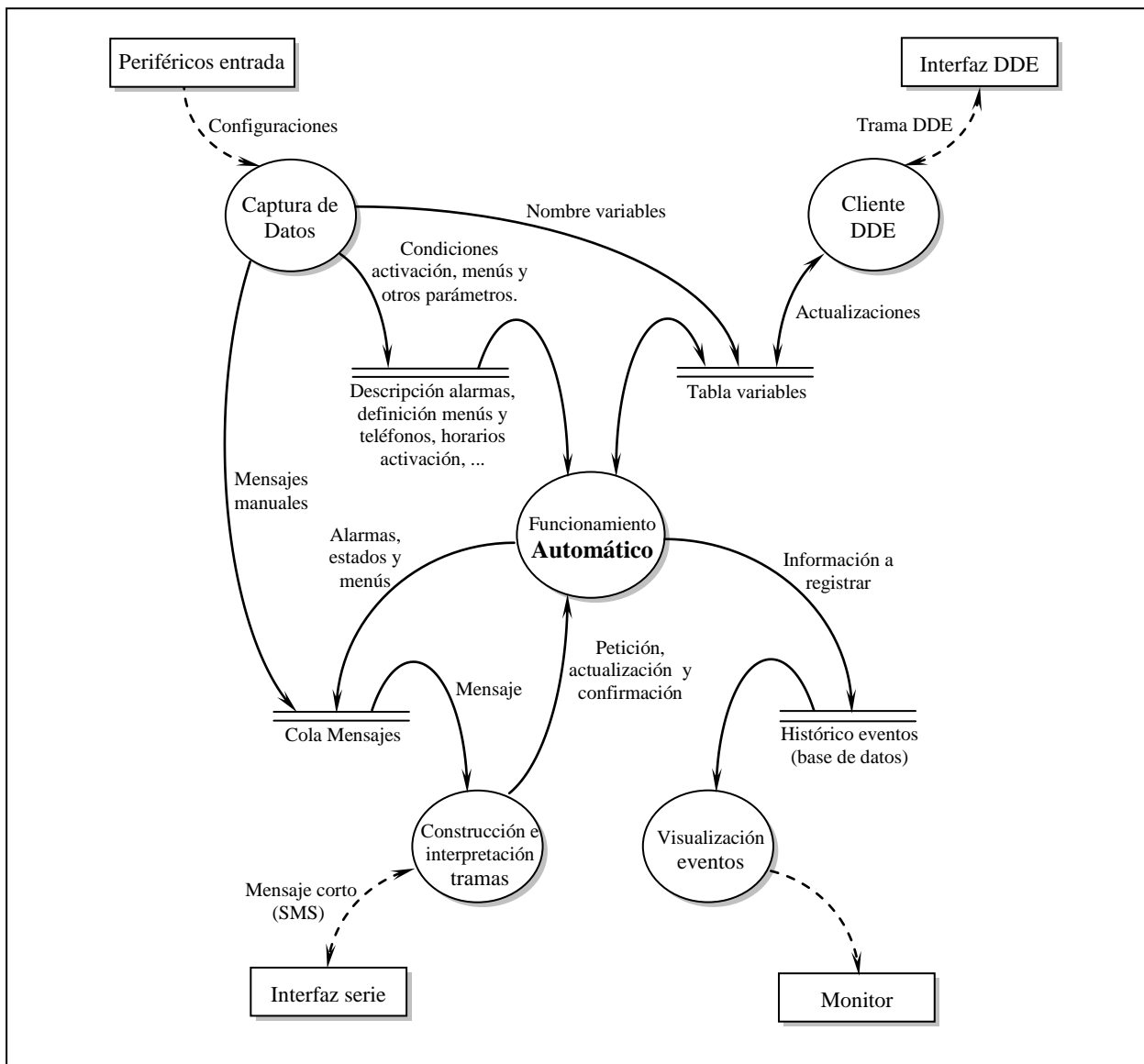


Figura 4.16: Diagrama de flujo de datos del Sistema GSM-9000



Figura 4.17: Ejemplos de funcionamiento del sistema Spiral GSM 9000

4.4 Estaciones remotas

El conjunto de estaciones remotas es muy variado, ya que el hardware, el software, y el tipo de comunicaciones, es decir, sus prestaciones, dependen del control que realizan y de su localización geográfica. De ahí que se haya optado por una clasificación en dos tipos, en función de los equipos que la integran:

1. Estaciones remotas con autómatas programables (o PLC).
2. Estaciones remotas con SP-5000.

4.4.1 Estaciones remotas con PLC

Se han elegido autómatas programables de Siemens, y más concretamente, la gama pequeña y compacta (Simatic S5-95U) para las estaciones más simples, y la gama alta y modular (Simatic S5-115U) para las estaciones que necesitan mayores prestaciones.

Cada estación remota, ver Figura 4.18, contiene un radiomodem Spiral 1200 para comunicaciones vía radio entre la central y el autómata programable encargado del control local de la estación. Puede darse el caso de que una estación remota esté formada por varias subestaciones, debido a su proximidad geográfica. En este caso las comunicaciones entre ellas se hacen por cable, utilizando o bien una red local entre autómatas (SINEC L1/L2) o bien una periferia descentraliza ET-100.

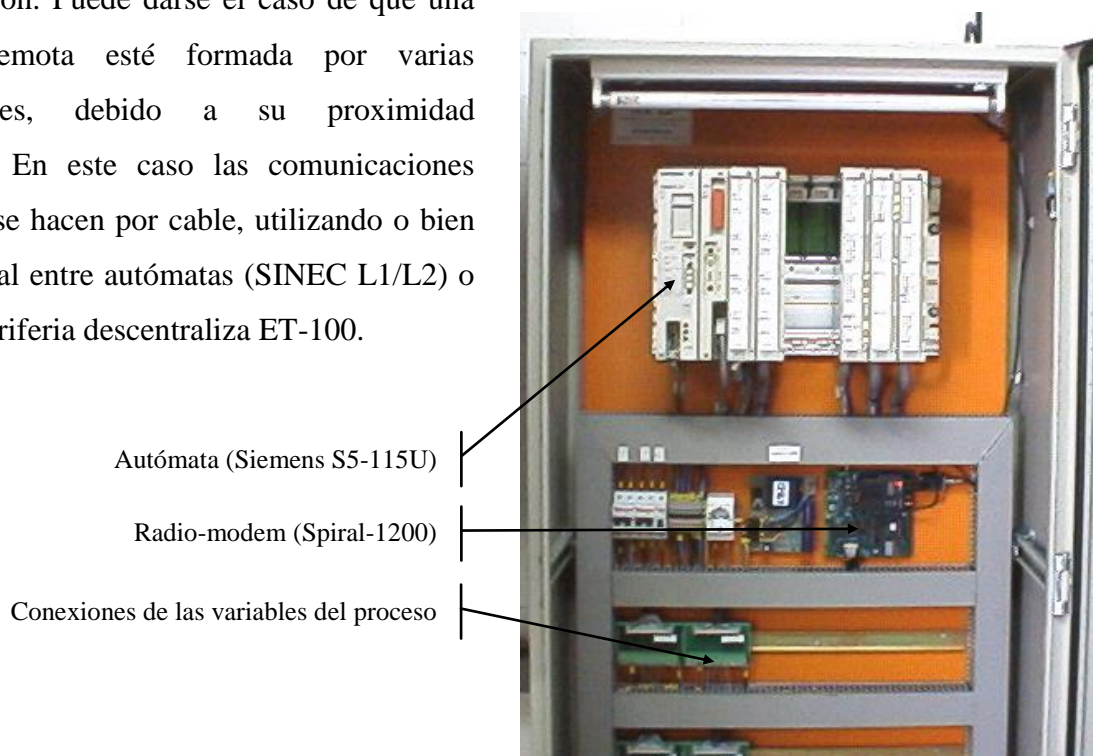


Figura 4.18: Fotografía de estación remota con PLC

Con relación al software de programación de esta gama de PLCs, se ha utilizado el lenguaje *Step 5* de Siemens, uno de los lenguajes más populares, con una programación estructurada y similar para todas las estaciones en cuanto a nomenclatura y numeración de los bloques (ver apartado de sistemas de automatización del capítulo 2 donde aparece un resumen de la programación estructurada en los autómatas Simatic S5 de Siemens). Se consigue de esta forma que las fases de desarrollo y mantenimiento posterior se realicen de manera fácil y cómoda.

En las secciones siguientes se tratan con más detalle los aspectos relacionados con el hardware y con el software que componen este tipo de estaciones, que en general se utilizan para la captación de datos y el control de los sistemas de la red de alta (estaciones con depósitos, válvulas, compuertas, grupos de bombeos, reductoras de presión,...). En el diagrama de contexto de la Figura 4.19 se pueden observar los distintos elementos que se han utilizado en las diferentes estaciones remotas de la red de abastecimiento de agua de la ciudad de Granada, con el objetivo de mostrar todas las posibles configuraciones que se han implementado en cuanto a interfaces de comunicaciones con el proceso. Las líneas discontinuas indican que la existencia o no de esos elementos depende de la estación en cuestión.

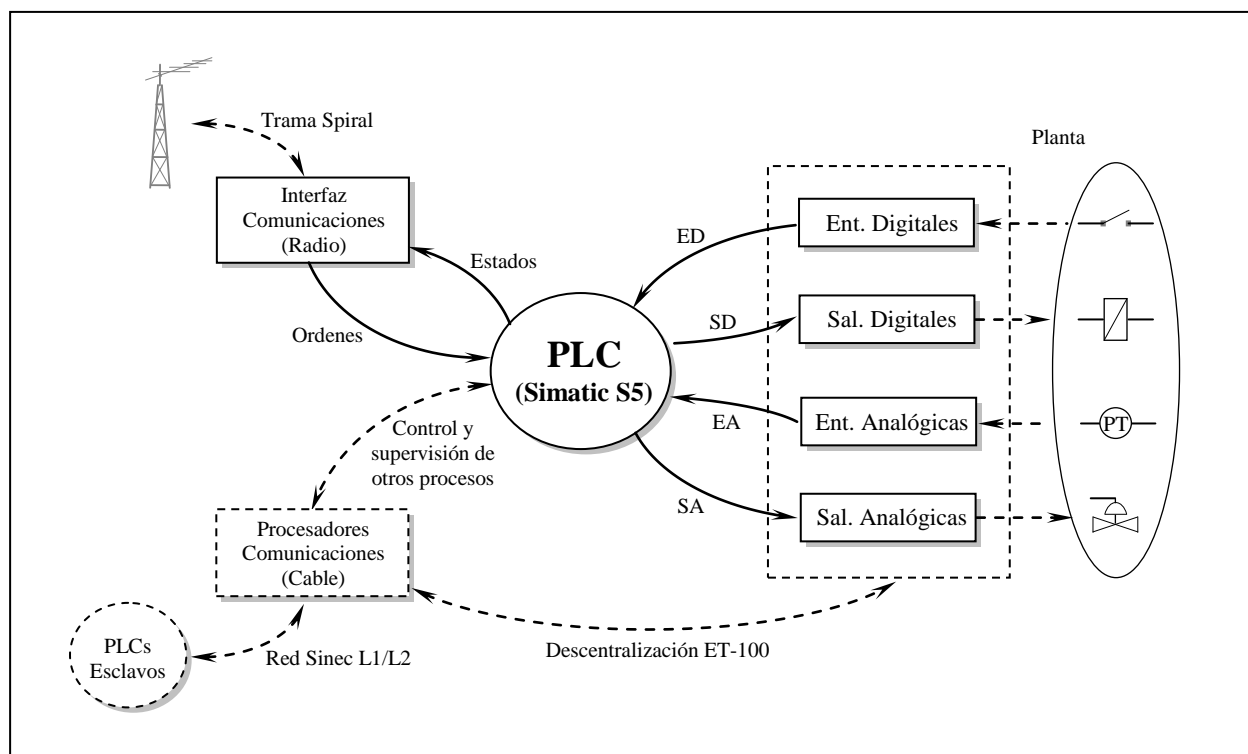


Figura 4.19: Diagrama de contexto para una estación remota con PLC

4.4.1.1 Descripción del hardware

En la Figura 4.20 se muestran los dispositivos más importantes que incluye una estación remota basada en PLC.

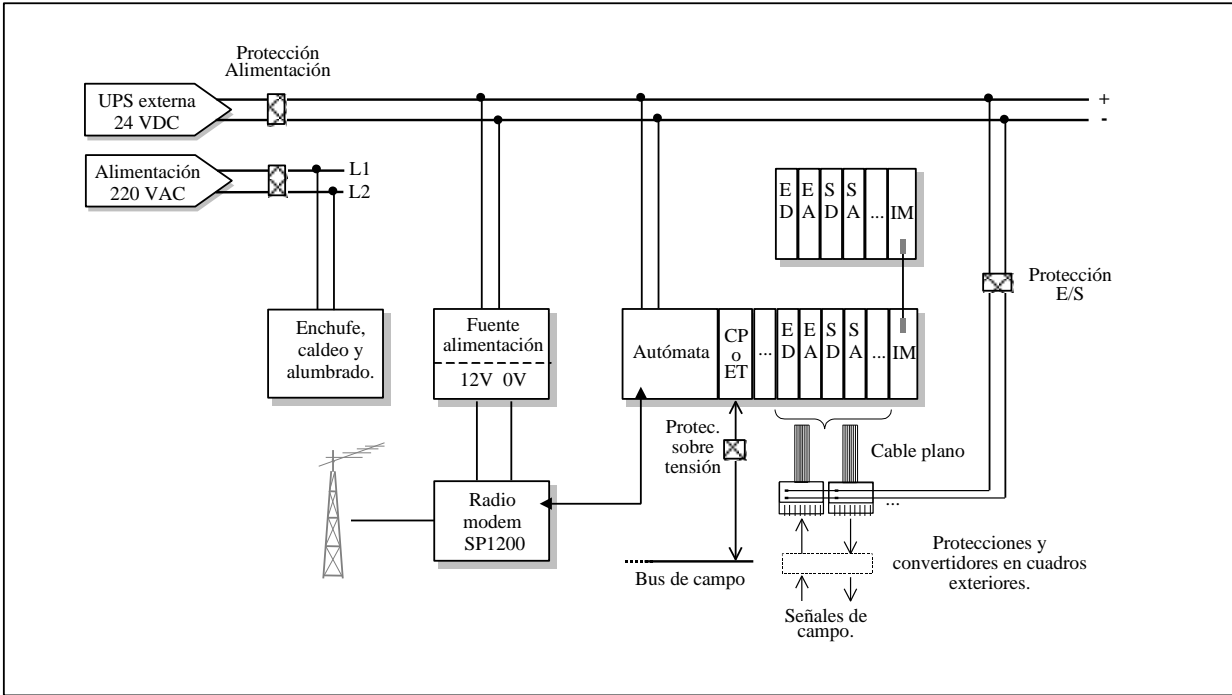
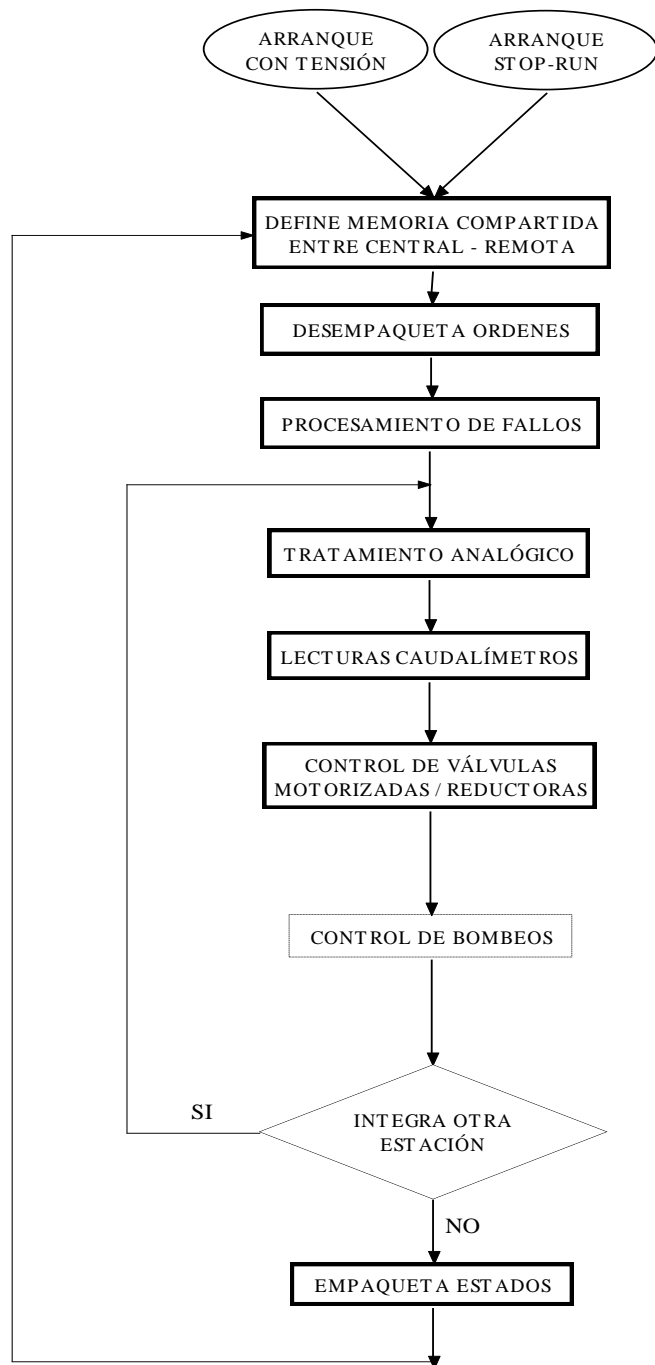


Figura 4.20: Esquema general de una estación remota con PLC

Se ha de tener presente que este sistema sólo incluye los elementos necesarios para el telecontrol, y que las protecciones, convertidores, relés, contactores, etc., necesarios para las maniobras se colocan en un sistema aparte. La gama del autómata industrial que integra la estación, la incorporación de bastidores de expansión (IM), y la necesidad de tarjetas procesadoras de comunicaciones (CP) o de descentralizaciones (ET), depende de la complejidad de dicha estación.

4.4.1.2 Estructura del software

El diagrama de flujo general que sigue cualquiera de los PLCs de este tipo de estaciones se muestra en el esquema de la Figura 4.21, donde las líneas discontinuas indican la presencia o no, según la estación en particular, de ese bloque o flujo. Junto a cada bloque se indican también las principales funciones que se realizan en cada uno de ellos.



TRATAMIENTO PRELIMINAR:
 Inicialización y configuraciones en general (variables, comunicaciones, ...).
 Crea zona de datos para comunicaciones.

El protocolo de comunicación del *driver* con los PLCs Siemens utiliza como "buffer" de comunicaciones un DB (o bloque de datos).

Las ordenes (almacenadas en palabras) se pasan a variables de tipo bits para su tratamiento.

Informa a la Central de su estado, comprueba si la Central está activa, detecta fallos en las comunicaciones (descentralizaciones, redes, ...).

Lectura y filtrados de los canales analógicos (saltos bruscos y rangos máximos/mínimos).

Lectura de pulsos (totalizador) y cálculo de litros/segundo (derivador).

Control apertura/cierre total y % apertura para válvulas sin y con lectura de posición respectivamente. Para las estaciones reductoras: inicialización de tiempos, control de la conmutación, y regulación de la presión.

Control de los minutos de marcha, de la programación y tarificación horaria, y del funcionamiento automático/manual del grupo de bombeo.

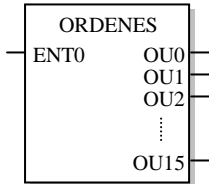
Si una estación remota integra varias estaciones (comunicadas por cable) se repiten determinados bloques para cada estación.

Los estados (variables binarias) se empaquetan en palabras de 16 bits para la comunicación por radio con la central.

Figura 4.21: Diagrama de flujo del programa de una estación remota con PLC

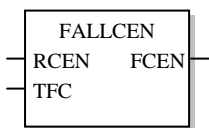
Ciertas estaciones, como la ETAP o las EDARs, no están incluidas en el diagrama general anterior puesto que son estaciones remotas mucho más complejas, y sólo se accede a ellas para obtener la información más relevante, desde el punto de vista de la central, de los procesos que integran.

Para conseguir una programación lo más estructurada posible se han construido una serie de bloques funcionales específicos, que se han utilizado en los módulos de programa de todos los PLCs de las estaciones remotas, y de los cuales, los más significativos son los siguientes⁸:



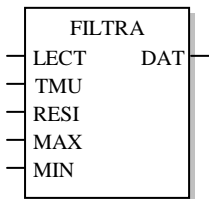
Desempaqueta órdenes:

Como entrada (ENT0) se proporciona una palabra de 16 bits, donde están empaquetadas las órdenes, y como salida (OU0, OU1, ..., OU15) se proporcionan las variables binarias internas de trabajo.



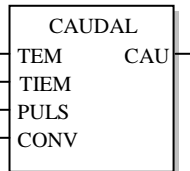
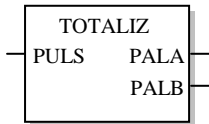
Detecta el fallo de la central:

Detecta el fallo de la central (FCEN) cuando el temporizador expira sin que antes la central lo haya cargado o reinicializado, lo cual significa que hay algún problema de comunicaciones o de avería en la central.



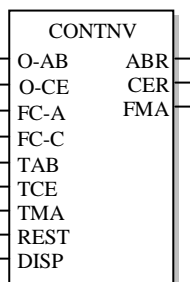
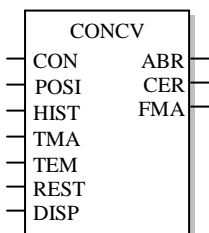
En el tratamiento analógico:

Filtra la lectura analógica (LEC) y la deja en una variable de salida (DAT) entre dos valores (MAX y MIN), eliminando además saltos bruscos o errores aleatorios (alisamiento de la señal) en aquellos casos que sea necesario.



Para la lectura de caudalímetros:

El módulo TOTALIZ es un totalizador de dos palabras de 16 bits, y el módulo CAUDAL calcula a partir de los pulsos (PULS) los litros/segundo (CAU) que pasan por ese caudalímetro.



Para el control de válvulas motorizadas:

El módulo CONCV sitúa la válvula en la posición que se le indica como consigna, dando pulsos de apertura o cierre hasta que la posición leída (POSI) coincide con la consigna (CON) con un margen de error (HIST). El módulo CONTNV abre o cierra totalmente la válvula teniendo en cuenta los finales de carrera. Ambos módulos detectan también posibles fallos de maniobra (FMA) si transcurrido un tiempo no alcanzan la posición de consigna.

⁸ La descripción de los bloques diseñados para la gestión de los grupos de bombeos y de las reductoras se pospone, puesto que en el capítulo de optimizaciones se tratarán con más detalle.

Presentamos a continuación la ejecución del programa descrito y el movimiento de la información entre las partes implicadas, mediante el correspondiente diagrama de flujo de la Figura 4.22. En dicho diagrama se puede observar también el tipo de información que manejan los distintos módulos, así como el sentido de entrada o salida de esa información tanto digital (estados y órdenes) como analógica (consignas y señales en general).

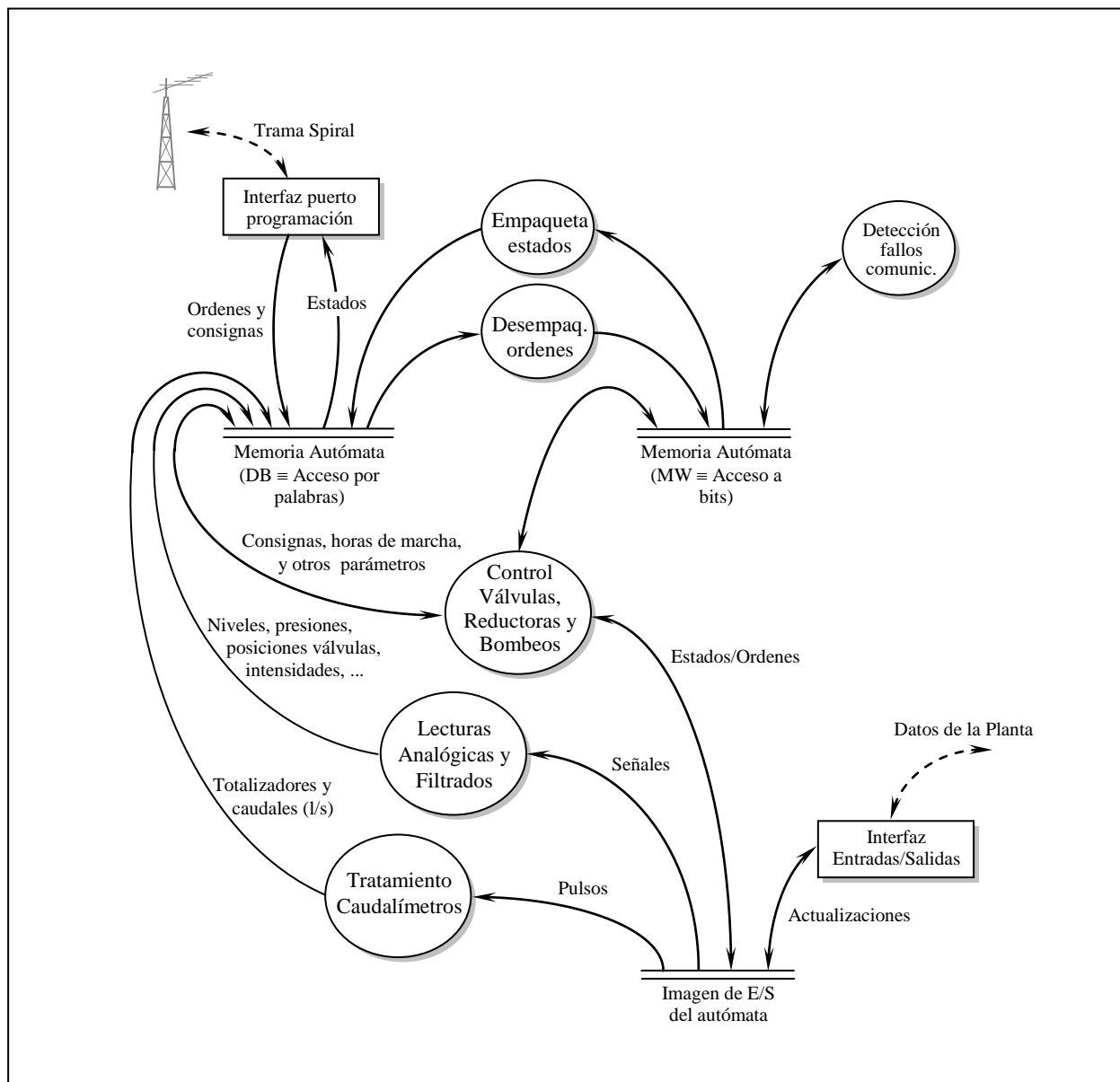


Figura 4.22: Diagrama de flujo de datos para una estación con PLC

4.4.2 Estaciones remotas con SP-5000

En la sección anterior se ha expuesto la utilización del PLC en las estaciones remotas para asumir las funciones de control, pero esta no es la única solución para la implementación de algoritmos de control en el diseño de sistemas distribuidos.

La aplicación de microprocesadores al control no es nueva; de hecho, los PLCs los incorporan dentro. La cuestión es que es posible diseñar equipos específicos basados en microprocesadores o en microcontroladores, que pueden realizar las funciones de control perfectamente. Un PLC tiene muchas posibilidades, pero rara vez se utilizan todas, ni siquiera una buena parte, por ello, los diseños específicos en ocasiones pueden resultar interesantes, porque se crean a medida, y atienden perfectamente a las necesidades.

Un caso particular es el sistema Spiral 5000 que se incluye en una serie de estaciones de la red de transporte hidráulico. La unidad central del sistema SP-5000 está basada en el microcontrolador NEC V25 de 16 bits, dispone de un número ampliable de canales de entrada/salida tanto digitales como analógicas, y de diversas vías de comunicación (radio, modem telefónico, RS-485 / RS-232, etc.), que le permiten actuar como un sistema distribuido de telecontrol y teledadida. Este sistema propuesto se describe con detalle en el Capítulo 6 de esta memoria.

Cada estación remota Figura 4.23 contiene un Radiomodem Spiral 1200 para comunicaciones vía radio con la central y una unidad central SP-5000 conectada a este. Incluye además una batería de respaldo y una fuente de alimentación para obtener la tensión necesaria para los dos equipos anteriores.

Radio-modem (Spiral-1200)

Unidad central (Spiral 5000)

Caudalímetro



Figura 4.23: Fotografía de estación remota con SP5000

Con relación al software de programación del sistema Spiral 5000, se utiliza una librería, escrita en lenguaje C, diseñada a medida para facilitar la programación de aplicaciones con este sistema (los detalles sobre este software se encuentran en el Capítulo 6). Además, como las estaciones que incorporan este sistema son similares, el programa es el mismo para todas. Se consigue así una estructura del software fácil de desarrollar y mantener para la explotación posterior del sistema.

En las secciones siguientes se tratan los aspectos relacionados con el hardware y con el software al nivel de usuario que se ha diseñado para este sistema, y que actualmente se utiliza en la mayoría de las estaciones de la sectorización de la ciudad de Granada. En el diagrama de contexto de la Figura 4.24 se puede observar que la estructura general es muy similar a la mostrada anteriormente para los PLC. Las líneas discontinuas indican que la existencia o no de esos elementos depende de la estación en cuestión.

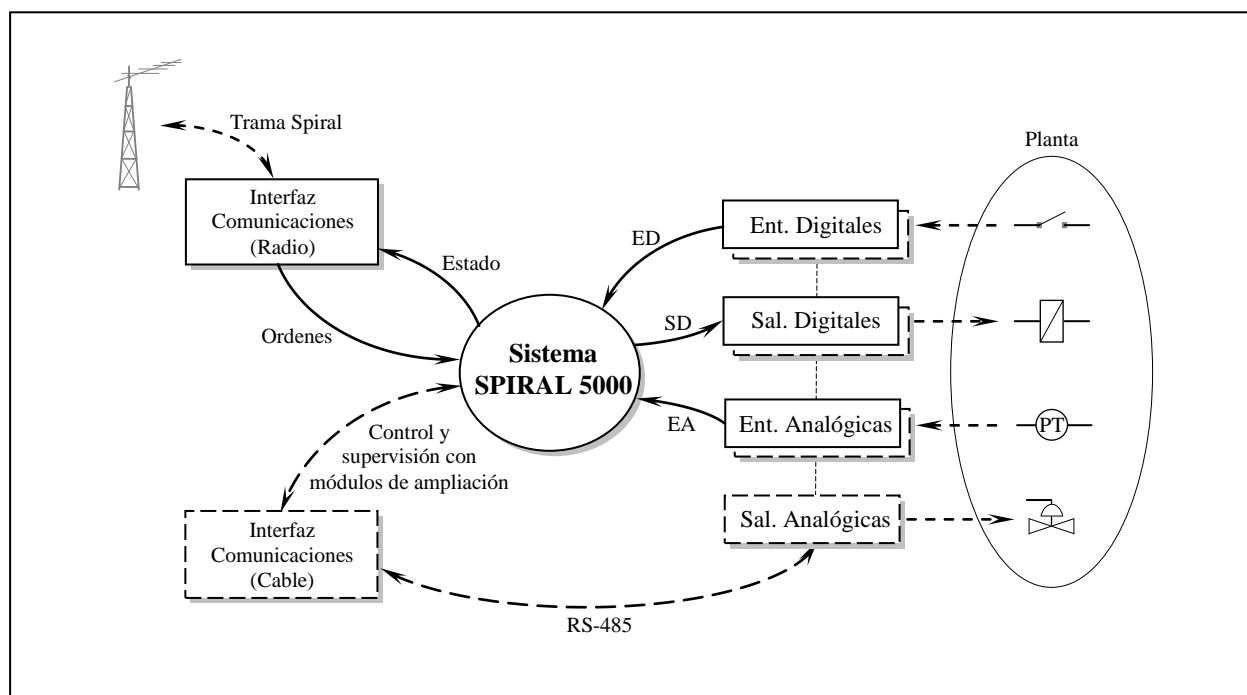


Figura 4.24: Diagrama de contexto para una estación con SPIRAL 5000

4.4.2.1 Descripción del hardware

En la Figura 4.25 se muestran los dispositivos más importantes que incluye una estación remota basada en SP-5000.

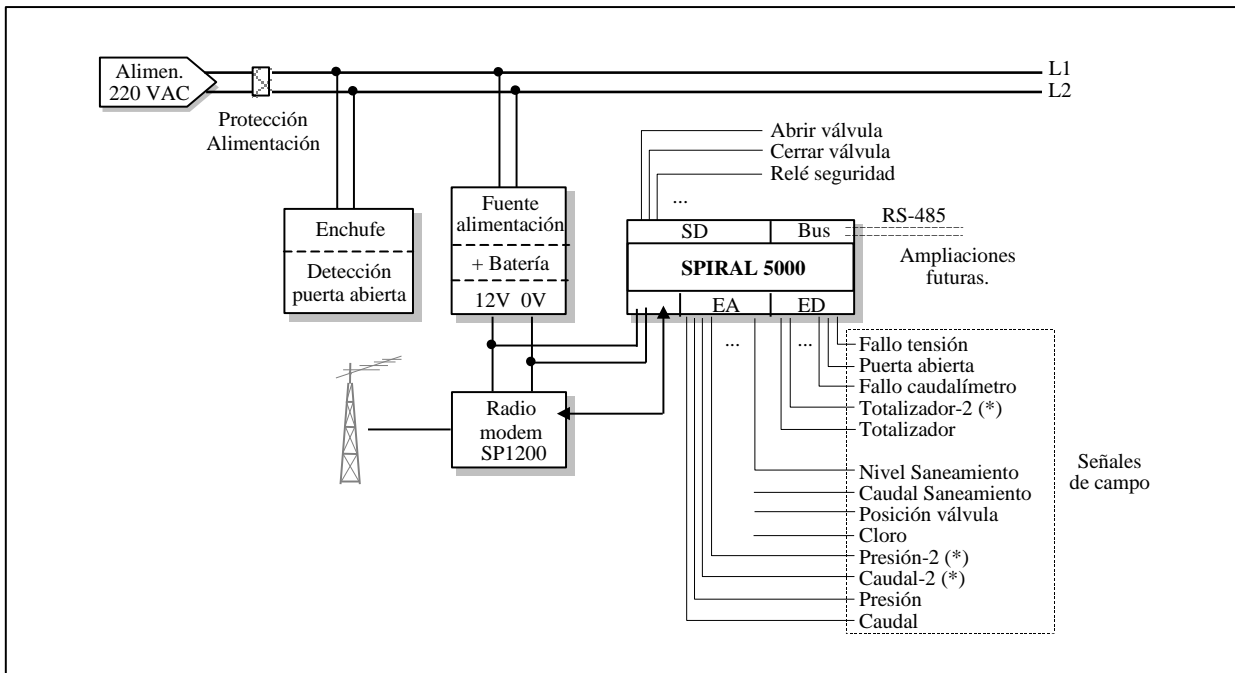


Figura 4.25: Esquema general de una estación remota con SPIRAL 5000

Se ha de tener presente que este sistema sólo incluye los elementos necesarios para el telecontrol de un sector, pero en realidad está preparado para que se le pueda conectar las señales de otro sector completo y algunas señales adicionales (cloro, turbidez,...). De hecho, algunas estaciones para la sectorización de la ciudad se encargan de dos sectores, aprovechando la proximidad geográfica de las válvulas que controlan el aporte de agua a dichos sectores (señales con * en la Figura 4.25).

4.4.2.2 Estructura del software

Para desarrollar el programa que se encarga básicamente de la captura de datos (analógicos y digitales) se ha utilizado la librería “*SPC++*”. Esta librería, escrita en lenguaje C++, define una serie de clases y métodos que facilitan la programación de aplicaciones utilizando este sistema basado en microcontrolador (los detalles sobre esta librería se pueden encontrar en el Capítulo 6). Se resuelve así uno de los problemas más graves a la hora de utilizar un sistema programado basado en microcontrolador: el desarrollo fácil y homogéneo en cuanto a la programación de la aplicación concreta.

A continuación se muestra la estructura del programa en C codificado para el Sistema SP5000 que incorporan la mayor parte de los sectores de la red de distribución de agua Granada.

```

/*-----
   main.cpp      Proyecto:   Sectorización de Emasagra, Fecha: Mayo 1998, Versión: 1.0
   -----*/

#include <stdio.h> , <stdlib.h> , <dos.h> , "serie.h" , "tramas.h" , "sp5000.h"

// Definiciones particulares
#define SD      1      // Salidas digitales de la unidad central
#define ED      2      // Entradas digitales de la unidad central
#define EA      4      // Entradas analógicas de la unidad central

#define DB0     sp.db[0] // Zona de acceso comunicaciones radio
#define WED     0      // Entradas digitales
#define WEDL1_0 1      // Detección flanco bajada
#define WEDL0_1 2      // Detección flanco subida
#define DWPTOT1 3      // Doble palabra para totalizador parcial 1
#define DWPTOT2 5      // Doble palabra para totalizador parcial 2
#define WEA0    7      // Entrada analógica 0
...
#define WEA10   29     // Entrada analógica 10
#define WCONTROL 30    // Palabra de control
#define WSD     31     // Salidas digitales
#define MIN     (unsigned int)sp.db[0][32] // Direcciones hora y fecha.
#define HOUR    (unsigned int)sp.db[0][33]
...
#define CHANGETIME    sp.db[0][WCONTROL].bit0 == 1 // Macros
#define TIMECHANGED  sp.db[0][WCONTROL].bit0 = 0
#define RESETEATOT1  sp.db[0][WCONTROL].bit1 == 1
...
#define DIGITAL_INPUTS      ( sp.Tarjeta(ED)->GetWord() )
#define ANALOG_INPUT(a)    ( sp.Tarjeta(EA)->GetWord( a ) )
#define SET_DIGITAL_OUTPUT(a) ( sp.Tarjeta(SD)->SetWord( a ) )

// Variables globales
SP5000 sp; // Definición sp como de tipo SP5000 (clase principal)
unsigned long tot1, tot2, tot3, tot4;

// Programación de las funciones

void Latches() // Calcula los flancos de subida y de bajada
{
    unsigned int change, aux, oldvalue, i;
    oldvalue = DB0[WED];
    DB0[WED] = DIGITAL_INPUTS;
    if ((change = oldvalue ^ DB0[WED]) != 0) {
        for (i=0, aux=1; i<16; i++, aux<<=1) {
            if (change & aux) {
                if (oldvalue & aux)
                    DB0[WEDL1_0] = (unsigned int)(DB0[WEDL1_0]) | aux; // Flancos bajada
                else
                    DB0[WEDL0_1] = (unsigned int)(DB0[WEDL0_1]) | aux; // Flancos subida
            }
        }
    }
}

void Totalizadores() // Tratamiento de totalizadores continuos y parciales
{
    unsigned int i, mask;
    unsigned long contador;
    static Reloj rlm(6000);
    if (DB0[WEDL0_1].bit0) {
        contador = DB0[DWPTOT1] | ((unsigned long)DB0[DWPTOT1+1] << 16);
        contador++;
        tot1++;
        DB0[DWPTOT1] = contador & 0x0000FFFF; // Inc. totalizador continuo
        DB0[DWPTOT1+1] = (contador & 0xFFFF0000)>>16;
        DB0[WEDL0_1].bit0 = 0;
    }
    // ... (igual para bit1, bit2 y bit3)
    rlm.Iniciar();
    if (rlm.Fin()) {
        int mes, dia, horas, minutos;
    }
}

```

```

    TIM t;
    rlm.Parar();
    rtc_rd(&t);
    horas = t.hour10*10+t.hour1;
    minutos = t.min10*10+t.min1;
    if (horas == 0 && minutos == 0) {
        DB0[DWPTOT1] = tot1 & 0x0000FFFF; // Actualiza totalizador parcial
        DB0[DWPTOT1+1] = (tot1 & 0xFFFF0000)>>16;
        // ... (igual para DWPTOT2, DWPTOT3 y DWPTOT4)
        tot1=tot2=tot3=tot4=0L;
    }
}
if (RESETEATOT1) { // Reset totalizadores
    DB0[DWTOT1] = DB0[DWTOT1+1] = 0;
    RESETEATOT1;
}
// ... (igual para RESETEATOT2, RESETEATOT3 y RESETEATOT4)
}

void CambioHora() // Actualiza el reloj interno con la hora almacenada
{
    ...
}

void AsignaEntradasAnalogicas() // Lectura y acotación de variables analógicas
{
    unsigned int uvalue;
    uvalue = ANALOG_INPUT(0); // Lectura canal analógico 0
    if (uvalue>MAXIMO) uvalue = MAXIMO; // Limita valor entre Max. Y Min.
    if (uvalue<MINIMO) uvalue = MINIMO;
    DB0[WEA0] = uvalue;
    // ... (igual para ANALOG_INPUT(1), ANALOG_INPUT(2), ..., y ANALOG_INPUT(10))
}

void Logica(void) // Programa para los sectores
{
    Latches();
    Totalizadores();
    CambioHora();
    AsignaEntradasAnalogicas();
    SET_DIGITAL_OUTPUT(DB0[WSD]);
}

void Inicializacion(void) // Inicializaciones varias.
{
    ve_init(); // Las tres sentencias que hay a continuación son obligatorias
    led(0); // Apaga el led de buen funcionamiento
    sp.Inicializacion(1, 1); // n° estación SPIRAL 1, hace uso de Watch-Dog
    IniciaPuerto(COM1, V9600, 0, 8, 1); // Inicializa comunicaciones con la Radio
    tot1=tot2=tot3=tot4=0L;
}

void main(void) // Programa principal (el mismo para cualquier aplicación)
{
    Inicializacion();
    sp.Run(Logica);
}

```

Como se puede apreciar los métodos de la clase y la estructura que se ha de seguir en la programación permiten que el desarrollador de un automatismo sólo se tenga que encargar de la parte de mando, y no de los detalles propios del hardware diseñado⁹.

⁹ En el capítulo dedicado al sistema Spiral 5000 se proporcionan más detalles en relación a la clase y métodos utilizados en los programas desarrollados para este sistema.

4.5 Comunicaciones

4.5.1 Sistemas de telecontrol vía radio

La intercomunicación de PLCs y dispositivos de campo en general, a través de enlaces punto a punto, buses de campo o redes locales, no siempre es posible o económicamente viable. La utilización de comunicaciones digitales por radio es una alternativa adecuada y simple en aquellas aplicaciones en que es difícil o costosa la instalación de cables, debido a las condiciones de entorno (carreteras, canales, población, etc.), y su utilidad se hace más patente cuando uno o varios nodos de comunicación se encuentran en movimiento.

Los criterios que determinan la elección de una comunicación vía radio frente a otras posibilidades se pueden resumir en los siguientes:

- Distancias del orden de decenas o cientos de Kms. La fiabilidad de las comunicaciones digitales vía radio es mayor que la de las líneas telefónicas. Además, el coste de líneas telefónicas dedicadas puede ser muy superior.
- Puntos lejanos o de difícil acceso para el tendido de cables. En determinados entornos las condiciones pueden ser tales que hagan inviable el tendido de redes cableadas.
- Cuando interesa libertad de movimientos entre el computador de supervisión y el sistema a controlar (vehículos o maquinaria móvil).

Por tanto, los sistemas de telecontrol vía radio se pueden aplicar, según estos criterios, a cualquier proceso de medida y control a distancia y, en general, a cualquier sistema de control distribuido con elementos dispersos, como es el caso de las redes de abastecimiento y saneamiento de agua.

4.5.2 El sistema SPIRAL

Como ya se ha mencionado, el objetivo principal que desde un principio se pretende afrontar consiste en la integración de los sistemas necesarios para la supervisión y control, vía radio, de la red de abastecimiento de la ciudad de Granada.

Para ello hemos adoptado y mejorado el sistema SPIRAL [Gom-91, Gom-94], con el propósito de utilizar su idea de arquitectura de red que permite el telecontrol vía radio de diferentes estaciones desde una estación central.

Como equipo de comunicaciones vía radio se utiliza el radiomodem SP-1200 [Gom-94], que actúa como nodo de red de radioenlaces, con posibilidad de ser adoptado como:

- Equipo terminal de comunicaciones (DCE)
- Equipo terminal de datos (DTE)
- Equipo de conmutación de datos (DSE) o nodo encaminador de red (*Router*)
- Nodo puente (*Bridge*)
- Nodo convertidor de protocolo o pasarela (*Gateway*)

Como equipos de supervisión y control se utilizan, o bien un PLC, o bien SP-5000. En ambos casos se dispone de entradas y salidas tanto digitales como analógicas, e interfaces de comunicación serie que permiten configurar una red local con los protocolos de comunicación de SPIRAL.

Para permitir al operador interactuar con el proceso, se ha utilizado una herramienta SCADA, configurada de forma adecuada, y se ha realizado el gestor de comunicaciones apropiado para manejar todo este sistema SPIRAL (en el Capítulo 5 se proporcionan los detalles correspondientes a este software de comunicaciones).

4.5.3 Modelo de capas del sistema de comunicaciones utilizado

La clave para simplificar la complejidad de diseño de redes de computadores consiste en la proyección de la red como un conjunto jerárquico de capas (Arquitectura de Red), cada una superpuesta a la anterior, reduciendo el diseño global al diseño de cada una de las capas y facilitando considerablemente el trabajo de desarrollo y mantenimiento, así como la interconexión de equipos heterogéneos.

La red de radiomodems mostrada en la Figura 4.3 utiliza el sistema SPIRAL descrito en el apartado anterior, y en ella se tienen en cuenta los principios arquitectónicos sobre los que se basa el modelo de capas RM-OSI de ISO [Hal-98], atendiendo además a las siguientes consideraciones:

- La arquitectura tratada es asimétrica, en el sentido de que se intercomunica un proceso real con una imagen virtual de este, representada sobre una computadora en la sala de control central.

- Se lleva a cabo la supervisión y control de diversas plantas o áreas de fabricación distribuidas geográficamente en distintos puntos. Se conectan, por tanto, decenas de estaciones a través de la red, con una topología en estrella, y con la posibilidad de que algunas estaciones actúen como equipos de conmutación de datos para enlazar estaciones lejanas o sin visibilidad directa con la central (estrella ramificada).
- La red es capaz de interconectar equipos heterogéneos: el protocolo de comunicación de los sistemas de control de fabricación puede no ser siempre el mismo; de hecho, se mantienen distintos protocolos de comunicación para las distintas plantas de fabricación.
- La conexión se realiza de forma continuada e ininterrumpida, por lo que carece de sentido el establecimiento de “*sesiones*” de comunicación.
- La información que se transfiere corresponde a la actualización cíclica de la imagen virtual del proceso. Las órdenes de forzado sobre el proceso o las alarmas críticas generadas por este rompen la secuencia de actualización cíclica.
- La red se destina al control y supervisión del proceso, aunque el control desde la estación central no cerrará, a través de la red de radiomodems, los lazos de control correspondientes a procesos críticos, que se resolverán con los controladores de planta (control distribuido).

La arquitectura se ha resuelto basándose en las cinco capas representadas en la Figura 4.26. Las tres capas inferiores (medios físicos, red y control) corresponden a la red de comunicaciones de radiomodems, y el resto de las capas (transporte y aplicación) al tratamiento de la información del proceso.

Los detalles de diseño de las tres capas inferiores quedan fuera de este trabajo y se pueden encontrar en [Gom-94]. Hay que destacar que una de las ventajas más importantes de usar la red de radiomodems SPIRAL, resuelta en estas tres capas inferiores, es la posibilidad de utilizar tanto un servicio de conexión transparente como cualquier protocolo de control (PLC) embebido sobre tramas del protocolo de red SPIRAL. Característica esta que se utiliza en nuestro caso, ya que muchas estaciones remotas están equipadas con PLC (conectados con los radiomodems a través del puerto de programación) cuyo protocolo de control no es adecuado para ser soportado por las conexiones de red.

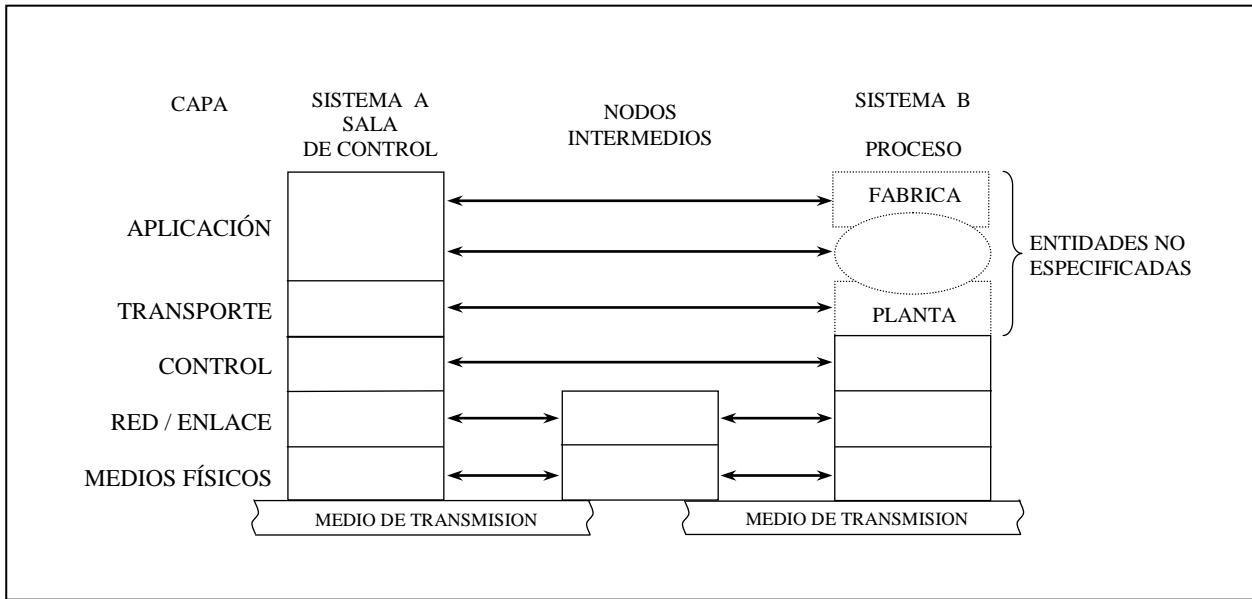


Figura 4.26: Estructura de capas del sistema SPIRAL

En este modelo de arquitectura sólo la estación central y los sistemas de control local de las estaciones remotas están involucrados en las capas de transporte y aplicación. En el Capítulo 7 se hace una descripción detallada de los programas de control y supervisión utilizados y configurados para gestionar los recursos de la red SPIRAL desde la estación central.

La capa de transporte se encarga de las tareas de multiplexación y sondeo sobre las distintas estaciones remotas, así como de la segmentación en bloques de variables consecutivas para optimizar el tamaño de los paquetes de datos sobre la red. En esta capa se resuelve también la detección de pérdidas de enlace, y además se ofrecen servicios de actualización y forzado de variables. Todas estas funciones corren a cargo del gestor de comunicaciones que describiremos con detalle en el Capítulo 5.

4.6 Conclusiones

El objetivo principal de este capítulo puede resumirse de la siguiente forma: *mostrar el trabajo realizado en relación con los equipos de adquisición, control y comunicaciones instalados en la ciudad de Granada para la gestión del agua.*

Para ello, se ha hecho un recorrido en el tiempo para mostrar cómo se han ido implantando los sistemas de adquisición y control para la explotación del agua en la Ciudad de Granada.

Se ha mostrado la arquitectura de la red de comunicaciones concebida para este sistema de control, tanto globalmente como para cada subproceso, utilizando esquemas detallados y tablas de los sistemas de automatización desarrollados.

A continuación, se han presentado los diseños realizados tanto en la central como en las estaciones remotas, destacando las siguientes aportaciones¹⁰:

- Un sistema para poder manipular a distancia, vía telefonía móvil GSM-SMS, una aplicación de control y supervisión, con el objetivo final de aprovechar los recursos humanos de forma eficiente, para así ahorrar costes y aumentar la operatividad del sistema.
- Una metodología de programación homogénea y sistemática para los PLC instalados en las estaciones remotas equipadas con este tipo de controladores industriales, con el objetivo de que las fases de desarrollo y mantenimiento posterior se realicen de manera fácil y cómoda.
- La utilización de un sistema de control distribuido, diseñado a medida y con prestaciones similares a un PLC, con el objetivo de equilibrar el sistema de automatización con las necesidades reales de ciertas estaciones, para las cuales un PLC no es el sistema más adecuado a utilizar según la relación coste/prestaciones.

Por último, se ha descrito también el sistema de comunicaciones utilizado, y en concreto se ha justificado el uso tanto de las comunicaciones por radio como del sistema SPIRAL.

¹⁰ El resto de las aportaciones, aunque se hayan mencionado a lo largo de este capítulo, se describen en capítulos independientes para tratarlas con más detalles.

Capítulo 5:

Gestor de comunicaciones distribuidas

El diseño de protocolos de red y de comunicaciones de datos unificados, así como de dispositivos con conexiones e interfaces estandarizados, permite la integración vertical, vía radio y telefonía, desde las salas de control a los procesos con elementos dispersos.

En el capítulo anterior ya se ha justificado el por qué de utilizar comunicaciones vía radio, y se ha introducido el sistema SPIRAL como arquitectura de red optimizada para las transmisiones de datos por este medio. En este capítulo se describe el diseño de una interfaz de comunicaciones (*driver*) entre aplicaciones en el entorno Windows y equipos de automatización en general, y en particular para las SP-5000 y los PLCs utilizados en la red de abastecimiento de agua de la Ciudad de Granada. Dicho software se está utilizando actualmente en otras aplicaciones de control distribuido (ver apéndice A).

En primer lugar, sección 5.1, se hace una descripción general de este sistema y a continuación, sección 5.2, se muestra su arquitectura interna y las características funcionales más importantes. En la sección 5.3 se desglosa la arquitectura anterior detallando los módulos más importantes. Se continúa, sección 5.4, indicando a modo de ejemplo cómo se ha utilizado este *driver* en una aplicación real, y por último en la sección 5.5 se exponen una serie de medidas de funcionamiento, tanto simuladas como reales, para mostrar algunas de las ventajas que se pueden conseguir al utilizar este gestor de comunicaciones.

5.1 Descripción general

El desarrollo de este software de apoyo a la integración de sistemas (en adelante ICeR, Interfaz de Comunicaciones con estaciones Remotas), se ha realizado en dos fases:

- Una primera versión que cubría las especificaciones necesarias y suficientes para un funcionamiento acorde con las exigencias de la arquitectura SPIRAL propuesta¹. Esta versión ha estado funcionando durante varios años y nos vamos a referir a ella para obtener datos de funcionamiento (tiempo de ciclo y tasa de fallos de comunicaciones global y por estaciones, principalmente) que nos sirvan para mostrar los progresos conseguidos con los nuevos desarrollos.
- Una segunda versión, mejorada respecto a la anterior en una serie de puntos relacionados básicamente con la optimización de las tramas a transmitir, mayores posibilidades de configuración, más información (estadísticas) del estado de las comunicaciones, y adaptación al estándar industrial OPC para la comunicación de sistemas y dispositivos. Actualmente es la que se encuentra operativa y por tanto la que se va a describir a lo largo del capítulo.

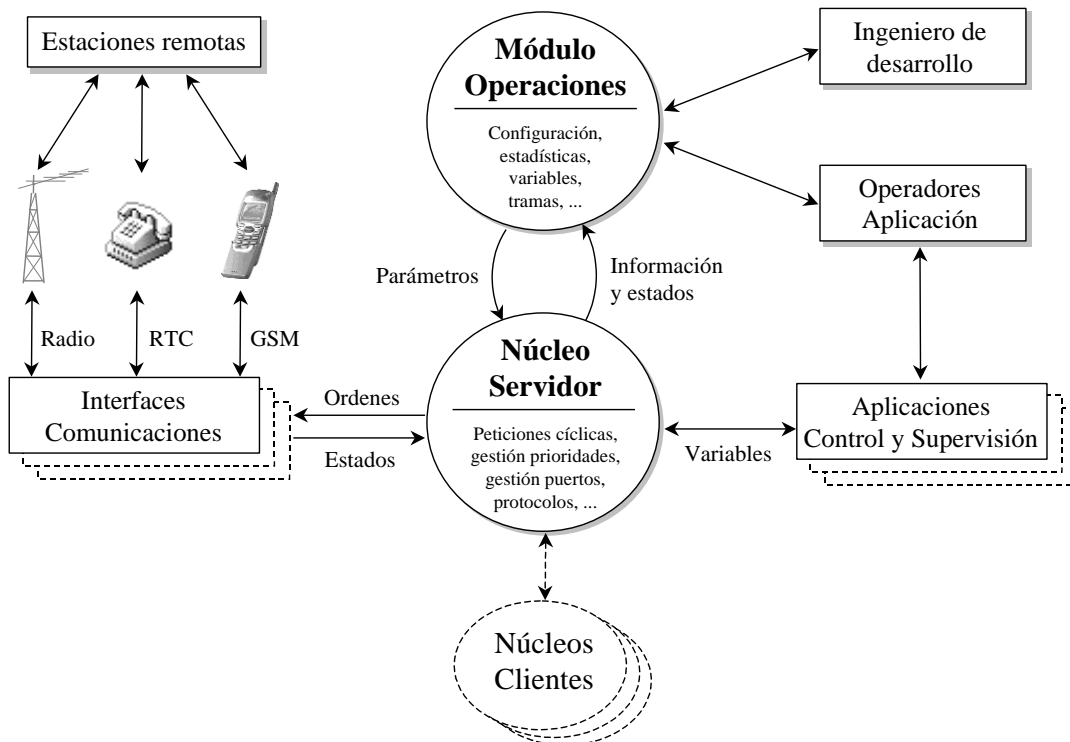


Figura 5.1: Diagrama de contexto del driver de comunicaciones ICeR

¹ Introducidas en el capítulo anterior, sección 4.5.3.

En la Figura 5.1 se muestra el diagrama de contexto de este software, cuyas características funcionales más importantes son:

1. Modularidad, es decir, existe un programa de operaciones (configuraciones, obtención de estadísticas y variables, visualización de tramas), y un núcleo principal para la gestión de las comunicaciones, totalmente independiente del anterior y conectados por TCP/IP.
2. Soporta las comunicaciones vía radio (RTC² y GSM en desarrollo) con estaciones remotas equipadas con PLC (Siemens y Omron actualmente) y con SP-5000, utilizando el radiomodem SP-1200³.
3. Incorpora el nuevo estándar OPC, además del clásico por DDE, para el enlace con las aplicaciones de control y supervisión de procesos (SCADA), y en general con cualquiera que soporte los estándares anteriores (hojas de cálculo, bases de datos, programas desarrollados a medida, etc.).
4. Gestiona eficientemente el protocolo Spiral⁴, en cuanto a las tareas de multiplexación y sondeo sobre las distintas estaciones remotas, así como de la segmentación en bloques de variables consecutivas, con el objetivo de optimizar el tamaño de las tramas y por consiguiente minimizar la probabilidad de que se produzca un fallo de comunicaciones.
5. Gestiona prioridades, maneja varios puertos y protocolos estándares, e incorpora un sondeo dinámico y más flexible sobre las estaciones.
6. Se puede utilizar como servidor, como cliente o en configuración redundante.

Se ha usado como herramienta de programación un lenguaje visual de última generación (C++ Builder de Borland), con el que se ha programado también un simulador de estaciones remotas para hacer las pruebas y obtener resultados de funcionamiento.

5.2 Arquitectura del sistema

Las características que acabamos de describir se aclaran de forma gráfica en esta sección y se amplían por separado en la siguiente. En primer lugar, la Figura 5.2 muestra un esquema de la arquitectura interna del software diseñado para este sistema de comunicaciones, en el que se aprecian los distintos módulos y los protocolos de enlace empleados entre ellos.

² RTC = Red Telefónica Conmutada

³ SP-1200 es un equipo de comunicaciones vía radio descrito en [Gom-94].

⁴ El protocolo espiral, también introducido en [Gom-94], se incluye en la sección siguiente.

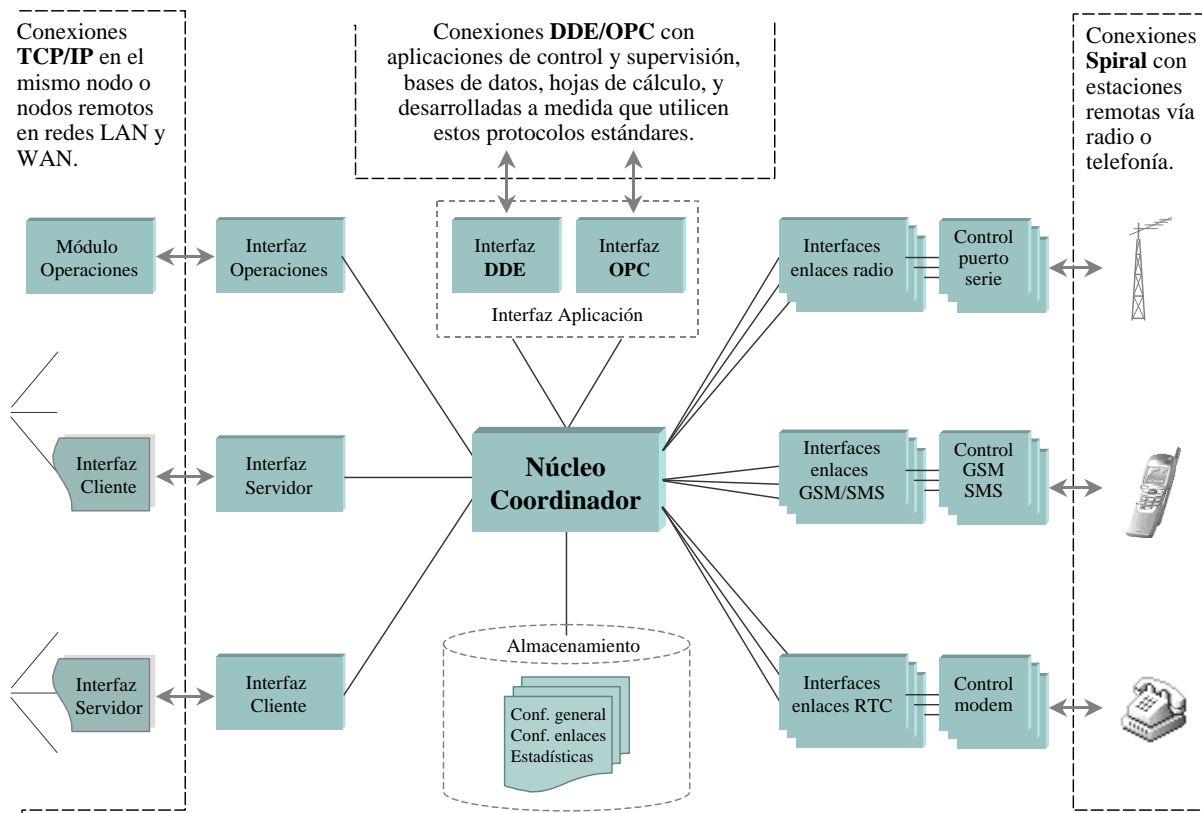


Figura 5.2: Arquitectura del driver de comunicaciones ICeR

ICeR está programado utilizando POO⁵, semáforos⁶ y hebras⁷ (*Thread*) para conseguir mayor estructuración y un procesamiento concurrente.

Los módulos para el control de puertos serie, mensajes cortos con GSM y módem telefónicos son por tanto hebras encargadas de realizar funciones como: configurar los puertos de comunicaciones (velocidad, paridad, bits de stop, bits de datos, etc.), detectar tramas, analizar tramas SMS, y gestionar las llamadas y los comandos AT para el control del módem, entre otras cosas. Los detalles del resto de los módulos merecen más atención y se describen en la siguiente sección.

⁵ POO = Programación Orientada a Objetos.

⁶ Semáforos = funciones de bloqueo que impiden la ejecución de otras hebras, y que se utilizan para establecer secciones críticas o zonas de programa que permiten solamente el paso de una hebra cada vez.

⁷ Una aplicación multihebra es una aplicación que contiene varias vías simultáneas de ejecución que comparten el mismo espacio de memoria.

Las distintas configuraciones que se pueden conseguir en cuanto a la utilización tanto del *driver* como del módulo de operaciones son las siguientes (ver Figura 5.3):

- ICeR CM servidor tanto en bus como en estrella (CM = Control y Monitorización).
- ICeR CM cliente/servidor, para conexiones en bus.
- ICeR CM cliente, para conexiones en estrella o extremos de conexiones en bus.
- ICeR SM cliente, para sistemas en escucha o nodos en los que no se generan tramas de escritura (SM = Solo Monitorización).
- ICeR SR para supervisar al servidor principal, y mostrar un mensaje y realizar la conmutación automática en caso de avería del principal (SR = Servidor Redundante).
- ICeR MO para configurar y monitorizar el *driver*, en el mismo computador o remotamente por red o vía módem telefónico (MO = Modulo Operaciones).

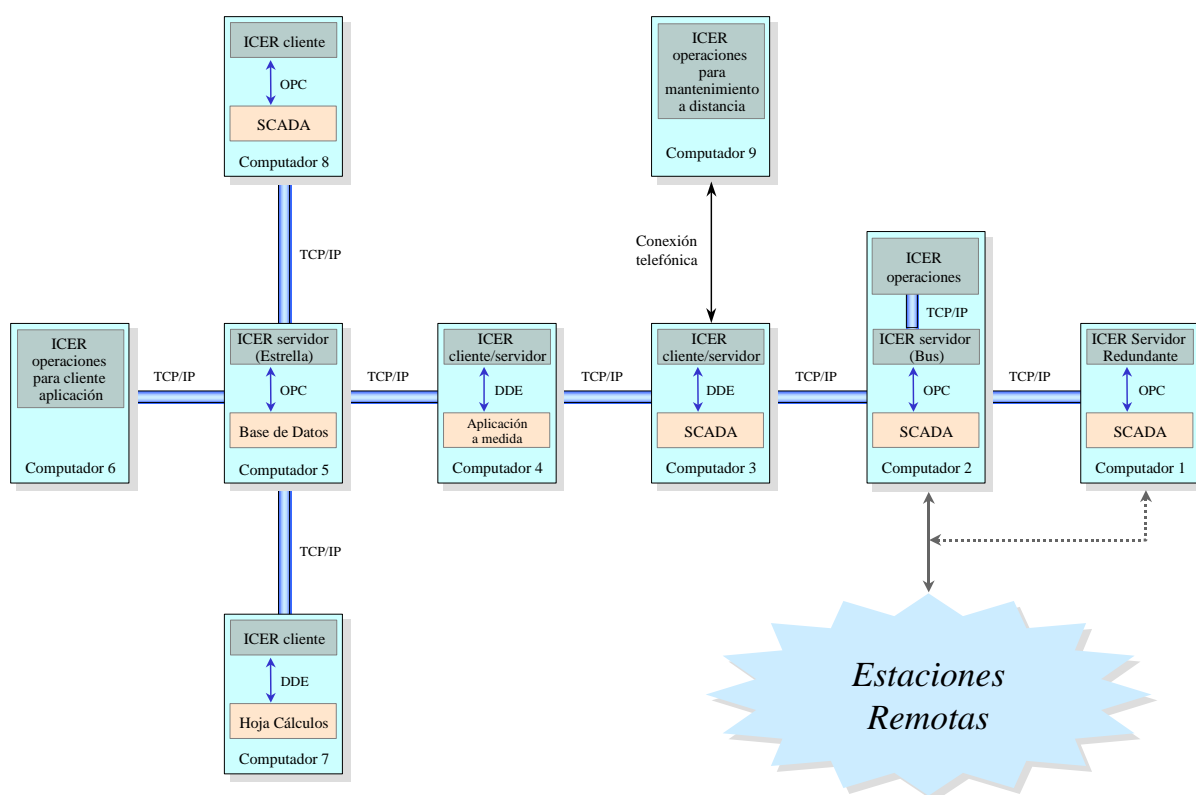


Figura 5.3: Posibilidades de conexión y configuraciones del *driver* de comunicaciones ICeR

5.3 Descripción de los módulos

5.3.1 Núcleo coordinador

Como se puede observar en la figura 5.2 el módulo coordinador es el núcleo del gestor y se encarga de repartir las tareas a realizar entre las distintas interfaces. Sus principales funciones son:

- Seleccionar de entre todas las interfaces de enlace la correspondiente según las peticiones⁸ realizadas desde la interfaz de aplicación. Para ello el núcleo dispone de dos listas dinámicas: una de los tópicos (o grupos) definidos, y otra de las interfaces de enlace por radio, SMS y RTC. Cuando se produce una petición desde la aplicación hacia las estaciones remotas, el núcleo busca primero en la lista de tópicos a que grupo de variables pertenece, y a continuación, utilizando la lista de interfaces de enlace, identifica el proceso (*thread*) que se encarga de esas variables.
- Realizar las actualizaciones necesarias según las variables modificadas como consecuencia de las recepciones de tramas desde las estaciones remotas. Según de que interfaz de enlace provenga la modificación, el núcleo busca en sus listas para obtener los tópicos que se han de actualizar, y genera mensajes para las siguientes interfaces:
 - ▶ Interfaz de aplicación: para actualizar las aplicaciones DDE/OPC.
 - ▶ Interfaz servidor: para actualizar los driver clientes.
 - ▶ Interfaz operaciones: para actualizar la ventana de visualización de tramas, los mensajes de aviso (estación no declarada o no incluida en la lista de prioridades, puerto no inicializado, etc.), y la ventana de funcionamiento (que muestra a que estación se está accediendo, si se está realizando una lectura o una escritura, si se ha producido un fallo de comunicación, etc.).
- Gestiona el temporizador para los tópicos que no pertenecen al sondeo cíclico. De forma que cuando expira el tiempo de uno de ellos se activan las variables para que en el próximo sondeo se incluyan en la trama de lectura.
- Utiliza un fichero de configuración general (*config.cfg*) que especifica los puertos físicos y servidores que se van a utilizar, los tópicos definidos, y el puerto TCP/IP de conexión. El núcleo, al ejecutarse o cuando se le indica (incluso remotamente), usa esta información para crear el resto de módulos encargados de la comunicación con las estaciones y nodos remotos.

El núcleo, junto con sus interfaces, es un programa que se ejecuta en segundo plano utilizando pocos recursos del sistema.

⁸ Correspondientes al protocolo de los estándares DDE/OPC: Advise, Request y Poke.

5.3.2 Módulo interfaz enlace radio (SMS y RTC)

El núcleo coordinador crea un módulo interfaz de enlace para cada red de estaciones remotas vía radio, mensajes cortos (SMS) o modem telefónico (RTC); y cada módulo interfaz de enlace crea su propia estructura de memoria. A continuación se describe dicha estructura y las características funcionales de los módulos de interfaz de enlace por radio, ya que para las interfaces con SMS y RTC la organización es similar. La gestión de los puertos serie y de los modems estaría a cargo de los módulos de control que crean a su vez cada uno de los módulos de interfaz (ver Figura 5.2).

5.3.2.1 Estructuras y gestión de memoria

Para manejar las comunicaciones y la información (variables) que se lee o escribe en los dispositivos de las estaciones remotas, cada módulo interfaz de enlace vía radio crea y utiliza las siguientes estructuras de memoria dinámica:

- Lista de estaciones definidas. Las estaciones se pueden crear desde la ventana de configuraciones del módulo de operaciones (descrito en la sección 5.3.5).
- Listas para el mapa de memoria de variables organizadas por estación y DBs⁹ según se muestra en la Figura 5.4. Se van creando conforme se van recibiendo mensajes de definición de variables desde la interfaz de aplicación, operaciones o cliente.

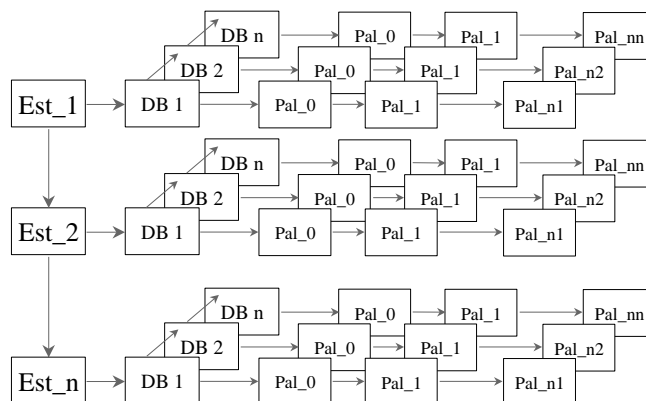


Figura 5.4: Mapa de memoria de variables para los módulos de enlaces de ICeR

- Lista para establecer las prioridades en el sondeo de las estaciones. La gestión de prioridades sólo se pueden definir desde el módulo de operaciones, y básicamente consiste en repetir y distribuir en esta lista las ocurrencias de las estaciones que son más prioritarias. El siguiente ejemplo muestra una posible definición de prioridades

⁹ Zonas de memoria identificadas por un número (DB100 por ejemplo), y organizadas en palabras de 16 bits (Bloque de Datos).

para tres estaciones según este sistema, donde se puede apreciar que la estación 1 es la más prioritaria puesto que se sondea tres veces en un ciclo, a continuación va la estación 2 que se sondea dos veces, y por último la menos prioritaria es la 3 puesto que se sondea sólo una vez en cada ciclo.

Est_1 → Est_2 → Est_1 → Est_3 → Est_1 → Est_2 →

- Lista de estaciones de alta prioridad, que define un nivel de prioridad mayor y gestionado de forma distinta al anterior. Las estaciones se pueden pasar temporalmente a esta lista de alta prioridad desde el módulo de operaciones o también desde la aplicación mediante una variable de control (el protocolo entre las aplicaciones y el driver se detalla en la sección 5.3.4). Cuando una o varias estaciones se pasan a alta prioridad en cada ciclo se sondean todas las de alta prioridad y una sola de la lista normal de prioridades, y así sucesivamente hasta que se vacíe la lista de alta prioridad. Si una estación lleva demasiado tiempo en esta lista de alta prioridad se eliminará automáticamente para que el sistema vuelva a su funcionamiento normal. En el ejemplo anterior, si pasamos la estación 3 a la lista de alta prioridad, el ciclo de sondeo resultantes sería el siguiente:

Est_3 → Est_1 → Est_3 → Est_2 → Est_3 → Est_1 → Est_3 → Est_1 → Est_3 → Est_2 →

- Lista de tramas de escrituras pendientes. Cada vez que se recibe un mensaje de escritura de variable desde la interfaz de aplicación, operaciones o cliente, se genera una trama de escritura que se introduce en esta lista.

5.3.2.2 Características funcionales

Basándose en las estructuras de memorias descritas en la sección anterior, los módulos interfaz de enlace de ICeR se encargan básicamente de las siguientes funciones:

- Gestión del sondeo cíclico. Se introduce para ocultar todas las características dependientes de la red a los niveles de aplicación, proporcionando una transferencia de datos transparente. Ofrece los siguientes servicios:
 - ▶ *Servicio de actualización cíclica de variables.* Permite el refresco continuado de las variables con los datos leídos en tiempo real de las estaciones remotas. El envío de los mensajes de petición se realiza de forma cíclica y según se hallan configurado las listas de prioridades (indicadas en la sección anterior).
 - ▶ *Servicio de forzado de variables.* Permite modificar parámetros de las estaciones remotas. A las conexiones de forzado o escritura se les da una mayor prioridad, interrumpiendo el ciclo de actualización para realizar la modificación lo más rápidamente posible.
 - ▶ *Detección de pérdidas de enlace con las estaciones remotas.* Si transcurrido un tiempo máximo de espera (*Time_Out*) el mensaje de respuesta de una estación remota no llega, se incrementa una cuenta de fallos de comunicación

correspondiente a esa estación. Este contador de fallos es reinicializado cuando se recibe alguna respuesta correcta desde la estación. Si sobrepasa el número máximo de fallos permitido se devuelve un código de pérdida del enlace para esa estación, generándose un mensaje de alarma hacia los distintos módulos de interfaz con el operador.

- Tratamiento (formateado y extracción de datos) de las tramas de comunicaciones de lectura, escritura y recepción. Tiene por objeto suministrar los medios necesarios para encauzar las tramas entre el ordenador central y los sistemas de control en planta, a través de los nodos de conmutación intermedios. En la siguiente sección se describe con detalle los aspectos más relevantes sobre el formato de las tramas utilizadas.
- Atiende las peticiones o eventos que le llegan del núcleo coordinador, y genera los mensajes correspondientes al módulo de operaciones para que este pueda visualizar tramas, así como mostrar los mensajes de atención y el estado de las comunicaciones.
- Utiliza el fichero config[nº_red].cfg para almacenar y recuperar las opciones de configuración dadas a una red de estaciones.
- Crea y empaqueta la información necesaria para generar las estadísticas de comunicación de las estaciones.
- Mantiene una lista de todas las variables definidas para las estaciones con los últimos valores actualizados.

En la Figura 5.5 se muestra un diagrama de flujo de datos donde aparecen interrelacionados las estructuras de memoria y los procesos que intervienen en estos módulos de enlace con estaciones remotas.

5.3.2.3 Protocolo y Tramas Spiral

Como se indica en [Gom-94], la elección de un único canal de radio compartido por todos los enlaces de una red de estaciones conlleva altos tiempos de conmutación con relación a los tiempos de transmisión, lo que, unido a la alta fiabilidad de la comunicación por radio, hace desaconsejable el empleo de técnicas de corrección de errores en el nivel de enlace de datos. Esto nos permitió incluir los procedimientos de control de *Enlace de datos* y de *Red* en una única capa (*Enlace-Red*, ver Figura 4.27), estableciendo un sencillo protocolo LLC orientado a byte que se basa en el siguiente formato de trama:

@	<i>Destino</i>	<i>Origen</i>	<i>Enlace</i>	<i>Información</i>	<i>Che</i>	<i>CR</i>
---	----------------	---------------	---------------	--------------------	------------	-----------

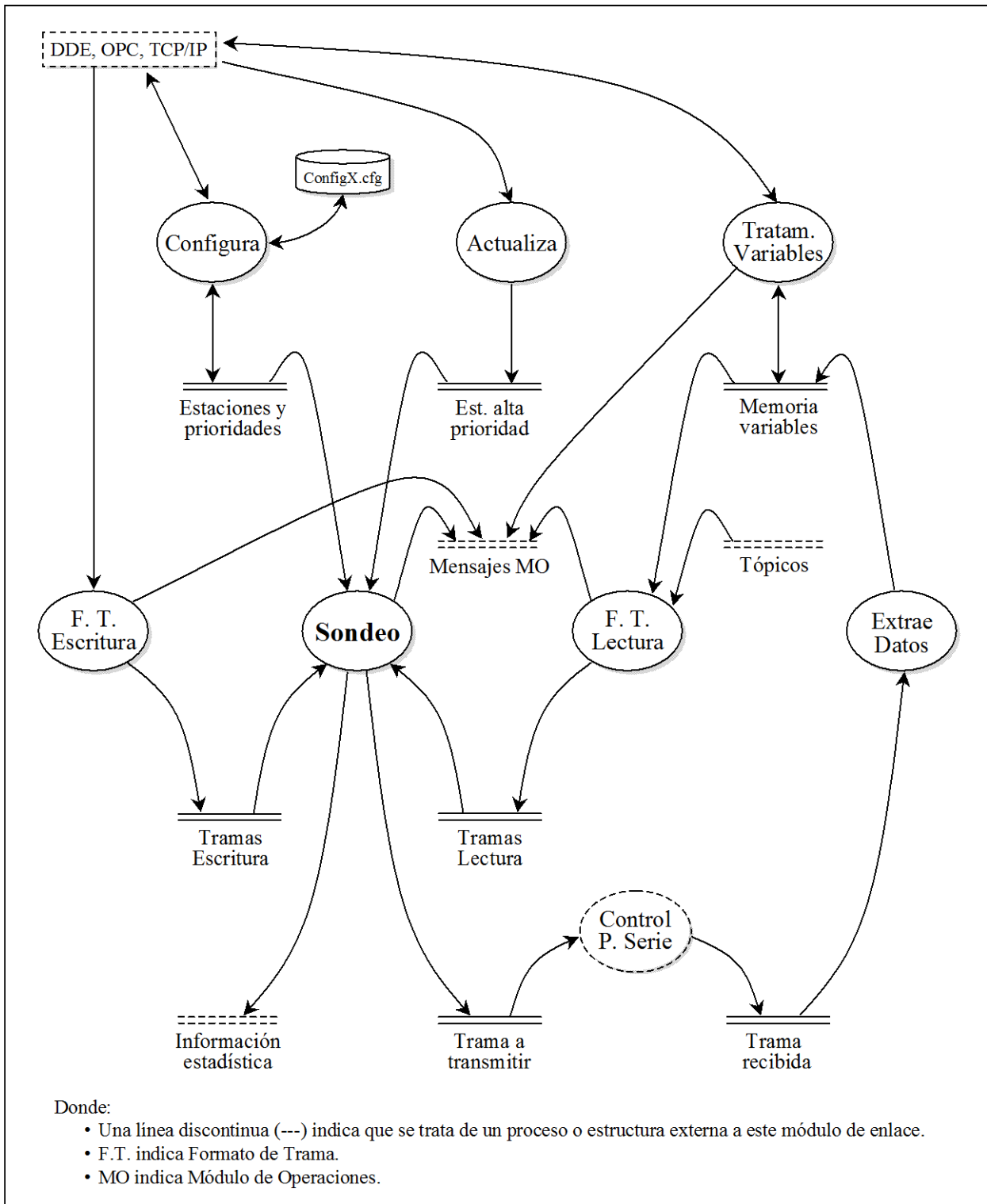


Figura 5.5: Diagrama de flujo de datos de la interfaz de enlace de ICeR

El primer campo (carácter @, 40_{Hex}) es el delimitador de inicio de trama. Los equipos receptores solo analizarán los datos recibidos a partir de este carácter y hasta que encuentren el delimitador de final de trama (carácter CR, 0D_{Hex}).

El campo **Destino** es el identificador de la estación destinataria de la información. Es una palabra de 16 bits codificada como cuatro caracteres hexadecimales, permitiendo hasta un total de 65535 estaciones. **Origen**, es un campo similar, que identifica a la estación origen del mensaje.

El siguiente campo, **Enlace**, es el identificador de la estación a través de la que el mensaje se encamina. Entre la estación origen y la destino no tiene por qué existir un enlace directo, es decir, los mensajes pueden seguir distintas rutas a través de diversos nodos hasta llegar a su destino. Al igual que los anteriores, **Destino** y **Origen**, este campo se codifica también con cuatro caracteres hexadecimales.

El campo **Información** contiene los datos que se desean enviar. El tamaño de este campo es variable. Para poder enviar en este campo cualquier carácter incluyendo los que el protocolo utiliza como caracteres de control (delimitadores de trama y ESC), a estos se les antepone el carácter de escape de control ESC (1B_{Hex}).

Che es un campo de dos caracteres de longitud, definido para reconocer posibles errores en la comunicación de la trama. Consiste en un byte codificado como dos caracteres hexadecimales, obtenidos por la operación *exclusive-OR* de todos los caracteres de la trama anteriores al campo **Che**, incluyendo el carácter @.

En cuanto a la capa de control del modelo SPIRAL que se ha utilizado ofrece dos posibilidades:

- a) Conexión transparente. Permite comunicar dos equipos mediante cualquier protocolo para el que los tiempos de conmutación a través de la red de radiomodems sean admisibles, como son los protocolos de diversos PLCs comerciales (Omron por ejemplo), *XMODEM*, etc. En este caso, los protocolos son embebidos en el campo información de las tramas anteriores.
- b) Conexión a través del protocolo de control SPIRAL. Permite comunicar equipos que utilizan un protocolo no adecuado al servicio de conexión transparente de la red (por ejemplo, protocolos con intercambio de gran número de mensajes para el establecimiento de una conexión. El protocolo que diseñamos, y que se describe a continuación, es el que se ha utilizado con los autómatas de Siemens y con SP-5000. El campo de información de la trama puede contener varias concatenaciones de acceso a datos internos de los dispositivos remotos con el formato de la Figura 5.6.

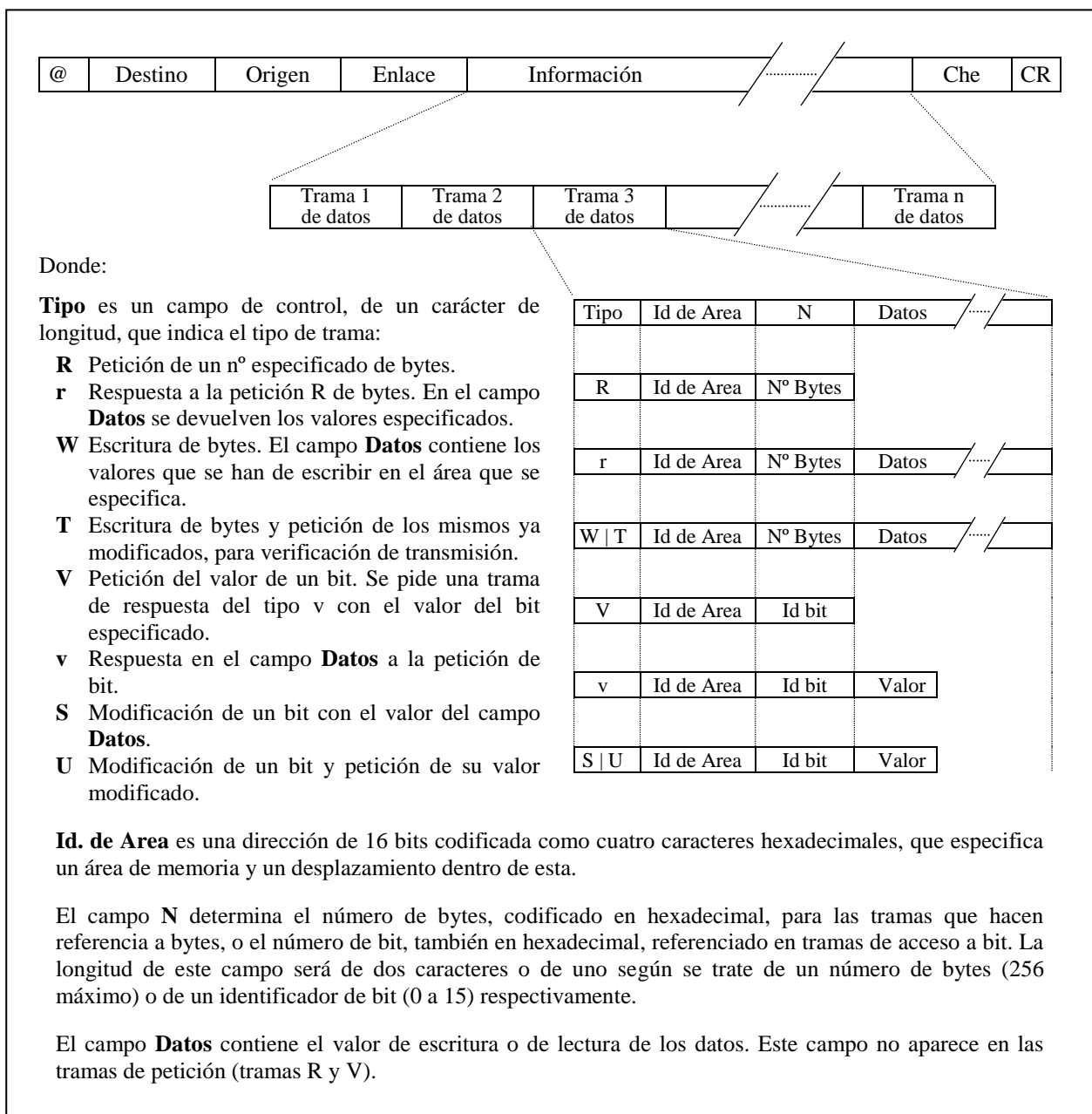


Figura 5.6: Trama y protocolo de control SPIRAL

Esta versión de ICeR aprovecha mejor la concatenación en la trama ya que se utiliza para:

- Realizar varias modificaciones simultáneas sobre una estación remota utilizando una única trama de escritura, como se venía haciendo desde las primeras versiones de este software.
- Minimizar el tamaño de la trama de recepción. ICeR, dependiendo de las variables que se le solicitan de una estación, genera una trama de lectura de forma automática eliminando aquellos huecos de memoria que no aportan información y que permiten ahorrar caracteres en la comunicación. Para ello, y teniendo en cuenta el formato de la

trama, se puede comprobar que eliminando los huecos de más de 7 caracteres se consigue reducir el tamaño de la trama de respuesta (o recepción).

ICeR dispone de otra característica que permite también minimizar el tamaño de las tramas de respuesta. Consiste en poder definir distintos grupos de variables (en adelante *tópicos*) con la posibilidad de poder distinguir entre dos tipos de información:

1. Variables que siempre se han de considerar para construir las tramas de lectura: estados del proceso o variables cuya información se necesita en tiempo real.
2. Variables que se incluyen en las tramas de lectura sólo cada cierto tiempo (configurable para cada *tópico*): consignas, valores de configuración, y variables cuya evolución no se necesita en tiempo real.

Las reducciones del tamaño de la trama que se consiguen de forma automática con estas funciones son considerables y permiten fundamentalmente minimizar los fallos de comunicaciones y acortar el tiempo que se tarda desde que se lee una estación hasta que se vuelve a leer de nuevo (en adelante *tiempo de sondeo*). En la sección 5.4 veremos algunos ejemplos concretos.

5.3.3 Módulos de interfaz para clientes, servidores y operaciones

Una de las propiedades ya indicadas de ICeR es su flexibilidad en cuanto a las posibles configuraciones que se pueden adoptar, tanto para las interconexiones cliente/servidor en puestos de trabajo de operador adicionales como para las conexiones en operaciones de mantenimiento y parametrización (ver Figura 5.3). Precisamente la parte encargada de proporcionar dicha flexibilidad son los módulos interfaz cliente, servidor y operaciones.

Estas conexiones se realizan mediante TCP/IP, con lo cual un puesto de trabajo o de operaciones puede estar ubicado en cualquier sitio que disponga de conexión a Internet/Intranet. Para ello se ha definido un protocolo específico, ver Figura 5.7, que permite un diálogo entre dos driver mediante el siguiente formato general de trama:

TICeR[_Función][_Concepto][#Parm-1][#Parm-2] ... [#Parm-n]CR

donde TICeR es un identificativo de trama para el driver de comunicaciones, CR es el delimitador de final de trama (se omite en adelante para mayor claridad), y los campos intermedios entre [] es información relativa al tipo de trama que pueden necesitarse o no dependiendo de la función que realiza.

Función	Petición Cliente	Respuesta Servidor
Reinicializar ICeR	REINICIAR	-
Eventos enlace cliente/servidor	E_ADV#Tópico#Var E_POKE#Tópico#Var#Valor	E_UPDATE#Tópico#Var#Valor E_UPDATE#Tópico#Var#Valor
Configuración de las redes	C_REDES_READ C_REDES_WRITE#Red#Puerto#Vel#Par#Bparo#Bdato C_REDES_CLEAR	C_HARD#Red#Puerto#Vel#Par#Bparo#Bdato (Se reciben tantas tramas como redes configuradas). - -
Configuración de las estaciones	C_LISTAEST_READ#Red C_LISTAEST_WRITE#Red#Id=NamEst#Id=NamEst ... C_LISTAEST_CLEAR#Red	C_LISTAEST#Red#Id=NamEst#Id=NamEst ... - -
Configuración de las prioridades	C_LISTAPRI_READ#Red C_LISTAPRI_WRITE#Red#Id#Id ... C_LISTAPRI_CLEAR#Red	C_LISTAPRI#Red#Id#Id ... - -
Configuración de los tópicos	C_LISTATOP_READ#Red C_LISTATOP_WRITE#Id#Est#Red#Tipo#Pro#Pol#Ref C_LISTATOP_CLEAR	C_LISTATOP_READ#Id#Est#Red#Tipo#Pro#Pol#Ref (Se reciben tantas tramas como tópicos configurados). - -
Visualización de tramas	- - V_VAR_READ#Tópico#Id V_VAR_WRITE#Tópico#Id#Valor	V_TRAMAS#Id#Num#Pos#Tipo#datos V_ATENCION #Red - -
Visualización sondeo estaciones	-	V_SONDEO#Red#Actual#Anterior
Visualización lista variables	V_VARIABLES	V_VARIABLES#Red[#Tópico#Est#Var] ([...] se repite para todas las variables)
Visualización de estadísticas de comunicaciones	V_EST#Red#Inicio#Final	V_EST#Red#Est#Pri#FC#TmaxFC#TmedFC#TotAcc #Lec#Esc#Tsondeo#Tciclo#Blec#Besc#Tret ... (Tantas tramas como estaciones remotas definidas)

Figura 5.7: Tramas TCP/IP utilizadas en el gestor de comunicaciones ICeR

Otra propiedad de ICeR realizada también por estos módulos de interfaz, es que se puede definir un cliente como servidor redundante, de manera que si el servidor principal (al que están conectados los equipos de comunicaciones con las estaciones remotas) se averiase, automáticamente se conmuta al servidor de respaldo y se notifica la información adecuada para que los operarios tomen las medidas oportunas. Así pues, se pueden configurar fácilmente sistemas de respaldo en aquellas aplicaciones que lo necesiten, sin tener que acudir a un software adicional, o a complicadas y costosas configuraciones del software de control y supervisión que se esté utilizando.

5.3.4 Módulos de interfaz de aplicación

Como se puede observar en la Figura 5.2, la parte de ICeR que interconecta el núcleo, y la información adquirida, con los programas de usuario que la necesitan son los módulos de interfaz de aplicación.

Una revisión detallada de los sistemas existentes (tabla de SCADA actuales, capítulo 2) nos ha llevado a la conclusión de que la mayor parte del software para el control y supervisión de procesos están basados en el entorno Windows. Por tanto, si la metodología de desarrollo que estamos proponiendo se basa en la integración de sistemas, será necesario adoptar aquellos estándares de interconectividad de procesos que se están utilizando actualmente en la plataforma más conocida y con mayor crecimiento en el ámbito del control industrial [Ayz-99]: DDE y OPC. Esta última versión permite utilizar estos dos tipos de protocolos estándares basados en la tecnología Windows de *Microsoft*.

5.3.4.1 DDE

Dynamic Data Exchange (DDE) es sencillamente un mecanismo que permite que dos aplicaciones intercambien datos automática y continuamente. El proceso DDE es análogo a una conversación entre dos personas. La aplicación que inicia la conversación se denomina cliente y la aplicación que responde, servidor. El tema de la conversación se llama tópico. Antes de que el cliente pueda comenzar una conversación con el servidor, tiene que especificar el nombre del servidor, así como el tópico y el ítem que se van a intercambiar entre las dos aplicaciones. Este ítem se refiere a los datos específicos que reconocen ambas aplicaciones.

ICeR mantiene la interfaz por DDE, actuando como servidor, para ser compatible con las versiones y aplicaciones anteriores que no permiten la conexión mediante OPC. Además, utilizando tanto DDE como OPC, el gestor se puede configurar para usar o bien la nomenclatura propia del fabricante para los dispositivos¹⁰ a los que se accede, o bien un único formato propio (en adelante FSpiral) pensado tanto para PLCs como para SP-5000. En la Figura 5.8 se muestra una tabla con la definición de dicho protocolo FSpiral, utilizado en la aplicación desarrollada para la red de abastecimiento de agua de la ciudad de Granada.

¹⁰ Actualmente ICeR incorpora los estándares para PLCs Siemens y Omron.

ICeR permite transacciones de dos tipos de variables: las que contienen datos propiamente dichos y las variables de control. El formato de estas variables, que se establece en el ítem de la conversación DDE, es el siguiente:

Variables de datos:

da_	TD.	<Num_Est>.	<Zona_Datos>.	<Desplazam.>.	<Num_bit>
-----	-----	------------	---------------	---------------	-----------

Donde:

- da_ Es un prefijo obligado que indica que se trata de una variable de datos.
- TD. Indica el tipo de dispositivo: **sp** sistema Spiral, **si** autómatas de Siemens, **om** autómatas de Omron, **al** autómatas de Alfa-Laval (ABB), y **hi** autómatas de Hitachi.
- Num_Est. Es el número de estación remota.
- Zona_Datos. Depende del tipo de dispositivo. Para "sp" debe ser 0 ó 1, para "si" indica el número de DB dentro de la estación, para "om" especifica el número de la subestación (de 0 a 32), etc.
- Desplazam. Especifica el desplazamiento (*offset*) dentro de la zona de datos anterior. Depende también del tipo de dispositivo. Para "sp" es el número de byte en un rango 0-128, para "si" es el número de palabra de 0 a 254 dentro del DB, para "om" es el número de canal (palabra de 16 bits), etc.
- Num_bit Este campo es opcional y sirve para acceder a un bit concreto del byte o de la palabra.

Variables de control:

Particulares de cada estación:

co_	TD.	<Num_Est>.	<Funcion>
-----	-----	------------	-----------

Donde:

- co_ Es un prefijo obligado que indica que se trata de una variable de control.
- TD. Indica el tipo de dispositivo (de la misma forma que para las variables de datos).
- Num_Est. Es el número de estación remota.
- Funcion
 - "Time_Out": Máximo tiempo de espera a una respuesta de la estación remota.
 - "Max_Reintentos": Número máximo de intentos permitidos para establecer comunicación. En el caso de las lecturas estos intentos no se consumen consecutivamente sino a cada vuelta del sondeo, mientras que en las escrituras sí se consumen uno tras otro.
 - "Fallo_Comunicacion": vale 1 cuando se consumen el número máximo de intentos especificados para la estación.
 - "Sondeable": vale 1 cuando queremos que la estación sea leída en los ciclos de sondeo y 0 cuando no queremos acceder a la estación.
 - "Prioridad": vale 1 cuando necesitamos pasar una estación a la cola de alta prioridad y 0 cuando se desee sacarla de dicha cola. Si transcurrido un tiempo (configurable) no se ha puesto a 0, el driver automáticamente pasa la estación a su prioridad por defecto.

- Globales:
 - "co_Estacion_Actual" : Número de estación que se está sondeando actualmente.
 - "co_Tempo_Polling" : Número de segundos que debe transcurrir entre dos peticiones de lecturas consecutivas.

Figura 5.8: Protocolo FSpiral para la interfaz de aplicación

5.3.4.2 OPC

OLE for Process Control (OPC) es un modelo de objetos y conjuntos de interfaces estándar diseñado para facilitar el intercambio de datos entre sistemas y dispositivos. Es decir, OPC simplifica la interfaz entre componentes de automatización de distintos fabricantes, así como programas y aplicaciones tales como sistemas administrativos y de visualización. Las especificaciones OPC se mantienen a través de la *OPC Foundation* [www.opcfoundation.org], o conjunto de especificaciones técnicas no-propietarias que define una serie de interfaces estándar basadas en las tecnologías OLE/COM/DCOM¹¹ de *Microsoft* [Kru-99].

En el mundo industrial, la integración de componentes desde diversos fabricantes ha resultado frecuentemente una tarea difícil. Antes de la disponibilidad de OPC, cada software requería de una interfaz específica (servidor, driver) para intercambiar datos con una determinada familia de dispositivos de campo, utilizando normalmente el estándar DDE descrito en el apartado anterior. OPC elimina esta necesidad y permite utilizar una misma operativa para comunicar aplicaciones, dispositivos y drivers. Se trata pues de un avance muy importante hacia los sistemas abiertos en el campo de los sistemas de control, que va a permitir que se emplee en un mismo sistema hardware o software de distintos fabricantes. Podríamos decir que permite hacer el "*Plug & Play*" en el mundo de la automatización. OPC es por tanto un modelo muy adaptado a la integración de sistemas en entornos de computación heterogéneos, y la mayor parte de los fabricantes de sistemas de control y supervisión lo están incorporando a sus productos.

Las aplicaciones que requieren servicios, es decir datos, de los niveles de automatización, los piden como clientes a los componentes de automatización, quienes a la vez proveen la información requerida como servidores. La idea básica de OPC consiste en normalizar la interfaz entre el servidor OPC y el cliente OPC independientemente de cualquier fabricante particular.

Los servicios prestados por los servidores OPC para clientes OPC por medio de la interfaz OPC implican normalmente la lectura, cambio y verificación de variables de proceso. Y en

¹¹ OLE (*Object Linking and Embedding*) es un interface que puede integrar información desde un programa a otra aplicación. Aquí, los objetos son creados, editados y compuestos por el servidor OLE. El OLE cliente integra estos objetos y los muestra. La norma para la interacción de componentes es definida por COM (*Component Object Model*). COM ofrece la ventaja de guardar de forma transparente todas las llamadas a un objeto, independientemente de si estas llamadas se dirigen dentro o fuera de la aplicación. Cuando se trata con llamadas entre diversos ordenadores, se habla de DCOM (COM Distribuido).

cuanto a la estructura interna, los servidores OPC se componen básicamente de objetos de tipo grupos, y estos a su vez de objetos de tipo items.

ICeR implementa este estándar OPC, actuando como servidor, de forma totalmente transparente y compatible con las aplicaciones anteriores desarrolladas para la interfaz DDE. Para ello se ha hecho una correspondencia directa entre los tópicos/items del DDE y los grupos/items del OPC. De esta forma, se puede desarrollar tanto una aplicación nueva como mejorar una realizada anteriormente, aprovechando las ventajas que ofrece OPC.

En la Figura 5.9 se puede observar como ICeR con esta nueva interfaz puede dar servicio, conjuntamente con otros servidores OPC, tanto a un SCADA como a cualquier otra aplicación que actúe como cliente OPC.

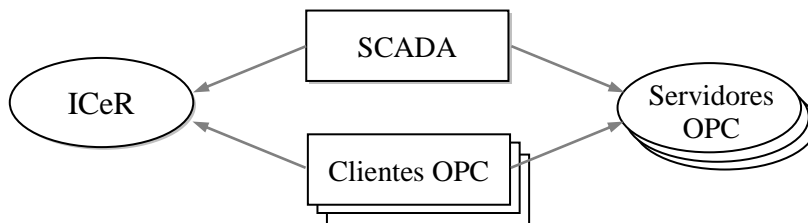


Figura 5.9: Interoperabilidad de ICeR utilizando la interfaz OPC

5.3.5 Módulo de operaciones

Este programa, ver Figura 5.10, se ha implementado fuera del núcleo. Es un programa totalmente independiente con el objetivo de que el gestor ocupe menos recursos y su ejecución sea más eficiente. Por tanto, no es necesario para el funcionamiento de ICeR, ya que este se parametriza utilizando unos ficheros de configuración (Sección 5.3.5.6), generados automáticamente desde este módulo de operaciones o también utilizando cualquier editor de textos.



Figura 5.10: Módulo de operaciones del driver de comunicaciones ICeR

El módulo de operaciones (en adelante MO) se comunica con el núcleo mediante TCP/IP a través de la interfaz de operaciones, permitiendo de esta manera conexiones tanto locales como remotas para realizar las siguientes funciones:

5.3.5.1 Configuración y estado de las comunicaciones

Para la parametrización y visualización de las comunicaciones se utilizan unas ventanas que facilitan la definición de:

- Tipo de nodo. En la sección 5.2 indicábamos que ICeR puede utilizarse para dar distintos servicios. Por tanto, cómo debe funcionar el driver se hace definiendo el tipo de nodo, que puede ser:
 - CM : Control y Monitorización. Es la configuración normal de un driver servidor o cliente de todas o parte de las estaciones remotas de una aplicación distribuida.
 - SM : Sólo Monitorización. Necesaria para aquellos nodos en los que sólo se desee una supervisión sin posibilidad de actuación sobre el sistema.
 - SR : Servidor Redundante. Se trata de un cliente CM que además supervisa el funcionamiento de un servidor CM para conmutar de forma automática en caso de avería. Si se selecciona esta opción se ha de especificar la dirección TCP/IP del servidor principal.
- Lista de redes. En este apartado se identifican las redes que se van a manejar, y se define para cada una de ellas el tipo de enlace o medio por donde va a llegar la información (Serie, TCP/IP, GSM, RTC).
- Lista de estaciones para cada red. En la Figura 5.11 se observa la ventana desde donde se puede monitorizar el sondeo de las estaciones definidas para cada red, asignar las listas de prioridades (alta y por defecto), y modificar los siguientes parámetros de comunicación: número de reintentos, tiempo máximo de espera de una trama de respuesta, y si la estación se ha de sondear.
- Lista de tópicos. En esta ventana se configuran, para cada grupo de variables de una red, las siguientes propiedades: protocolo o formato a utilizar en el direccionamiento de las variables de la aplicación SCADA (FSpiral o Standard del fabricante del equipo), estación y tipo de estación (Siemens u Omron) en el caso de elegir el formato propio del fabricante ya que para FSpiral esta información está especificada en el propio formato (ver sección 5.3.4.1), y características del sondeo para ese grupo de variables (siempre o cada cierto tiempo).

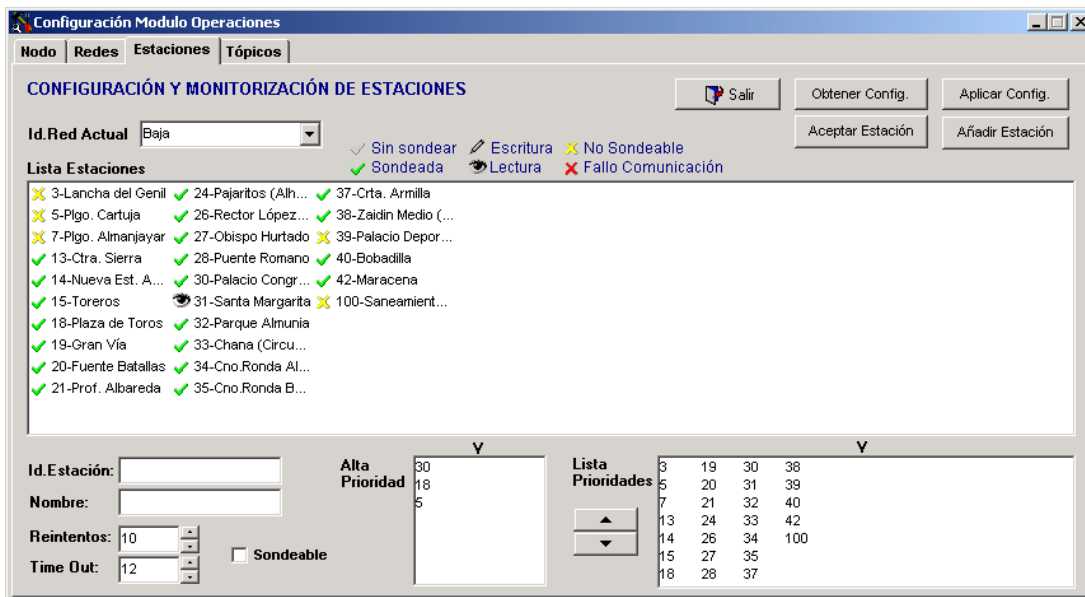


Figura 5.11: Ventana de configuración y monitorización de ICeR

5.3.5.2 Visualización de tramas

Uno de los servicios más utilizados del MO es la visualización de las tramas de comunicaciones con las estaciones remotas, así como la notificación de los eventos en la parametrización. Más concretamente con esta ventana (Figura 5.12) se puede: monitorizar el envío y recepción de tramas para cada estación, estudiar el formato y tamaño de esas tramas, generar manualmente una trama de lectura o escritura sobre uno de los tópicos de una de las redes definidas, y detectar fallos en el direccionamiento de las variables de la aplicación SCADA.

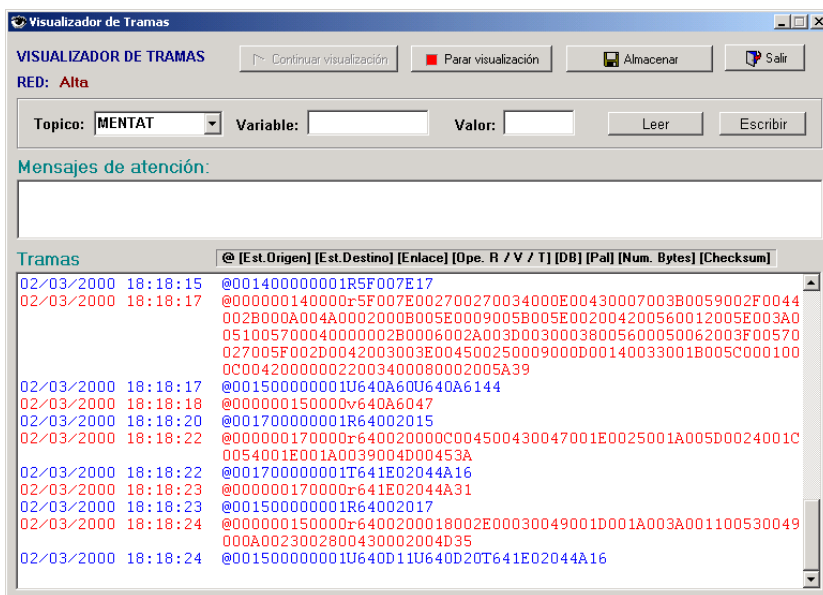


Figura 5.12: Ventana de visualización de tramas de comunicaciones de ICeR

5.3.5.3 Generación de estadísticas de comunicaciones

Se ha incluido también una ventana de visualización de estadística de comunicaciones en intervalos de tiempo configurables por el operador (ver Figura 5.13), con el objetivo de poder comprobar y estudiar el funcionamiento y evolución de las comunicaciones. En concreto, la información que se proporciona para cada estación de una red es la siguiente:

Prior :	Número de ocurrencias dentro de la lista de prioridades. Cuanto mayor sea este número mayor prioridad asignada.
Nº FC :	Numero de fallos de comunicaciones reales, es decir, independientemente del número de reintentos configurados por la aplicación.
Max.T.FC :	Tiempo máximo sin comunicación (en minutos).
Med.T.FC :	Tiempo medio sin comunicación (en minutos).
Tot. Acc. :	Número total de accesos (tanto de lecturas como de escrituras).
Nº Lec :	Numero total de lecturas realizadas.
Nº Esc :	Numero total de escrituras realizadas.
T.Sondeo :	Tiempo desde que se sondea una estación (lectura) hasta que se vuelve a sondear de nuevo (en segundos).
T.Ciclo :	Tiempo necesario para sondear todas las estaciones definidas en la lista de prioridades (en segundos).
%T.Sondeo/T.Ciclo :	Porcentaje entre el tiempo de sondeo y el tiempo de ciclo.
Max.B.Lec :	Máximo numero de bytes en tramas de lectura.
Med.B.Lec :	Media del numero de bytes en tramas de lectura.
Max.T.Ret :	Retardo máximo en realizar la operación de lectura (en segundos).
Med.T.Ret :	Retardo medio en realizar la operación de lectura (en segundos).
%Max.T.Ret/Med.T.Ret :	Porcentaje entre el máximo y la media del tiempo de retardo.

Nº Est.	Prior.	Nº FC	T.Max.FC	T.Med.FC	Tot. Acc.	Nº Lec.	Nº Esc.	T.Sondeo	T.Ciclo	% T.Sondeo/T.Ciclo	Max. B. Lec	Med. B. Lec	T.Ret.Max	T.Ret.Med	% TRMed/TRMax
7	0	181	831,2	204,9	515	506	9	114,5	150,1	76 %	152	111	3,3	2,7	82 %
13	0	18	137,9	80,8	513	507	6	114,4	150,1	76 %	104	74	3,1	2,4	77 %
14	0	40	351,2	107	511	505	6	114,4	150,1	76 %	104	74	5,1	4,2	83 %
15	0	56	326,5	126,1	511	505	6	114,4	150,1	76 %	104	75	7,2	5,4	75 %
18	0	34	403,5	110,2	515	506	9	114,4	150,1	76 %	108	75	5	4,4	86 %
19	0	39	242	92,8	515	506	9	114,5	150,1	76 %	112	91	5,2	4,4	86 %
20	0	16	284,4	65,6	512	506	6	114,5	150,1	76 %	126	93	4	2,6	66 %
21	0	11	133,7	64,6	512	506	6	114,4	150,1	76 %	108	75	3,1	2,3	75 %
24	0	43	266,5	103,3	511	505	6	114,4	150,1	76 %	104	78	6,8	4,3	63 %
26	0	51	345,7	122,7	511	505	6	114,4	150,1	76 %	108	76	6,5	5,4	83 %
27	0	40	209,2	103,9	513	507	6	114,4	150,1	76 %	108	75	4	2,9	73 %
28	0	12	151,5	68,1	516	507	9	114,4	150,1	76 %	108	75	2,9	2,4	82 %
30	0	16	148,8	62,4	512	506	6	114,4	150,1	76 %	108	75	3	2,3	78 %
31	0	31	261,4	92,6	512	506	6	114,5	150,1	76 %	104	75	4,7	4,1	86 %
32	0	25	241,5	98,7	513	507	6	114,4	150,1	76 %	108	75	5,1	4,3	85 %
33	0	81	2944,2	548,5	512	506	6	114,4	150,1	76 %	108	81	5,1	4,5	87 %
34	0	24	268,2	90	520	507	13	114,4	150,1	76 %	138	117	5,7	5,2	90 %
35	0	25	167,7	86,5	515	506	9	114,5	150,1	76 %	108	76	2,9	2,4	81 %

Figura 5.13: Ventana de visualización de las estadísticas de comunicaciones de ICeR

5.3.5.4 Gestión de variables

Visualización (con filtros de búsquedas) de toda la lista de variables. El operador en cualquier momento puede solicitar una lista parcial o completa de las variables definidas en la aplicación para verificar el direccionamiento y comprobar el valor actual. Permite también generar esa misma información en un fichero para su tratamiento posterior.

5.3.5.5 Registro y control de acceso

ICeR se puede proteger contra copias ilegales desde este módulo de operaciones introduciendo una clave adecuada. En cuanto al control de acceso, inicialmente sólo se permite acceder a las opciones de conexión y visualización de variables y estadísticas de comunicaciones; si introducimos una clave de nivel medio, se puede acceder además a la visualización de tramas, y si introducimos la clave de nivel alto también se puede entrar en las opciones de configuración del driver.

5.3.5.6 Generación de ficheros de parametrización

El MO de ICeR crea y actualiza unos ficheros de configuración donde se almacena toda la información y propiedades definidas sobre las redes de estaciones. La estructura de estos ficheros se muestra a continuación:

Config.cfg Contiene la configuración de las redes de estaciones.

[TCPIP]

Port = #Puerto de escucha del driver.

[TIEMPO_PRIORIDAD]

Tprioridad = #Tiempo máximo (en segundos) permitidos en alta prioridad.

[TIEMPO_INACTIVIDAD]

Tiempo = #Tiempo máximo (en segundos) con las escrituras inhabilitadas al inicializar el driver.

[Puerto#Identificador]

Puerto = { 1,2,3,4 }

Descriptor = { Nombre de la red }

Tipo = { 0=Serie,1=Cliente,2=Redundante }

Activo = { lain0=Inactivo,1=Activo }

Velocidad = { 1200,2400,4800,9600,38400 }

Paridad = { 0=NONE,1=PAR,2=IMPAR }

Stop = { 0=1,1=1.5,2=2 }

Datos = { 7,8 }

AddressTCPIP1 = { Dirección TCPIP }

PortTCPIP1 = #Puerto
AddressTCPIP2 = {Dirección TCPIP}
PortTCPIP2 = #Puerto

[#Nombre Tópico]
Estación = #Numero de la estación
Puerto = #Identificativo del Puerto
Tipo = {0=Siemens,1=Omron}
InPolling = {0=Por tiempo de refresco, 1=Siempre}
Protocolo = {0=Spiral,1=Standard}
Refresco = #Tiempo de Refresco del Tópico en milisegundos.

Config#NPuerto.cfg Contiene la configuración específica de cada red definida.

[Estadísticas]
Arranque = Fechas en las que se arranco el drivers.
Apagar = Fechas en las que se apago el drivers.

[Estaciones]
#Numero de la estación = #Identificativo - #Time Out - Max Reintentos - Sondeable

[FalloComunicacion]
Est_#Nestacion = Fecha en la que se produjo un fallo de comunicación con la estación.

Para cada estación se crea un fichero con el nombre Est-#Nred-#NEstacion-#Mes-#Año.cfg, donde, diariamente y en un formato no legible, se guarda la información necesaria para generar las estadísticas.

5.4 Aplicación al abastecimiento de agua de la ciudad de Granada

En cualquier sistema de control y supervisión basado en estaciones remotas, y en concreto para la red de abastecimiento de la ciudad de Granada, un gestor de comunicaciones como el que se ha desarrollado, permite que a través de la cada vez más extendida Internet/Intranet se puedan añadir puestos de operadores locales o remotos con capacidad de acceder a la información suministrada por las estaciones remotas.

En el caso que nos ocupa, ICeR está configurado en la estación central para gestionar dos redes de estaciones remotas: Red de alta (11 estaciones equipadas con autómatas de Siemens) y Red de baja (24 estaciones equipadas con SP-5000). En la subcentral de la ETAP también se ha instalado este driver para gestionar una red de 3 estaciones necesarias para los sondeos de emergencia (Ver Figura 4.3 del capítulo 4).

Para dar mayor fiabilidad y robustez a la aplicación de control y supervisión, se ha utilizado la posibilidad que tiene ICeR de configurar uno de los clientes como servidor redundante, que hace de sistema de respaldo al que se conmuta automáticamente en caso de avería del servidor principal.

La utilización de este nuevo software nos ha permitido además poder supervisar y mantener las comunicaciones a distancia, así como poder disponer de los datos en tiempo real en nuestro departamento, como un puesto de operador más, para los diversos estudios y proyectos de investigación que se han realizado y en los que se están trabajando en cuanto a la modernización y optimización de los sistemas de distribución de agua.

También se ha instalado en otras muchas aplicaciones, algunas de ellas relacionadas con la gestión del agua¹², convirtiéndose por tanto en unos de los productos más importantes utilizados por la empresa ICR S.A. en la integración de sistemas. En la sección siguiente se dan datos concretos (estadísticas) del comportamiento de las comunicaciones con las estaciones remotas existentes en la ciudad de Granada para la gestión del agua.

¹² En el apéndice A se da una lista aproximada de los proyectos, realizados conjuntamente con la empresa ICR S.A., donde se ha utilizado este driver de comunicaciones.

5.5 Medidas y pruebas experimentales

En esta sección vamos a comprobar, tanto de forma simulada como en aplicaciones reales, como algunas de las mejoras incorporadas a esta última versión de ICeR pueden acelerar las comunicaciones de una red de estaciones que utilice el sistema SPIRAL.

5.5.1 Resultados de simulaciones

Para observar como se optimiza el tamaño de las tramas de comunicaciones y como consecuencia el tiempo de acceso a la información, se considera como ejemplo una red de 10 estaciones (2 de ellas con prioridad alta, 4 con prioridad media y el resto, otras 4, con prioridad baja), comunicaciones a 1200 baudios, tiempo de conmutación de radio de 1 segundo, y un total de 30 palabras a acceder organizadas en la memoria de la siguiente forma:

5 - A	5 - B	15 - H	5 - A	5 - B	15 - H	10 - C
-------	-------	--------	-------	-------	--------	--------

Donde: 5 - A indica 5 palabras del tipo A (necesarias cada sondeo, estados del proceso)

5 - B indica 5 palabras del tipo B (necesarias cada 15 minutos, totalizadores)

10 - C indica 10 palabras del tipo C (necesarias cada hora, consignas o parámetros)

15 - H indica 15 palabras de hueco o sin correspondencia con ninguna variable

En la tabla de la Figura 5.14 se muestran los resultados de las distintas optimizaciones que se pueden conseguir, donde:

- La simulación 1 corresponde a los resultados obtenidos sin ninguna de las mejoras introducidas en el ICeR. Se puede observar que en cada ciclo se lee todo el bloque de memoria (60 palabras más los bytes de control).
- La simulación 2 corresponde a los resultados debido a la detección y eliminación automática de huecos o palabras a las que no se necesita acceder. Se observa como se reduce el tamaño de la trama sin necesidad de modificar ni la aplicación ni la ubicación de las variables en el sistema de control distribuido (PLC normalmente).
- La simulación 3 corresponde a los resultados debido a la asignación de distintos tópicos a variables en función del tipo de estas. Se comprueba que se vuelve a reducir el tamaño de la trama y por consiguiente los tiempos de acceso y de sondeo, asignando adecuadamente los tópicos a las variables de la aplicación.
- Y la Simulación 4 corresponde a los resultados debido a la posibilidad de establecer distintas prioridades a las estaciones en función de su importancia o necesidades de acceso, con la consiguiente disminución del tiempo de sondeo para las más prioritarias.

Nº Est.	Simulación 1			Simulación 2			Simulación 3			Simulación 4			
	T.Son.-T.Cic.	MaxBL-MedBL	MaxTR-MedTR	T.Son.-T.Cic.	MaxBL-MedBL	MaxTR-MedTR	T.Son.-T.Cic.	MaxBL-MedBL	MaxTR-MedTR	Pri.	T.Son.-T.Cic.	MaxBL-MedBL	MaxTR-MedTR
1	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	4	11,0-44,5	198-100	2,9-2,0
2	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	2	22,5-44,5	198-101	2,9-2,0
3	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	2	22,5-44,5	198-101	2,9-2,0
4	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	1	44,5-44,5	198-102	2,9-2,0
5	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	4	11,0-44,5	198-100	2,9-2,0
6	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	2	22,5-44,5	198-101	2,9-2,0
7	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	2	22,5-44,5	198-101	2,9-2,0
8	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	1	44,5-44,5	198-102	2,9-2,0
9	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	1	44,5-44,5	198-102	2,9-2,0
10	38,9	286	3,8	29,6	198	2,9	21	198-100	2,9-2,0	1	44,5-44,5	198-102	2,9-2,0

Donde:

T.Son.= Tiempo (seg.) desde que se sondea una estación (lectura) hasta que se vuelve a sondear de nuevo.

T.Cic.= Tiempo (seg.) necesario para sondear todas las estaciones definidas en la lista de prioridades.

MaxBL = Máximo número de bytes en tramas de lectura.

MedBL = Media del número de bytes en tramas de lectura.

MaxTR = Retardo máximo (seg.) en realizar la operación de lectura.

MedTR = Retardo medio (seg.) en realizar la operación de lectura.

Pri. = Número de ocurrencias dentro de la lista de prioridades (cuanto mayor sea mayor prioridad asignada)

Cuando en una columna aparece un solo dato en vez de dos (XXX en lugar de XXX-YYYY) es porque coinciden ambos resultados.

Estos resultados se han generado mediante una aplicación ejemplo, el driver ICeR, y un simulador (SICeR) capaz de generar todo tipo de tramas Spiral a través de un puerto serie, e implementado para poder realizar las pruebas de laboratorio.

Figura 5.14: Tabla de resultados de mejoras que consigue ICeR para un ejemplo de aplicación

En resumen, podemos ver como el tamaño de las tramas de comunicaciones se han reducido de 286 bytes a 100 bytes en el sondeo cíclico que se realiza continuamente sobre las estaciones remotas, lo cual ha significado una mejora en los tiempos de retardo de cada acceso de 3,8 a 2,0 segundos. También podemos observar cómo la posibilidad de asignar prioridades a las estaciones más críticas permite reducir sus tiempos de ciclos de 21 (o 38,9 sin optimización de tramas) a 11 segundos.

5.5.2 Resultados en aplicaciones reales

Para comprobar, con datos en instalaciones reales, como ha influido ICeR en las tres redes de comunicaciones del sistema de supervisión y control del abastecimiento de agua de la ciudad de Granada, se proporcionan a continuación tablas con los resultados del comportamiento de estas tres redes (alta, baja y bombeos de emergencia) antes y después de introducir este nuevo software de comunicaciones.

5.5.2.1 Red de alta

La red de alta está formada por 11 estaciones remotas, unas más complejas y con mayores necesidades que otras. Es la red que lleva más años funcionando, sobre la que se ha hecho más ampliaciones y actualizaciones, y la que soporta la automatización y optimización de los grupos de bombeos a depósitos y de las estaciones reductoras de presión, además de otros procesos

como apertura/cierre de válvulas/compuertas, control de caudalímetros, etc. Por lo tanto, la cantidad de información a la que se necesita acceder, sobre todo en algunas de estas estaciones, es ya tan grande que la probabilidad de fallo y los tiempos de acceso han aumentado demasiado como se puede apreciar en la tabla de la Figura 5.15, donde además se puede observar que utilizando el nuevo software de comunicaciones se consiguen las siguientes mejoras:

- Se consigue reducir la relación de fallos de comunicaciones con respecto a los accesos totales a la estación (Rel. FC/Acc.), sobre todo en aquellas estaciones que más cantidad de información necesitan transmitir (estaciones 10, 8 y 4). Se aprecia también una pequeña disminución en la duración media de los fallos de comunicación (Med. T. FC, en segundos).
- Se acorta el tiempo de sondeo (T. Sondeo) para todas las estaciones por igual (todas las estaciones tienen la misma prioridad) en 28 segundos, que equivale a casi un 20 % de ahorro de tiempo en total. Esta reducción de tiempo es importante sobre todo si se tiene en cuenta que se ha conseguido sólo por software, es decir, sin cambiar ningún hardware o equipo de comunicaciones y ningún direccionamiento en los sistemas de control distribuido de las estaciones remotas.
- Se aprecia también la disminución del número medio de bytes de información que se han de transmitir por estación en los accesos de lectura (Med. B. Lec.), en torno a un 30% en total. Como es lógico esta reducción en los tamaños de las tramas afecta también a los tiempos medios de retardo en los accesos de lectura (Med. T. Ret., en segundos), que se pueden comparar con los anteriores y con los máximos (Max. T. Ret., en segundos).

Nº Est.	Rel. FC/Acc. (%)		Med. T. FC		T. Sondeo		Med. B. Lec.		Max. T. Ret.		Med. T. Ret.	
	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora
2	6,1	6,0	75	74	146	118	94	86	3,7	3,5	3,4	3,1
4	6,3	3,5	80	40	146	118	280	191	7,1	6,9	6,9	5,3
5	5,2	3,1	65	50	146	118	242	202	6,6	6,1	6,1	5,4
8	6,8	4,0	45	40	146	118	334	224	7,8	7,7	7,6	5,9
9	4,9	3,1	32	32	146	118	338	206	8,2	7,8	7,9	5,7
10	9,6	2,5	60	20	146	118	512	330	11,3	11,2	11,1	8,1
15	6,0	5,1	75	70	146	118	94	86	3,7	3,5	3,4	3,1
20	7,2	6,5	25	19	146	118	298	170	7,4	7,2	7,2	5,0
21	6,4	5,2	135	100	146	118	110	88	3,5	3,5	3,5	3,1
23	6,5	6,1	85	82	146	118	130	108	4,5	3,8	3,8	3,5
24	6,4	6,0	80	75	146	118	130	108	4,5	3,8	3,8	3,5
Med.	6,4	4,6	68	54	146	118	232	163	6,21	5,91	5,88	4,7

Figura 5.15: Tabla comparativa de datos de comunicaciones utilizando ICeR en la red de alta

5.5.2.2 Red de baja

Esta constituida, actualmente, por 24 estaciones remotas con las que se necesita transmitir menor cantidad de información (y como consecuencia tramas de menor tamaño) que en el caso de las estaciones de la red de alta. Sin embargo, debido a que la cobertura para estas estaciones es peor (las antenas están incluidas dentro de los armarios), cualquier reducción, por pequeña que sea, en el tamaño de la trama mejorara las comunicaciones, sobre todo en la relación fallo de comunicaciones con respecto al número de accesos totales (Rel. FC/Acc.), como se puede apreciar en la tabla de la Figura 5.16.

N° Est.	Rel. FC/Acc. (%)		Med. T. FC		T. Sondeo		Med. B. Lec.		Max. T. Ret.		Med. T. Ret.	
	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora	Antes	Ahora
3	24,4	13,6	353	283	136	115	104	76	4,2	4,2	4,1	3,5
7	36,0	21,5	581	375	136	115	152	116	4,1	3,3	4,0	2,8
13	3,9	3,0	86	77	136	115	104	72	2,9	2,9	2,9	2,4
14	6,4	6,0	117	109	136	115	104	73	5	4,9	5	4,2
15	16,1	11,1	155	239	136	115	104	73	6,8	6,5	6,7	5,3
18	8,3	5,7	111	109	136	115	106	73	5	5,2	5	4,3
19	6,3	5,1	109	98	136	115	110	91	5,4	4,9	5,3	4,4
20	3,1	2,8	79	69	136	115	124	91	4	3,9	4	2,6
21	3,6	2,9	78	65	136	115	106	73	2,8	2,9	2,8	2,3
24	13,5	8,3	311	115	136	115	104	75	5	5	5	4,3
26	16,7	6,8	405	113	136	115	106	73	6,9	6,4	6,5	5,3
27	6,1	5,7	124	93	136	115	106	73	4,5	4,4	4,5	2,9
28	2,5	2,2	68	64	136	115	106	73	2,9	2,8	2,8	2,3
30	3,9	2,1	105	57	136	115	106	73	3,6	3,3	3,4	2,3
31	7,3	3,8	128	95	136	115	104	73	5,1	4,7	5	4
32	7,5	7,0	135	119	136	115	108	73	5,2	5,1	5,2	4,3
33	10,9	10,4	167	147	136	115	106	80	5,8	5,1	5,5	4
34	6,6	4,6	121	114	136	115	136	117	5,7	5,6	5,7	5,2
35	10,3	7,6	121	104	136	115	106	73	3,9	3,8	3,9	2,3
37	9,6	5,9	123	88	136	115	142	117	7	6,9	7	5,3
38	10,4	9,2	198	162	136	115	140	99	9,2	9,1	9,1	4,8
40	6,9	6,1	107	105	136	115	104	73	5	4,9	5	4,2
42	14,5	13,7	449	354	136	115	62	40	2,5	2,4	2,5	2,2
100	5,4	5,4	115	100	136	115	68	68	3,3	3,2	3,3	3
Med.	10,01	7,1	181	135	136	115	109	79	4,83	4,64	4,76	3,68

Figura 5.16: Tabla comparativa de datos de comunicaciones utilizando ICeR en la red de baja

5.5.2.3 Sondeos de emergencia

Las comunicaciones por radio para acceder a toda la información de los sondeos de emergencia tienen el problema de que en una única estación (maestro de comunicaciones de la red de autómatas) se encuentran casi todos los datos que se han de transmitir a la central (situada en la ETAP) relacionados con esta instalación. Existen otras estaciones remotas para sondeos individuales y depósitos en las cuales no existe este problema, y que por tanto no las vamos a considerar en este estudio. Así pues, los datos que se transmiten originalmente se ubicaron en tres zonas de memorias (DBs), y se leían un total de 250 palabras, entre las cuales existían palabras sin correspondencia con variables de la aplicación y palabras que no tienen por qué leerse en cada sondeo de la estación.

Por tanto, antes de la instalación de esta nueva versión de ICeR las comunicaciones obedecían a los siguientes parámetros:

Rel. FC/Acc. = 30 % Med. B. Lec. = 1100 bytes Med. T. Ret. = 13,5 seg.

Y actualmente, utilizando este nuevo software de comunicaciones, el comportamiento es el siguiente:

Rel. FC/Acc. = 11 % Med. B. Lec. = 248 bytes Med. T. Ret. = 4,8 seg.

Es decir, se ha reducido el tamaño de las tramas de lectura en más del 75 %, lo que ha supuesto una mejora clara, aproximadamente un 20 %, en cuanto a la probabilidad de que se produzcan fallos de comunicaciones, ya que el tiempo medio de retardo en un acceso de lectura baja considerablemente, en más del 60 %.

5.6 Conclusiones

Como herramienta orientada a la integración de sistemas en este capítulo se ha descrito el diseño e implementación de un gestor de comunicaciones orientado al control y supervisión de sistemas distribuidos.

La arquitectura de este software esta basada en el concepto de modularidad y estandarización, y ofrece una gran variedad de posibilidades en cuanto a modos de acceso (DDE, OPC), topologías de interconexión (TCP/IP), medios de comunicación (radio, RTC, GSM), y manejo de dispositivos de distinto tipo (PLCs, SP-5000, etc.).

Se ha mostrado la utilización del programa en una aplicación real y se han obtenido resultados (tanto en simulaciones como experimentalmente) que demuestran las ventajas que supone su utilización. Por lo tanto se trata de un producto totalmente terminado que está siendo explotado por la empresa ICR S.A, de tal forma que se está instalando en aplicaciones nuevas, y sustituyendo a otras versiones más antiguas en aplicaciones ya existentes para aprovechar las mejoras que introduce en cuanto a gestión de prioridades, optimización del tamaño de las tramas y del tiempo de sondeo, control de redundancia, mayores posibilidades de interconexión, mantenimiento a distancia, y generación automática de estadísticas.

Capítulo 6:

Concepción y diseño del sistema Spiral-5000

En la implementación de automatismos en procesos industriales en general se suelen utilizar PLCs para asumir las funciones de adquisición y control, pero como ya se indicaba en el Capítulo 4 de esta memoria, no son la única solución. La aplicación de PCs industriales en control es otra de las tendencias que en estos últimos años está adquiriendo un notable protagonismo, fundamentalmente por la gran diversidad de dispositivos, software y normalizaciones en comunicaciones que permiten integrar [Bag-97, Mar-99a, Mar-99b].

Este capítulo está dedicado a la concepción y diseño de un sistema de control distribuido alternativo a los PLCs y con las ventajas de un PC industrial, para facilitar la integración de sistemas dentro del modelo SPIRAL con el que se está trabajando.

Después de la introducción general en la que se justifica su diseño, en la sección 6.2 se hace una primera descripción a nivel de bloque del sistema, para a continuación, secciones 6.3 y 6.4, mostrar la arquitectura interna y las características funcionales de los módulos más importantes. En la sección 6.5 se tratan los aspectos relacionados con la programación de este dispositivo, y por último, en la sección 6.6, se dan las conclusiones para este capítulo.

6.1 Introducción

El desarrollo del sistema SPIRAL-5000 surgió de la necesidad de disponer de un equipo que se adaptara a las especificaciones concretas de uno de los proyectos más innovadores y ambiciosos del sistema de explotación de la red de distribución de agua de Granada, la sectorización de la ciudad. Se necesitaba un sistema potente pero a la vez sencillo y económico, que se adaptase a las necesidades actuales y futuras de más de 40 sectores, y que a su vez fuese idóneo para otras aplicaciones similares de control distribuido. Los requisitos de dicho sistema eran:

- **Canales de comunicaciones.** Es frecuente que un único controlador necesite comunicarse vía radio, terminal GSM y/o modem RTC. Simultáneamente en algunas ocasiones es necesario también emplear un bus de campo para comunicarse vía cable con otras unidades, módulos de ampliación de E/S, instrumentación serie, terminales de operador, etc. Para disponer de un PLC con estas características de comunicaciones hay que seleccionarlo entre los de gama alta con el consiguiente coste añadido.
- **Programación en C estándar** para control y comunicaciones. Es conveniente disponer de este lenguaje para implementar algoritmos de control complejos y protocolos de comunicación específicos, tareas que usualmente se desarrollan con mucha dificultad empleando PLCs. Además se deberá simplificar la programación de la lógica de control y la configuración de las comunicaciones.
- **Capacidad de memoria.** En ocasiones se necesita un tamaño de memoria suficiente como para almacenar registros de datos de varios días, semanas o incluso meses (data logger) para un volcado o transmisión eventual de la información (telectura de contadores y almacenamiento local de históricos).
- **Bajo consumo.** El sistema debe ser capaz de funcionar con paneles fotovoltaicos y baterías para su utilización en ubicaciones donde no haya suministro de red eléctrica.
- **Dimensiones reducidas.** Es usual tener que incluir controladores locales fácilmente en pequeños cuadros de control, donde es complicado utilizar PC industriales o PLCs con características funcionales similares.
- **Entradas/Salidas.** En una sola unidad se deberá incluir un conjunto suficiente de canales de conversión A/D de alta resolución, así como de entradas y salidas digitales. En caso de necesidad de ampliación es conveniente disponer de módulos de muy bajo coste que puedan incluso conectarse con la unidad central vía bus de campo a distancias considerables.

Con estos criterios de diseño se ha desarrollado el sistema SPIRAL-5000 que aparece en las fotografías de la Figura 6.1.



Figura 6.1: Fotografías sistema SPIRAL 5000

6.2 Descripción general del sistema.

El sistema SPIRAL-5000 está compuesto de un módulo principal SP-5000 al cual se le puede conectar (vía RS-485) un conjunto de módulos de ampliación (SP-16ED, SP-8REL, SP-DA), así como diversas interfaces externas de transmisión de datos (radiomodem, modem telefónico, GSM, interfaz de infrarrojos, etc.). La Figura 6.2 muestra un diagrama de bloques del sistema.

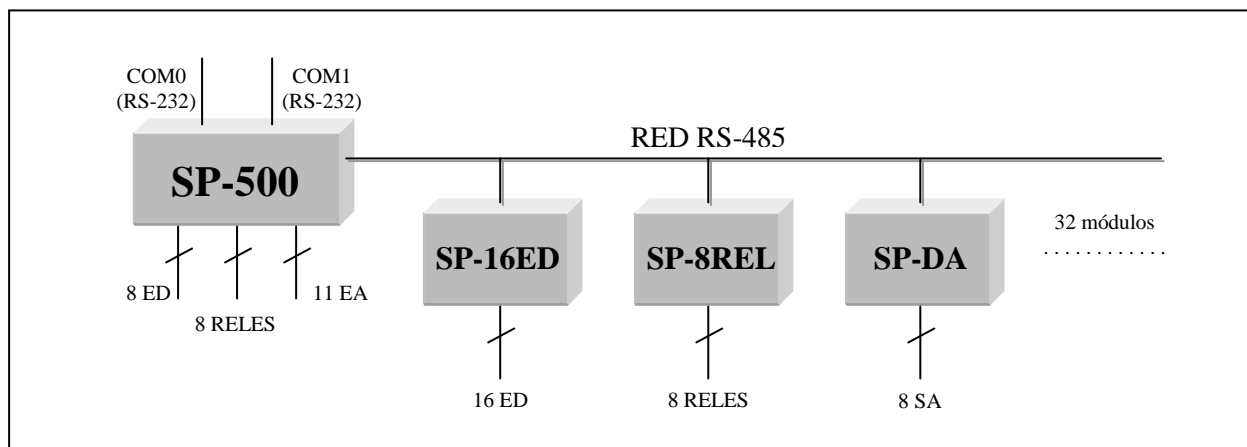


Figura 6.2: Diagrama de bloques del sistema SPIRAL-5000

La unidad central SP-5000, de aproximadamente 10 cm x 6 cm x 7 cm, incluida en un soporte para carril DIN, utiliza un microcontrolador NEC V25 de 16 bits, integrado en el módulo C-Engine de TERN Inc. [www.tern.com, Ter-95], dispone de:

- 8 Entradas Digitales Optoacopladas.
- 8 Salidas de Relés (220 V, 1A).
- 11 Entradas de Conversión A/D (12 bits de resolución).

- 512 Kbytes de memoria RAM.
- Batería de Litio para respaldo de datos almacenados en RAM.
- 512 Kbytes de memoria EPROM.
- 8 Kbytes de memoria EEPROM.
- Sistema de reset gurdían (*Watchdog*) para la supervisión del software.
- Reloj-calendario de tiempo real.
- 2 puertos serie RS-232, y
- 1 puerto serie RS-485 para la comunicación con los módulos de ampliación.

Su consumo máximo es de 100 mA (12V DC) y varía en función de los relés activados. Opcionalmente también se puede alimentar con 24V DC (en ese caso es necesario instalar relés de 24V y colocar un disipador apropiado en el regulador).

La programación del sistema se realiza en PC mediante lenguaje Borland o Microsoft C/C++, pudiendo residir el programa en memoria RAM con baterías o bien en EPROM. El programa se transfiere vía puerto RS-232 a 115 Kbaudios a una distancia máxima de 1200 metros (sin repetidores). Para el desarrollo y depuración de aplicaciones se emplea el paquete de software DEBUG/RT-V25 de Paradigm [Par-94].

Para las aplicaciones donde se requiere un mayor número de entradas y salidas se han desarrollado varios módulos adicionales de ampliación, de muy bajo coste, basados en el microcontrolador PIC16C74 de Microchip Technology [www.microchip.com, Mic-96]. La comunicación con el módulo central se realiza por cable utilizando la norma física RS-485 a 115 Kbaudios. Tras considerar la posibilidad de utilizar algún bus de campo "normalizado" se decidió realizar un protocolo propio, por software, ya que el objetivo no era intercomunicar con elementos de otros fabricantes sino emplear este bus exclusivamente para ampliar la capacidad de E/S del módulo central SP-5000, y todo ello, con distancias máximas de 1200 metros.

El número máximo de módulos de ampliación (nodos del bus) es de 32 a 256 (dependiendo del *transceiver* utilizado), y hasta el momento se han desarrollado los siguientes:

- SP-16ED : Módulo con 16 entradas digitales optoacopladas.
- SP-8REL : Módulo con 8 relés de salida (220V, 1A).
- SP-DA : Módulo con 8 canales de conversión D/A de 8 bits de resolución, y con salida en tensión, 0-10V, o en corriente (0-20 mA/ 4-20 mA).

6.3 Módulo central SP-5000

6.3.1 Consideraciones de diseño.

Tal y como se ha comentado en la descripción general, SP-5000 es el módulo o unidad central del sistema de telecontrol distribuido SPIRAL-5000, basado en el microcontrolador CMOS NEC V25 compatible (repertorio de instrucciones software) con Intel 8086 [Ter-95]. En la Figura 6.3 se representa un diagrama de bloques de este módulo central.

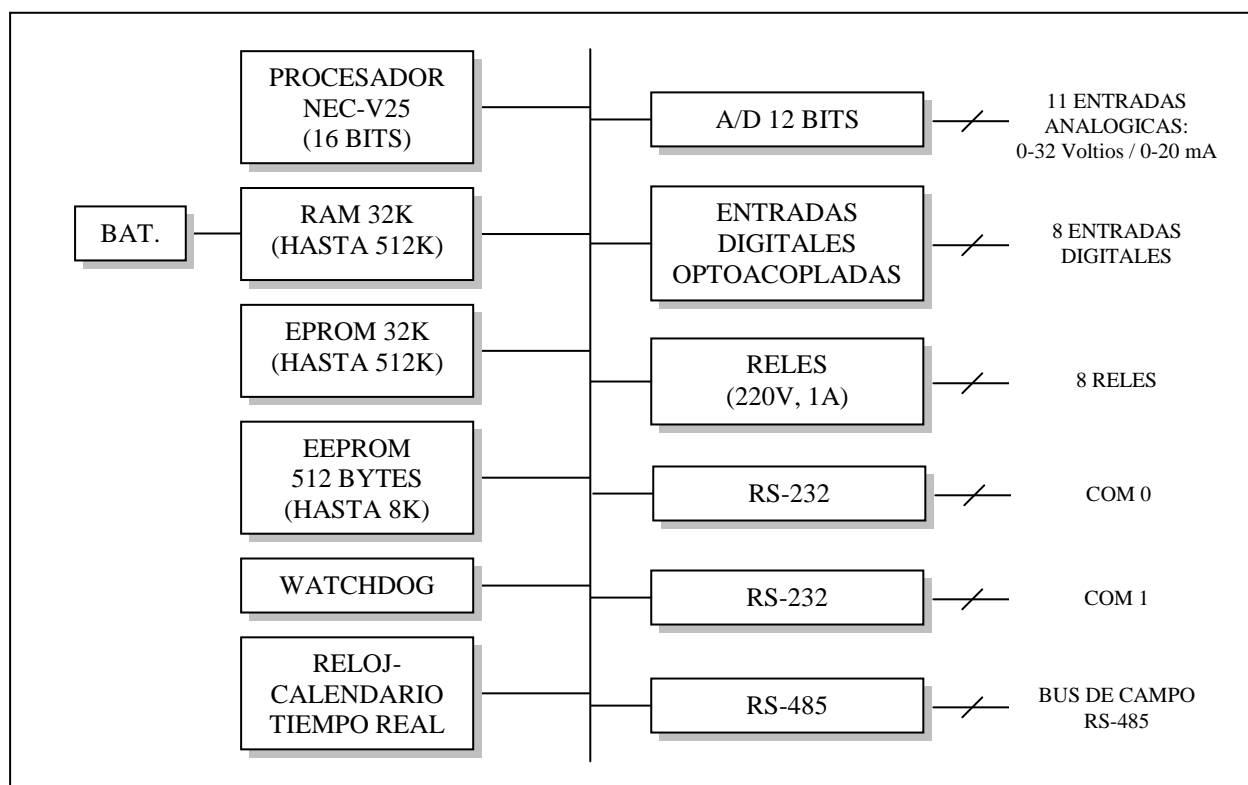


Figura 6.3: Diagrama de bloques del módulo central SP-5000

SP-5000 se ha diseñado de acuerdo a los requisitos expresados en la introducción (sección 6.1), incluyendo además algunas características específicas:

- En los canales A/D se han incluido redes de resistencias para ampliar el rango de la tensión de entrada hasta 30 voltios; colocando los “jumpers” de forma adecuada, la entrada también puede ser en corriente (0-20 mA). Es decir, además de poder conectar sensores con salida 0-20 mA o 4-20 mA, se puede seleccionar el rango de tensión, con niveles de hasta 30 voltios (útil por ejemplo para medir tensiones de baterías de 24 voltios). Además se incluye para cada canal un filtro paso baja (*antialiasing*).

- Los 2 canales serie RS-232 disponen de diodos zener de protección contra sobretensiones.
- La interfaz RS-485 dispone también de diodos zeners de protección. Mediante “jumpers” se pueden conectar resistencias terminadoras en los extremos del bus.
- El módulo incluye conectores normalizados para carril DIN. De esta forma se puede colocar fácilmente en un armario de control industrial junto con el resto de los elementos del mismo.
- Los conectores de tornillo empleados para la interconexión de E/S son del tipo "enchufables", lo que facilita la tarea de mantenimiento.
- SP-5000 permite ser alimentada entre 12 y 24 voltios DC.
- Las entradas digitales están aisladas galvánicamente, pudiendo utilizarse una fuente de alimentación distinta. Hay un sólo común para las 8 entradas. Por otro lado al disponer de optoacopladores el margen de tensión de entrada digital, correspondiente al 1 lógico es mayor (desde 5 voltios a 24 voltios DC).
- En SP-5000 se han incluido 8 microrelés con contactos independientes para cada uno de ellos COM, NA y NC (SPDT). La ventaja de los relés electromecánicos es que también aíslan galvánicamente. En la mayoría de aplicaciones la intensidad máxima admitida por los contactos de los relés no es suficiente para conmutar dispositivos de gran potencia y será necesario añadir externamente los contactores apropiados.

6.3.2 Conexión y configuraciones hardware

Las conexiones exteriores de este módulo vienen rotuladas en la serigrafía del mismo (ver Figura 6.4). A continuación se detallan dichas conexiones y su uso:

a) **Alimentación (J2)**, situada en la parte inferior izquierda de la tarjeta.

La tarjeta SP-5000 se alimenta a través de este conector con tensión continua de 12V (opcionalmente también se puede alimentar a 24V). El positivo de la fuente se conecta a +V y el negativo a uno de los dos contactos GND. El módulo está protegido mediante un diodo en caso de equívoco en la polaridad. El consumo medio para 12 VDC es aproximadamente de 100 mA (este consumo varía en función de los relés activados).

b) **Entradas analógicas (J9)**, ubicadas en la parte inferior de la tarjeta, a continuación de la alimentación, y compuesto por 11 bornas (AI0, ..., AI10).

Cada canal se puede configurar para admitir tensiones (hasta 31.9 voltios) o corrientes (0-20 mA). Esta selección se realiza cerrando (modo corriente) o abriendo (modo tensión) los “jumpers” apropiados en J10 (de 22 terminales), según la asignación siguiente: Canal A/D 0 terminales 1-2 de J10, canal A/D 1 terminales 3-4 de J10, ..., canal A/D 10 terminales 21-22 de J10.

- c) **Entradas digitales (J13)**, situadas en la parte inferior de la tarjeta a continuación de las bornas correspondientes a las entradas analógicas.

Están disponibles 8 entradas digitales (directamente al ánodo del fotodiodo del optoacoplador) y una conexión común de retorno negativo conectada a todos los cátodos. El optoacoplador se ha polarizado de tal forma que conmuta (valor lógico complementado) a partir de 4 voltios, pudiendo admitir niveles máximos de hasta 24 voltios de entrada digital. A cada entrada digital se le ha conectado un diodo LED.

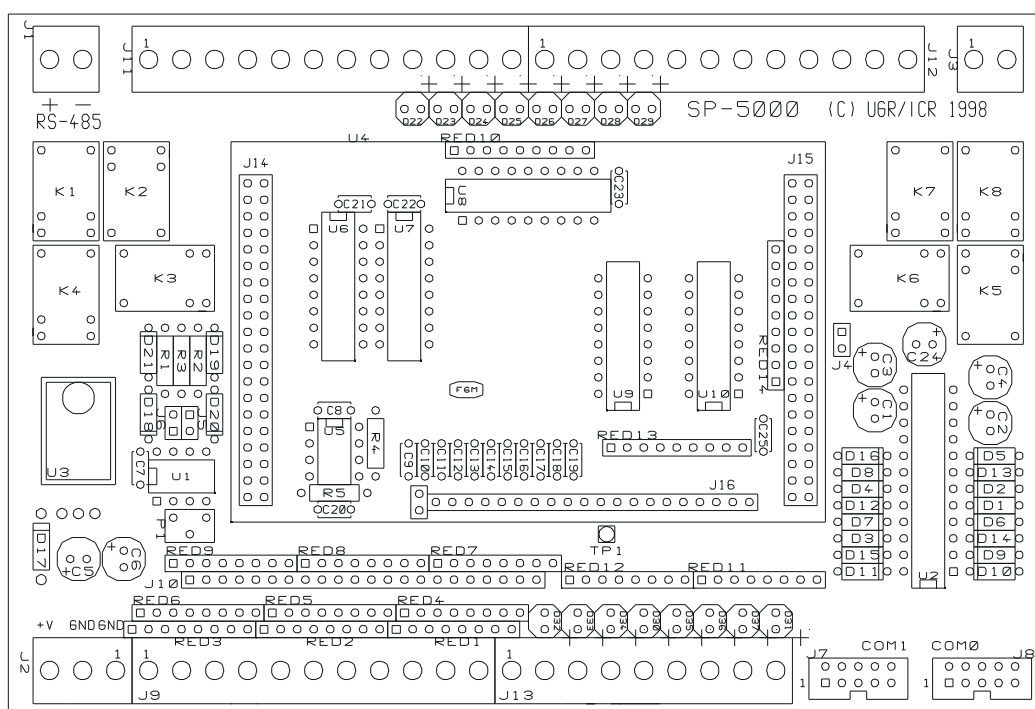


Figura 6.4: Distribución de componentes de SP-5000

- d) **Puertos serie RS-232, COM1 y COM0 (J7,J8)**, situados en la parte inferior derecha. Son conectores machos (2x5) preparados para cable plano, de forma que si al otro extremo del cable se coloca un conector canon DB9 macho el resultado es una configuración RS-232 en modo terminal para los dos canales. Se ha elegido esta opción debido a que normalmente los equipos a conectar son del tipo “modem”.

Para programar la tarjeta es necesario transferir por cualquiera de estos puertos el programa desde un PC hasta el módulo SP-5000. En la Figura 6.5 se detallan los dos tipos de cables a emplear para conectar un modem o para conectar un PC.

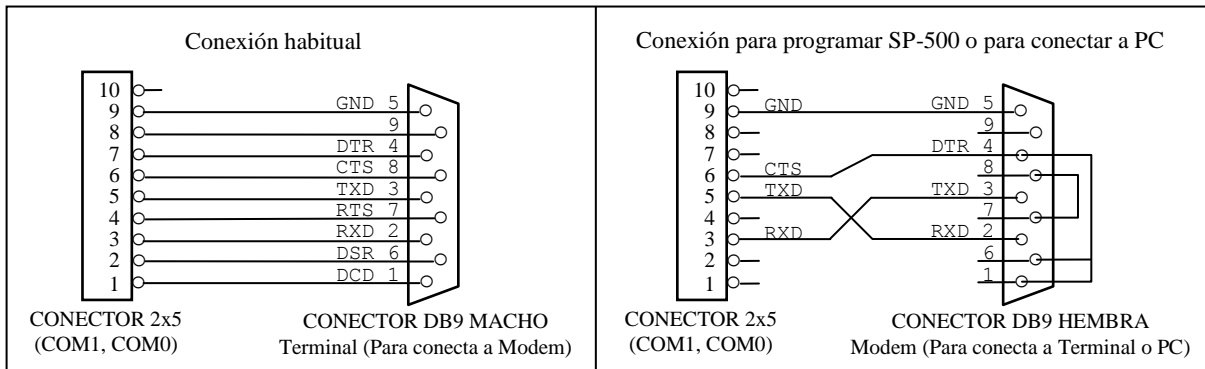


Figura 6.5: Cableado para COM1 y COM0 de SP-5000

- e) **Puerto serie RS-485 (J1)**, ubicado en la parte superior izquierda del módulo. Este conector dispone de 2 bornas (RS485+ y RS485-) para la conexión RS-485 entre SP-5000 y los módulos de ampliación. Opcionalmente también se puede conectar con otros equipos RS-485. Normalmente este puerto serie se configura a 115 Kbaudios para comunicar con los módulos de ampliación ubicados generalmente dentro del mismo cuadro y mediante un protocolo específico. Está prevista también la posibilidad de conectar con dichos módulos (o con otros sistemas) a menos velocidad, lo que es especialmente útil para distancias lejanas no superiores a 1.2 Km. Mediante los “jumpers” J5 y J6 se conectan las resistencias terminadoras necesarias en los extremos de la red. Para añadir dichas resistencias es necesario cerrar estas conexiones.
- f) **Relés (J11, J12)**, situados en la parte superior central. Cada relé dispone de un contacto común (COMx), un contacto normalmente abierto (NAX) y un contacto normalmente cerrado (NCx), de manera que cuando se activa la bobina de un relé se cierra la conexión entre COMx y NAX y viceversa. Estos contactos están preparados para conmutar como máximo 220 V, 1 A. En caso de necesitar conmutaciones de mayor potencia será necesario añadir relés o contactores auxiliares externos. La activación de un relé específico queda indicada con el encendido de su correspondiente LED junto al conector.

6.4 Módulos básicos de ampliación

6.4.1 Consideraciones generales

Al igual que ocurre con SP-5000, todos los módulos se ubican en cajas con carril DIN para ser colocados dentro de cuadros de control, aunque no necesariamente tiene que ser cerca del módulo central. Esta es una ventaja derivada del uso de la norma RS-485, que permite colocar los módulos de forma descentralizada (hasta 1200 metros, dependiendo de la velocidad de transmisión utilizada) mediante sondeo cíclico (*polling*). Esta característica es especialmente útil

en sistemas de control hidráulico dentro de ciudades donde hay canalizaciones disponibles; de esta forma reducimos el número de estaciones conectadas vía radio (ver capítulo 4).

Como ya se ha indicado en apartados anteriores, todos los módulos son de muy bajo coste, se basan en el microcontrolador PIC16C74 y se comunican con SP-5000 con un protocolo de comunicación específico. Cada módulo tiene un código interno fijo, correspondiente al tipo de módulo, y un código externo que se configura con los 6 microinterruptores de S1, concretamente los cuatro primeros. Los módulos que se colocan en los extremos de la red de cable RS-485 deben llevar conectadas las resistencias terminadoras, activando los dos microinterruptores restantes de S1. La velocidad máxima de transmisión es de 115 Kbaudios.

En algunas ocasiones empleamos estos módulos con comunicación *peer-to-peer* utilizando un método de acceso MAC del tipo CSMA/CD persistente con retransmisión con tiempo aleatorio de retroceso exponencial binario (descrito en la sección 6.4.5). Este procedimiento se utiliza para descentralizar entradas/salidas digitales sin necesidad del módulo maestro SP-5000 (de mayor coste), como se puede apreciar en el esquema de la Figura 6.6.

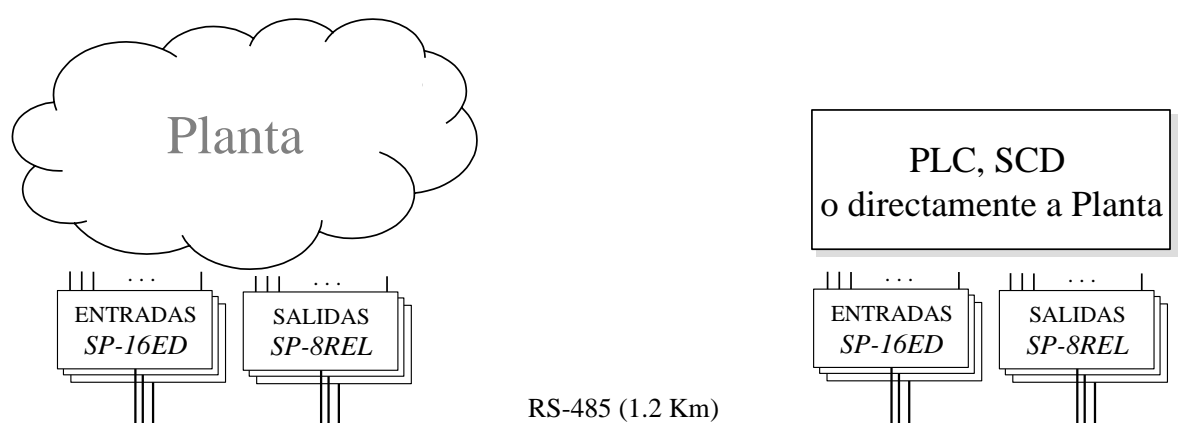


Figura 6.6: Comunicación *peer-to-peer* entre módulos SPIRAL 5000

6.4.2 SP-16ED: 16 Entradas Digitales Optoacopladas

En la Figura 6.7 aparece un diagrama de bloques y la fotografía de este módulo de dimensiones: 9 cm x 10 cm x 7 cm. El microcontrolador incorpora el software que realiza las lecturas de las entradas digitales e interpreta el protocolo de comunicación con SP-5000.

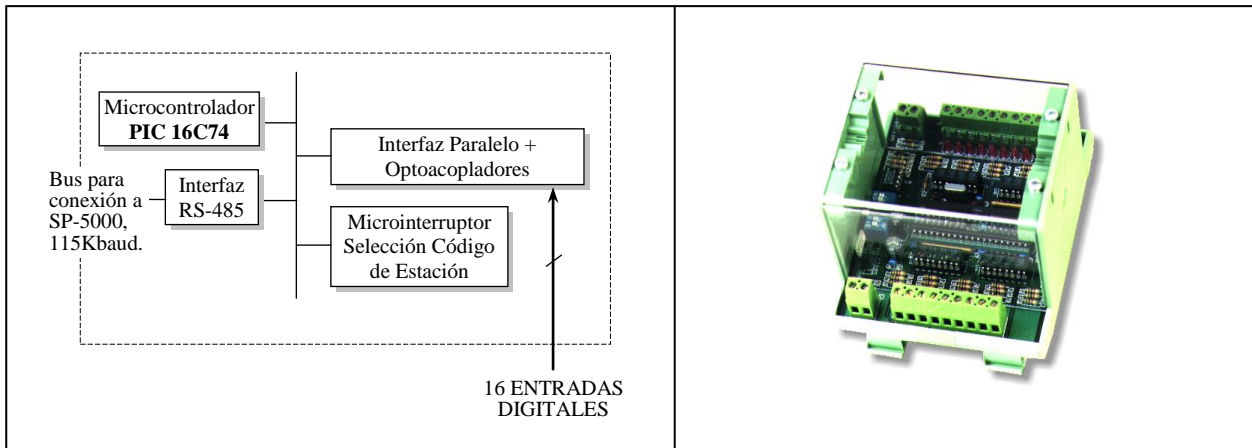


Figura 6.7: Diagrama de bloques y fotografía de SP-16ED

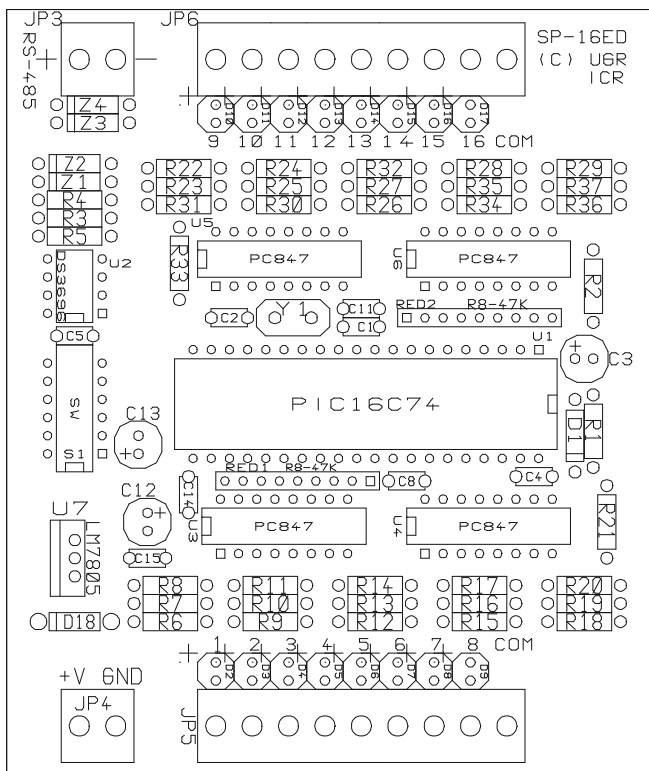


Figura 6.8: Distribución de componentes de SP-16ED

En cuanto al conexionado del módulo (ver Figura 6.8), el conector de alimentación (JP4) y el conector del puerto serie RS-485 (JP3) tienen las mismas características que los equivalentes en el módulo central SP-5000. Respecto a los conectores de las entradas digitales (JP5, JP6), el que se encuentra situado en la parte inferior derecha (JP5) corresponde a las entradas digitales 1-8, mientras que el conector JP6 corresponde a las entradas digitales 9-16.

Las 16 entradas tienen el mismo común, pudiendo utilizarse una fuente de alimentación diferente para aislar realmente. A cada entrada digital se le ha conectado un diodo LED.

6.4.3 SP-8REL: 8 Salidas de Relés

En la Figura 6.9 aparece un diagrama de bloques y la fotografía de este módulo cuyas dimensiones son: 9 cm x 10 cm x 7 cm. El microcontrolador incorpora el software que interpreta el protocolo de comunicación y activa los relés correspondientes según las ordenes recibidas.

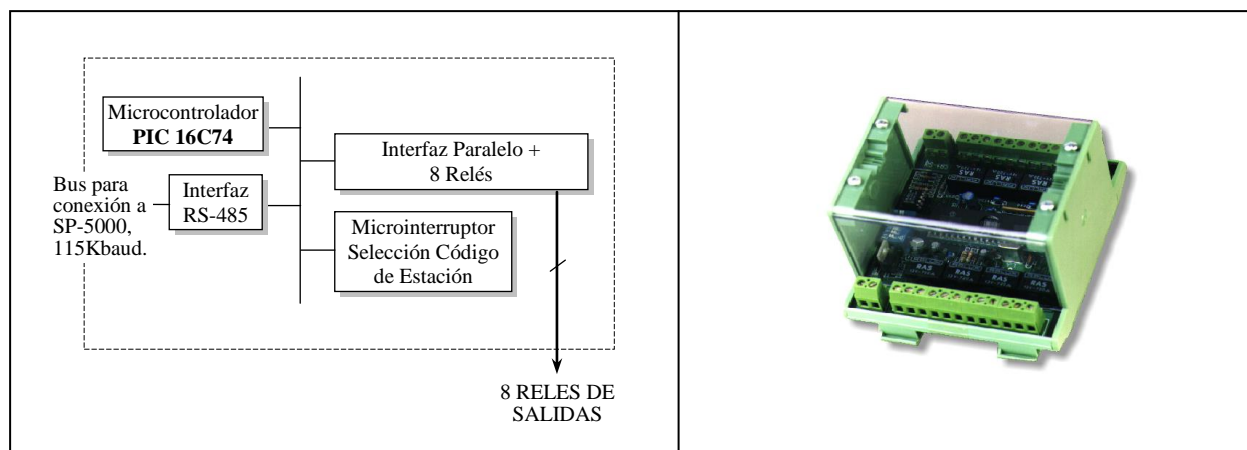


Figura 6.9: Diagrama de bloques y fotografía de SP-8REL

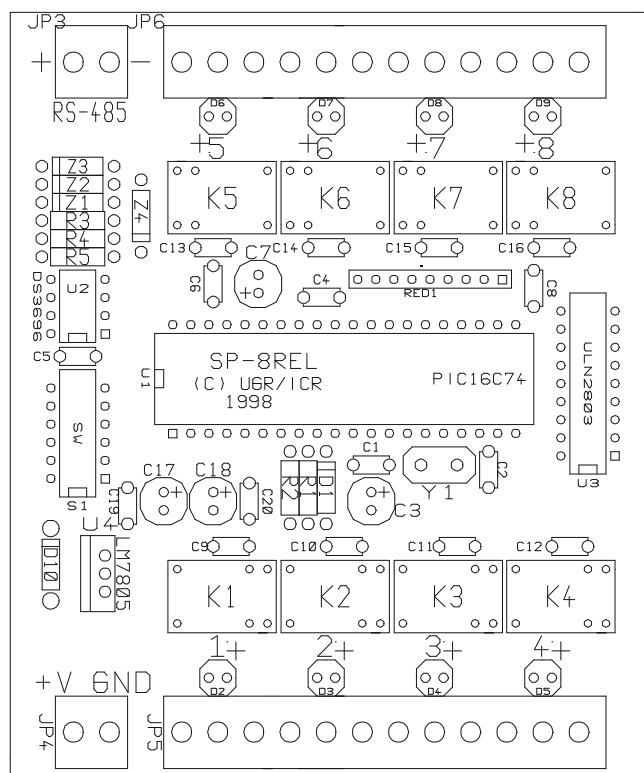


Figura 6.10: Distribución de componentes de SP-8REL

En cuanto al conexionado del módulo (ver Figura 6.10), el conector de alimentación (JP4) y el conector del puerto serie RS-485 (JP3) tienen las mismas características que los equivalentes en el módulo central SP-5000. Respecto a los conectores de los relés (JP5, JP6), el que se encuentra situado en la parte inferior derecha (JP5) corresponde a los relés 1-4, mientras que el conector JP6 corresponde a los relés 5-8.

Cada relé tiene un contacto común (COM), un contacto normalmente abierto (NA) y un contacto normalmente cerrado (NC). Estos relés normalmente conmutan hasta 1A para 220 V.

En caso de que sea necesario conmutar dispositivos de más potencia es necesario añadir contactores auxiliares. La activación de un relé en concreto queda indicada, de la misma forma que en el módulo central SP-5000, con el encendido de un LED ubicado junto al conector correspondiente.

6.4.4 SP-DA: 8 Salidas Analógicas

En la Figura 6.11 aparece un diagrama de bloques y la fotografía de este módulo cuyas dimensiones son: 16 cm x 10 cm x 7 cm. Además del microcontrolador, el interfaz RS-485 y el regulador de tensión, dispone de un convertidor D/A de 8 canales y 8 bits con referencia de tensión precisa y adaptadores de tensión a corriente. Por lo tanto, cada canal analógico de salida puede suministrar tensión (0-10 voltios / 1-10 voltios) y corriente (0-20 mA / 4-20 mA).

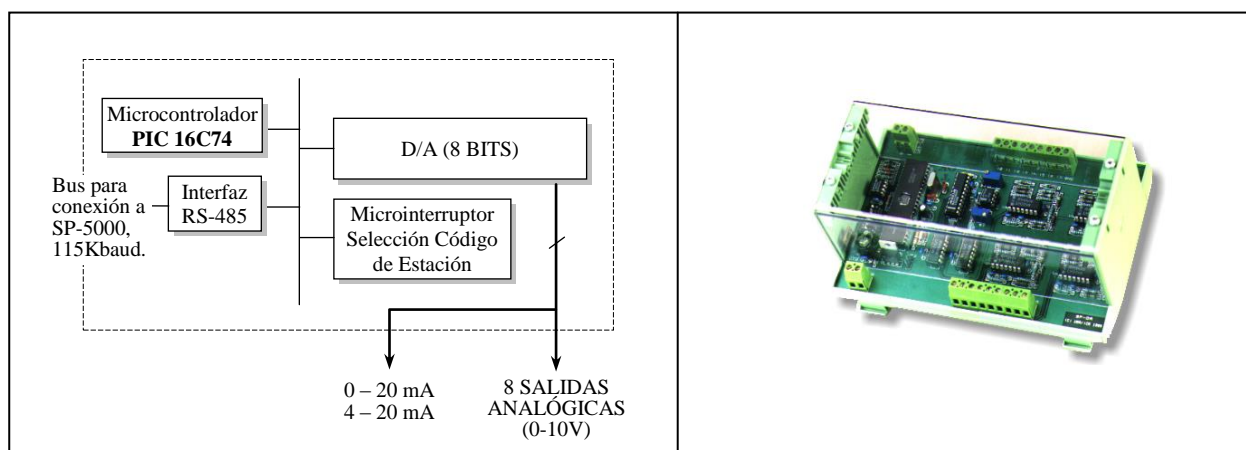


Figura 6.11: Diagrama de bloques y fotografía de SP-DA

Conexión del módulo: En la Figura 6.12 aparece la serigrafía del módulo donde se pueden observar los conectores que se comentan a continuación:

- a) **Alimentación** (JP3), situada en la parte inferior izquierda del módulo. Puede ser a 12V DC o a 24V DC. Para evitar daños en el convertidor D/A, si se desea alimentar a 24V DC es necesario configurar el “jumper” JP2 de la siguiente forma: para 24V DC conectar 2-3 y para 12V DC conectar 1-2.
- b) **Puerto serie RS-485** (JP6), situado en la parte superior izquierda del módulo y con las mismas características que en el resto de los módulos anteriores.
- c) **Salidas analógica en tensión** (JP4), situadas en la parte inferior derecha del módulo. Estas 8 salidas en tensión pueden ser entre 0-10 voltios o bien entre 1-10 voltios. La

selección se realiza con el “jumper” JP8 de la siguiente forma: Para 0-10 voltios conectar 1-2 y para 1-10 voltios conectar 2-3.

- d) **Salidas analógicas en corriente (JP5)**, situadas en la parte superior derecha del módulo. Estas 8 salidas en corriente pueden ser entre 0-20 mA o bien entre 4-20 mA. La selección se realiza también con el “jumper” JP8 de la siguiente forma: Para 0-20 mA conectar 1-2 y para 4-20 mA conectar 2-3. En este modo de corriente la resistencia de carga no debe ser superior a 1K.

Ajustes: Las referencias de tensión del convertidor D/A se ajustan mediante los potenciómetros R6 y R7 de la siguiente forma:

- Colocar la sonda de un voltímetro entre GND y el punto de prueba JP7 y mover el potenciómetro R7 hasta conseguir una lectura de 1.0 voltios.
- A continuación, colocar el voltímetro en JP1 y mover el potenciómetro R6 hasta conseguir una lectura de 10.00 voltios.

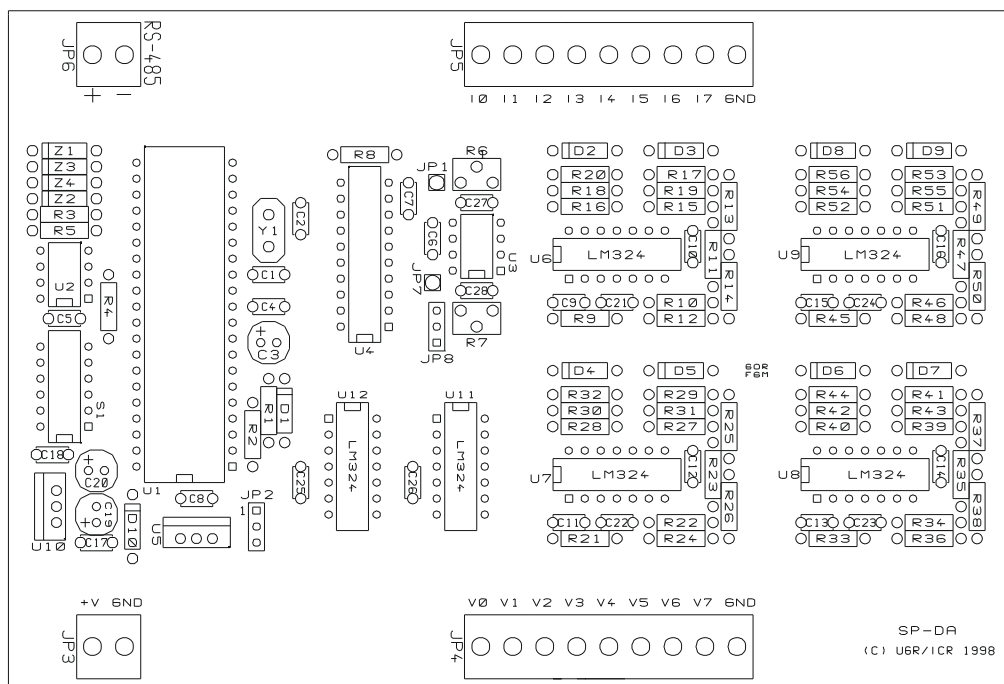


Figura 6.12: Distribución de componentes de SP-DA

Se han diseñado otros módulos para conectar a la SP-5000, como el SP-8E/S (módulo de 8 Entradas y 8 Salidas digitales) o el SP-ADES (similar a la SP-5000 en cuanto a capacidad de entradas y salidas), o incluso un modem telefónico con el mismo formato que los módulos del sistema SP-5000.

6.4.5 Comunicación *peer-to-peer* entre módulos

Como indicamos en la sección 6.4.1, cuando los módulos de ampliación se utilizan sin conexión a la SP-5000, (para reducir costes en ciertas aplicaciones), para acceder al canal se usa un método CSMA/CD persistente con retransmisión en tiempo aleatorio de retroceso exponencial binario [Hal-98].

Cuando se desea transmitir un mensaje, la estación verifica si el canal está ocupado, y si es así, espera a que se libere (CSMA). Al detectar que está desocupado, la estación inicia inmediatamente su transmisión (CSMA persistente), y si no hay ninguna otra estación esperando para transmitir, el paquete se enviará sin problemas.

Si dos o más estaciones desean transmitir y el canal está desocupado, se ocasionará una colisión, las estaciones interrumpirán su comunicación, y esperarán un tiempo aleatorio para iniciar un nuevo intento de transmisión. Con el método utilizado, este intervalo aleatorio de espera obedece a un cierto control denominado "retroceso exponencial binario", consistente en que cada estación espera una cantidad entera y aleatoria de ranuras (tiempo que asegura que todas las estaciones restantes detectan la ocupación del canal) antes de intentar una nueva retransmisión, repitiendo la tentativa hasta que el paquete sea transmitido con éxito. La cantidad de espera es función de la tentativa consecutiva que se está realizando; es decir, si la estación está intentando retransmitir por n -ésima vez, esperará un número aleatorio de ranuras comprendido entre 0 y 2^{n-1} , con un máximo de 256. Si mientras se reintenta una transmisión la estación detecta que alguna otra logra transmitir con éxito (recuperación del canal), se reinicia el procedimiento.

6.5 Software de control de la SP-5000

Se ha implementado una librería (en adelante SPC++), con un conjunto de objetos y API's de programación para el sistema SP-5000, que permite:

- La sencilla programación de las secuencias y procedimientos de control sobre los elementos de E/S del sistema SP-5000.
- Su inclusión en redes SPIRAL [Gom-94] basada en un servidor de comunicaciones, tipo ICeR (descrito en el capítulo anterior).

Para poder realizar una aplicación con esta herramienta es necesario tener instalados los siguientes productos:

1. Paradigm locate v5.0 o posterior, y paradigm debug/rt – V25 v4.0 o posterior [Par-94].
2. Compilador y “Linker” en línea (bcc.exe y tlink.exe) de Borland C++ v3.1 o posterior.
3. SPC++ SDK, que consta de los siguientes ficheros: *makefile*, *serie.cpp*, *tramas.cpp*, *sp5000.cpp*, *serie.h*, *tramas.h*, *sp5000.h*, *p.bat*, *make.bat*, *main.cpp*.

En los siguientes apartados se describen los aspectos más significativos del software SPC++, diseñado especialmente para que el hardware del sistema SP-5000 sea lo más transparente posible al programador de aplicaciones. Además de las funciones que se detallan a continuación, también se han implementado otras utilidades propias de los sistemas de control, como por ejemplo los *temporizadores*, *contadores*, *biestables*, *detectores de flanco*, etc. Actualmente estamos trabajando en la incorporación de lenguajes de la norma IEC-1131-3, en concreto estamos en proceso de validación de los lenguajes IL y ST de dicha norma.

6.5.1 Esquema de funcionamiento

En la Figura 6.13 se puede observar el esquema general del ciclo de operación del sistema SP5000 cuando se utiliza con el software SPC++. Como se puede apreciar utiliza un esquema similar al diagrama descrito para los PLCs (ver capítulo 2).

Después de un proceso inicial de tests del hardware e inicialización de variables, comienza un ciclo de operación secuencial que se repite continuamente. En caso de bloqueo, un temporizador interno reinicia el sistema si alcanza un valor prefijado (*watchdog*).

La lectura y escritura de los módulos de entrada/salida, respectivamente, utilizan un protocolo LLC propio, pensado y simplificado para tipos de aplicaciones como las redes de abastecimiento de agua, con el siguiente formato:

@	Identificador	Tipo trama	Información	Che	CR
---	---------------	------------	-------------	-----	----

Donde:

@	Delimitador de inicio de trama
Identificador	Nº del módulo de expansión
Tipo trama	Indica el tipo de la trama (T - Escritura, R - Lectura, A - Respuesta)
Información	Valor de escritura o de lectura de los datos
Che	Campo para la detección de errores
CR	Delimitador de final de trama

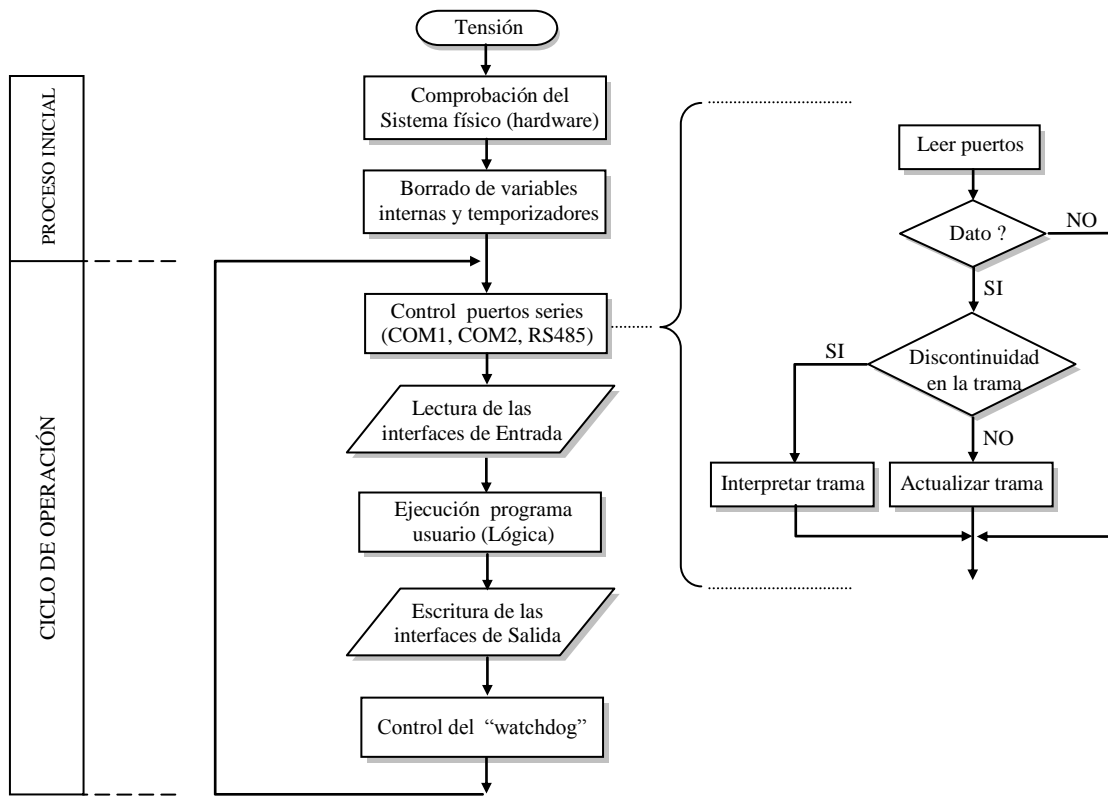


Figura 6.13: Ciclo de funcionamiento del sistema SP5000

6.5.2 La clase SP5000

La manipulación del sistema SP-5000 se realiza mediante un objeto de la clase SP5000, que realiza las tareas de lectura/escritura de los periféricos, control de las comunicaciones, ejecución del programa de usuario, etc.

El objeto de la clase SP5000 contiene la información global del sistema incluyendo los datos en tiempo real de los módulos de E/S, a los que se puede acceder mediante unos manipuladores propios del objeto denominado Ttarjeta (el uso de estos manipuladores se detalla en la sección 6.5.6)

Además de los datos de E/S existen una serie de variables públicas de la clase con información complementaria como: Número asignado a la estación (NESTACION), fecha y hora en que se realizó el último acceso a la estación desde la red (FECHA_ULTIMA_AC y HORA_ULTIMA_ACT), un vector de punteros a DB's (su uso y utilidad se detallan en la sección 6.5.3), etc.

Los métodos públicos de la clase SP5000 que pertenecen al API y que se detallan a continuación se invocan anteponiendo el nombre del objeto y un punto; por ejemplo:

```
SP5000 a;
a.api1(arg1,arg2);
```

<u>Inicialización</u>	<p>Sintáxis: <i>void Inicializacion (int estacion, int wd)</i> Esta función inicializa el objeto y es preciso llamarla antes de hacer uso de cualquier otra API del sistema.</p> <p>Parámetros: <i>estacion</i> es el número de estación para la red spiral. <i>wd</i> si vale 1 activa el “watchdog” de la tarjeta.</p>
<u>IniciaRTC</u>	<p>Sintáxis: <i>void IniciaRTC (unsigned char *time_now)</i> Esta función inicializa el reloj en tiempo real del sistema.</p> <p>Parámetros: Cadena de caracteres con el siguiente formato: <i>time_now[0]</i> = Día de la semana (0 lunes, ..., 6 domingo) <i>time_now[1]</i> = Decenas de números de año <i>time_now[2]</i> = Unidades de números de año <i>time_now[3]</i> = Decenas de números de mes <i>time_now[4]</i> = Unidades de números de mes <i>time_now[5]</i> = Decenas de números de día del mes <i>time_now[6]</i> = Unidades de números de día del mes <i>time_now[7]</i> = Decenas de hora <i>time_now[8]</i> = Unidades de hora <i>time_now[9]</i> = Decenas de minuto <i>time_now[10]</i> = Unidades de minuto <i>time_now[11]</i> = Decenas de segundo <i>time_now[12]</i> = Unidades de segundo</p>
<u>DefineTarjeta</u>	<p>Sintáxis: <i>void DefineTarjeta (int id)</i> Antes de poder hacer uso de un determinado elemento de E/S periférico, salvo los de la unidad central (números 1 para las SD, 2 para las ED y 4 para las EA), hay que definir el módulo en SP5000.</p> <p>Parámetros: <i>id</i> determina el número y el tipo de periférico: 0x10 – 0x1F → módulos de salidas digitales (SD). 0x20 – 0x2F → módulos de entradas digitales (ED). 0x30 – 0x3F → módulos de salidas analógicas (SA). 0x40 – 0x4F → módulos de entradas analógicas (EA).</p> <p>Retorno: Devuelve 0 si hay error.</p>
<u>Tarjeta</u>	<p>Sintáxis: <i>Ttarjeta *Tarjeta (int idx)</i> Esta función recupera un puntero a un objeto manipulador de un determinado módulo de E/S.</p> <p>Parámetros: <i>idx</i> es el número de módulo de entrada/salida: 1 → salidas digitales de la unidad central. 2 → entradas digitales de la unidad central. 4 → entradas analógicas de la unidad central. [0x10 – 0x4F] → como se ha descrito antes en <i>IniciaTarjeta</i>.</p> <p>Retorno: Puntero a un objeto manipulador de tipo <i>Ttarjeta</i>.</p>
<u>Run</u>	<p>Sintáxis: <i>void Run (void (*Logica)(void))</i> Esta función itera indefinidamente realizando funciones internas y de comunicaciones además de la función especificada como parámetros.</p> <p>Parámetros: <i>Logica</i> es la función definida por el usuario.</p>

6.5.3 La organización de la memoria

La zona de datos accesible desde el exterior mediante el protocolo Spiral consta de 10 vectores (de 0 a 9) de 128 palabras de 16 bits (*T16Bits*), cada uno de ellos llamado *DB* (bloque de datos). Esta zona de datos es accesible directamente como variable del objeto. Por ejemplo:

```
SP5000 a;
unsigned int b;

a.db[0][0] = 5;           // se asigna el valor 5 a la palabra 0 del DB 0
a.db[3][1].bit4 = 1;     // se asigna un 1 al bit 4 de la palabra 1 del DB 3
b = (unsigned int) a.db[0][0]; // se asigna a la variable b el contenido de la palabra 0 del DB 0
```

El *DB0* se crea al realizar la inicialización del objeto, mientras que los demás *DB's* deben crearse y eliminarse haciendo uso de las siguientes API específicas: *CreaDB*, *DestruyeDB* y *RecreaDB* (redefine el tamaño de un *DB* previamente creado).

6.5.4 Gestión de comunicaciones

Para cubrir las necesidades de comunicación de la estación SP-5000 en una red Spiral utilizando una interfaz de comunicaciones telefónica o de radio, se han implementado las siguientes funciones:

<u>IniciaPuerto</u>	<p>Sintáxis: <i>void IniciaPuerto (int port, int velocidad, int datos, int stop)</i> Inicia un puerto serie RS-232.</p> <p>Parámetros: <i>port</i> es el número de puerto serie (0 o 1). <i>velocidad</i> velocidad del puerto serie en baudios (V1200, V2400, ..., V115200). <i>paridad</i> (0=sin paridad, 1=par, 2=impar). <i>datos</i> (7=7bits, 8=8 bits). <i>stop</i> (1=1 bit de stop, 2=2 bits de stop).</p> <p>Retorno: Nada.</p>
<u>IniciaModem</u>	<p>Sintáxis: <i>void IniciaModem (int port)</i> Asigna a un puerto serie RS-232 previamente inicializado un modem telefónico.</p> <p>Parámetros: <i>port</i> es el número de puerto serie (0 o 1).</p> <p>Retorno: Nada.</p>
<u>LanzarTrama</u>	<p>Sintáxis: <i>void LanzarTrama (char *trama, int port, int destino, int fuente, char *phone)</i> Envía una trama de comunicaciones hacia un puerto serie RS-232.</p> <p>Parámetros: <i>trama</i> es la trama de comunicaciones construida con una función llamada <i>CreaTramaXXXX</i>. <i>port</i> es el número de puerto serie (0 o 1). <i>destino</i> es el número Spiral de la estación de destino de la comunicación.</p>

fuelle es el número Spiral de la estación que envía la comunicación (este número se le asigna a un objeto SP5000 en la inicialización y se almacena en la variable de clase *nestacion* de acceso público).

phone es el número de teléfono a especificar si el puerto serie tiene asignado un modem telefónico, si no este parámetro debe ser NULL.

Retorno: Devuelve 1 si la trama ha sido enviada, y 0 si el puerto está ocupado y no puede enviarla.

CreaTramaE... Sintáxis: *void CreaTramaEscritura (char *trama, int idb, int palini, int npal, int ibit, T16Bits **db)*

Crea una trama de escritura.

Parámetros: *trama* buffer donde se almacena la trama. Dicho buffer debe haber sido creado con anterioridad.

idb índice del vector de punteros a DB's *db*.

palini palabra inicial a escribir.

npal el número de palabras a escribir.

ibit indica el número de bit de 0 a 15 a modificar de la palabra *palini*, en este caso se asumirá *npal=1* cualquiera que sea su valor. Si se desea realizar una escritura de palabras este valor debe ser NO_BIT.

Retorno: Devuelve la trama.

CreaTramaL... Sintáxis: *void CreaTramaLectura (char *trama, int idb, int palini, int npal)*

Crea una trama de lectura.

Los parámetros son equivalentes a los de la función *CreaTramaEscritura*.

6.5.5 Manipulación de los elementos de E/S

Los manipuladores de los módulos de entrada/salida tanto analógicos como digitales son objetos *Ttarjeta*, y los métodos públicos de esta clase que pertenecen al API son los siguientes:

GetTipo Sintáxis: *int GetTipo ()*
Devuelve el tipo de elemento de E/S al que está asociado el manipulador.
Parámetros: Ninguno.
Retorno: Uno de los siguientes valore: ED, SD, EA, SA.

SetBit Sintáxis: *void SetBit (unsigned int value, int nbit)*
Aplicada sobre un elemento de salidas digitales pone la salida *nbit* al valor indicado por *value*. Para los demás elementos (de entrada o analógicos) esta función no hace nada.
Parámetros: *value* es el valor de la salida digital.
nbit es el número de salida digital.
Retorno: Nada.

SetWord Sintáxis: *void SetWord (unsigned int value, int n=0)*
Pone el elemento *n* al valor indicado por *value*.
Aplicada sobre los elementos de salidas digitales actúa sobre todas ellas asignándoles a cada una un valor según la correspondencia de *value*.
Sobre los elementos de salidas analógicas *n* especifica el número de salida. Esta función no hace nada aplicada a los elementos de entrada.
Parámetros: *value* es el valor de la salida.

		<i>n</i> es un parámetro opcional que puede ser obviado, en cuyo caso se considera con valor 0.
	Retorno:	Nada.
<u>GetBit</u>	Sintáxis:	<i>unsigned int GetBit (int nbit)</i>
		Recupera el valor de la entrada o salida digital <i>nbit</i> . Para los demás elementos (de entrada o salida analógica) esta función no hace nada.
	Parámetros:	<i>nbit</i> es el número de la E/S digital.
	Retorno:	El valor de la E/S digital.
<u>GetWord</u>	Sintáxis:	<i>unsigned int GetWord (int n=0)</i>
		Aplicada sobre los elementos de E/S digitales recupera una palabra con el estado de todas ellas. Sobre los elementos de E/S recupera el valor de la E/S <i>n</i> .
	Parámetros:	<i>n</i> parámetro opcional.
	Retorno:	Una palabra con el valor de una E/S analógica o con el valor de todas las E/S digitales (según el manejador sobre el que se aplique).

También hemos incorporado funciones de error como *GetError* o *ResetError* que devuelve o resetea el estado de error de un elemento respectivamente.

6.5.6 Los manipuladores de palabras

En la sección 6.5.3 (la organización de la memoria) se introdujeron los *DB's* como zonas lineales de palabras. Para facilitar el acceso a nivel de bits y de byte estas palabras no se definen sobre un tipo básico como podría ser *unsigned int*, sino que se definen en base a manipuladores de palabras *T16bits*.

Se permite el acceso a un bit individual añadiendo al nombre del objeto un punto y la palabra *bitX*, donde *X* es el número de bit al que queremos acceder de 0 a 15; por ejemplo:

```
T16Bits a;
a.bit1 = 1;    // a vale 2
a.bit0 = 1;    // a vale 3
```

Y los métodos asociados a estos objetos son los siguientes:

- ”=” asigna un valor de tipo *unsigned int* a un objeto *T16Bits*.
- “unsigned int” casting sobre el objeto *T16Bits*.
- “low” recupera el byte bajo de la palabra.
- “high” recupera el byte alto de la palabra.

6.5.7 Programación de aplicaciones

A la hora de implementar una aplicación completa utilizando los módulo SP-5000 como sistema de control distribuido se ha de construir un proyecto e incluir en él los ficheros *serie.cpp*, *tramas.cpp* y *sp5000.cpp*.

El software de aplicación debe escribirse en el módulo *main.cpp*.

En el fichero *makefile* existen una serie de opciones que se han de configurar según las necesidades de cada proyecto. Fundamentalmente hay cuatro parámetros:

1. El uso de números reales:
FLOAT = 0 # 0 – none, 2 – emulator
2. La cantidad de memoria RAM de la SP-5000:
SRAM = 1 # 0 – 32k, 1 – 128k, 2 – 256k, 3 – 512k
3. El tamaño de la EPROM:
EPROM = 1 # 0 – 32k, 1 – 64k
4. El tamaño del STACK. Para aplicaciones medianas y pequeñas basta con 4 Kbytes aprox. Aunque podemos especificar hasta 20 Kbytes:
STACK = 4096

Hay dos fases a la hora de desarrollar una aplicación con SPC++ SDK:

1. **DEPURACIÓN:** se ha de tener conectado el ordenador al puerto serie 0 de la SP-5000 mediante un cable serie. En el fichero *makefile* se ha de especificar que el código objeto va a residir en RAM.

```
DEBUG = 1      # 0 – none, 1 – TDREM, 2 – PDREMOTE
```

Una vez compilado el proyecto (*make.bat*) se tiene que transferir el programa a la SP-5000 para ejecutarlo (*p.bat*).

2. **INSTALACIÓN:** se debe generar un fichero con extensión *.bin*, que es el que se tiene que grabar en la EPROM. Para ello se ha de especificar además el tamaño de dicha EPROM.

```
DEBUG = 0      # 0 – none, 1 – TDREM, 2 – PDREMOTE
EPROM = 1      # 0 – 32K, 1 – 64K
```

A continuación, como ejemplo, se describe una aplicación para replicar los estados de un módulo periférico de 16 entradas digitales sobre dos módulos periféricos de 8 salidas digitales, haciendo uso de la herramienta descrita (SPC++):

```
# include <stdio.h>
# include <stdlib.h>
# include <dos.h>
# include "serie.h"
```

```

#include "tramas.h"
#include "sp5000.h"

// definición de variables globales
SP5000 sp;
Extern struct puerto p[3];

// prototipo de funciones
void Logica();
void Inicializacion();
void main(void);

// implementación de funciones
void Inicializacion(void)
{
    ve_init(); // se inicializa el hardware
    led(0); // se apaga el LED
    sp.Inicializacion(1, 0); // se inicializa el kernel SPC++, se asigna el número de estación
    // spiral 1, y se desactiva el "watchdog"
    sp.DefineTarjeta(0x10); // se definen los módulo periféricos
    sp.DefineTarjeta(0x11);
    sp.DefineTarjeta(0x20);
}

void Logica(void)
{
    T16Bits value;
    /* Se asignan las entradas 0 – 7 de la tarjeta 0x20 a las salidas 0 – 7 de la tarjeta 0x10 y las entradas
    8 – 15 de la tarjeta 0x20 a las salidas 0 – 7 de la tarjeta 0x11 */
    value = sp.Tarjeta(20)->GetWord();
    sp.Tarjeta(10)->SetWord(low(value));
    sp.Tarjeta(11)->SetWord(high(value));
}

void main(void)
{
    Inicializacion(); // inicializaciones
    sp.Run(Logica); // entramos en el bucle SPC++
}

```

Esta es pues la estructura general que ha de tener todo programa que utilice las API's que se han desarrollado para facilitar la programación del sistema SP-5000. Se puede observar, que la función **Logica(void)** (insertada en el cuadro) es la que ha de modificar el usuario para cada aplicación.

Un ejemplo un poco más completo se puede ver en el capítulo 4, donde se comenta la utilización del sistema SP-5000 en la sectorización de la ciudad de Granada. En concreto, en el apartado 4.4.2.2, se observa cómo se utiliza este software para implementar la adquisición de datos en cada uno de los sectores equipados con una tarjeta SP-5000.

6.6 Conclusiones

En este capítulo se ha tratado la concepción y el diseño de un sistema de control distribuido, basado en microcontrolador, que se ha utilizado como sistema de adquisición y control en la sectorización de la ciudad de Granada, y como una alternativa que se adapta mejor a las especificaciones iniciales de esta instalación que los tradicionales PLC, en cuanto a canales de comunicaciones, lenguaje de programación, capacidad de memoria, bajo consumo, dimensiones reducidas y periferia (entradas/salidas, módulos en red).

El sistema Spiral-5000 es un producto totalmente terminado y operativo, modular, fácil de programar y configurar, y con un precio competitivo con el resto de sistemas para control distribuido existentes.

Se trata por tanto de otro sistema concebido y diseñado en la Universidad de Granada, que se está actualmente utilizando como producto comercial por parte de la empresa ICR S.A., al igual que el gestor de comunicaciones distribuidas descrito en el capítulo anterior. Concretamente se ha utilizado en diversos proyectos, como por ejemplo:

- Sistema de telecontrol y supervisión de faros marítimos en El Ferrol (La Coruña), en el Distrito de Shanghai (China) y en Cap-Bengut (Argelia).
- Procesador de comunicaciones en sistemas de riego [Dam-00a] (Jumilla, Huelva, etc.).
- Sistema de telecontrol de estaciones meteorológicas para el AMA (Agencia del Medio Ambiente en Granada).

Capítulo 7:

Diseño de software para control y supervisión

*La función principal que caracteriza la oferta actual de software de supervisión y control es la consecución de un entorno abierto unificado que integre aplicaciones con conectividad a través de estándares internacionalmente reconocidos [Ayz-99]. En el Capítulo 2, donde se tratan aspectos generales sobre la integración de procesos, se ha proporcionado una introducción general sobre los sistemas SCADA. Recordando, el acrónimo SCADA corresponde a *Supervisión, Control And Data Acquisition*, y se refiere precisamente al conjunto de elementos cuyo fin es la adquisición de datos y la supervisión de operaciones. Se hizo también la distinción entre *software SCADA* (paquete de software que soporta la aplicación y permite su diseño) y *aplicación SCADA* (lo que el usuario del sistema percibe y utiliza).*

En este capítulo se tratan aspectos de diseño de aplicaciones SCADA y la metodología seguida en el caso concreto de la aplicación para la gestión del ciclo integral del agua de Granada. Después de una breve introducción, en la sección 7.2 se describen algunos detalles sobre el software de desarrollo utilizado, para en las secciones 7.3 y 7.4, y usando dicho software, mostrar como se ha de diseñar una aplicación para que cumpla una serie de objetivos, resumidos en las conclusiones finales (sección 7.5).

7.1 Introducción

Las redes de comunicación de datos entre diversos equipos siguen la tendencia común de una estructura piramidal (concepto CIM, Capítulo 2), en cuya base se encuentran los sistemas de regulación más básicos, con redes de comunicación de bajo nivel de complejidad (buses de campo), y en las zonas altas de la estructura se encuentran los sistemas superiores del sistema SCADA, es decir, los ordenadores en los que reside y ejecutan las aplicaciones de supervisión y control.

La interfaz Hombre-Máquina (*Man Machine Interface, MMI*) es la parte visible del sistema SCADA, y su diseño es importante no sólo desde un punto de vista gráfico, sino también para ayudar a mejorar la operatividad en el uso de las aplicaciones desarrolladas con esta herramienta.

En un sistema para la gestión integral de un abastecimiento de agua, el sistema SCADA, parte fundamental en el control operacional, se compone de:

- *Estaciones remotas*, instaladas en puntos estratégicos de la red, como ETAP's, EDAR's, estaciones de bombeo, depósitos, estructuras de control de flujo y puntos críticos de algunas tuberías. Cada terminal remoto se compone, básicamente, de un sistema basado en microprocesador con entradas analógicas y digitales e interfaces de salida conectados a los instrumentos locales y a los actuadores. La unidad remota permite la adquisición de datos y su transmisión al ordenador central y viceversa; es decir, también recibe señales del ordenador central y las transforma en acciones sobre el dispositivo de control.
- *Computador central*, que recibe las señales transmitidas por las estaciones remotas y las procesa, proporcionando información gráfica y/o numérica a analizar por el operador, y un plan de operación que tendrá que ser transmitido a las estaciones remotas para que dirijan la ejecución de las maniobras que ya han sido definidas por el programa del ordenador central.

En cuanto al nivel de automatización que se quiera implantar, estará relacionado con la decisión de hasta qué punto el sistema SCADA debe controlar ciertas operaciones más o menos usuales y complejas. Por lo general, cuanto mayor nivel de automatización se requiere mayor número de señales se van a necesitar en el puesto de control central.

En el Capítulo 4 ya se ha descrito el sistema de control para la red de abastecimiento de agua de Granada desde el punto de vista de los equipos físicos, la topología y la red de comunicaciones, y en el Capítulo 2 se han descrito las características fundamentales que ha de

tener todo software SCADA. Por tanto, sólo queda mostrar como se ha diseñado dicha aplicación utilizando como software principal el SCADA de Wonderware y el gestor de comunicaciones distribuidas ICeR (ver Capítulo 5).

7.2 Consideraciones generales sobre el SCADA utilizado

FactorySuite® 2000 de Wonderware [www.wonderware.com] es un sistema MMI integrado y basado en componentes [Won-99a]. Esta compañía fue pionera (1989) en el uso de Windows en el área del control de procesos de fabricación. FactorySuite 2000 funciona bajo el sistema operativo Windows NT/9X de Microsoft®.

Actualmente, no basta con instalar solamente una base de datos, o un MMI, sino que se necesita un sistema que integre todos los componentes (visualización, control de procesos, adquisición de datos de la planta, almacenamiento y análisis de esos datos, acceso vía Internet/Intranet, etc.) para así conseguir mayor productividad.

Wonderware ha adoptado tecnologías de vanguardia como DDE, NetDDE, OLE, ActiveX, DCOM y Java; y también ha creado algunas propias [Won-99b]. Concretamente NetDDE fue desarrollado y luego licenciado a Microsoft, en donde pronto se convirtió en un estándar para la industria. Wonderware participa activamente en los comités involucrados en la creación de nuevos estándares tales como OPC, Profibus, IEC-1131 y los estándares de proceso por lotes S88. Actualmente todos los componentes de FactorySuite 2000 se basan en estos estándares.

InTouch, la herramienta de visualización de FactorySuite 2000 utilizada, incluye, entre otras, las siguientes características:

- Programación orientada a objetos.
- Lenguaje de programación simple, versátil y cómodo en cuanto a su depuración, y con posibilidad de incorporar funciones a medida desarrolladas con lenguajes de programación como C++, Delphi, Visual Basic, etc.
- Programación asociada a condiciones o eventos. El código se ejecuta cuando se produce algún evento, como cambios en los valores de las variables, actuaciones del operador, condiciones lógicas, etc.
- Editor gráfico con amplias posibilidades de animación.
- Sumario e históricos de alarmas totalmente configurables.

- Comunicación con tareas concurrentes mediante DDE y/o OPC.
- Base de datos en tiempo real.
- Sondeo optimizado. Sólo las variables necesarias se transmiten a y desde los dispositivos de entrada/salida al servidor, lo que reduce la carga asociada a la adquisición de datos.
- Mecanismos implementados de seguridad que permiten dar acceso a parte del sistema y ocultar información dependiendo del nivel de acceso de cada operador.
- Registros históricos y en tiempo real, que permiten configurar diferentes tipos de gráficas.
- Almacenamiento indefinido de información histórica recuperable para su consulta.
- Configuración *on-line*; es decir, no hace falta detener la ejecución para hacer un cambio en la aplicación.

La siguiente sección está dedicada a describir la metodología adoptada en cuanto al modo y nivel de programación, diseño de las pantallas (ubicación de elementos, colores elegidos, animaciones, y representación de sistemas), procedimientos de operación y acceso, y, sobre todo, al desarrollo de algunos módulos imprescindibles para el trabajo rutinario de los operadores de las salas de control. Esta metodología nos ha permitido: implementar aplicaciones complejas pero a la vez fáciles de usar, y generar una serie de módulos comunes que forman una base reutilizable que facilita y acelera el desarrollo de nuevas aplicaciones.

7.3 Diseño de una aplicación SCADA. Caso de la gestión del agua en la ciudad de Granada.

A continuación se describe el uso de esta herramienta y la metodología que se ha propuesto para el diseño de redes de abastecimiento de agua, utilizando como ejemplo la gestión del agua en la ciudad de Granada.

7.3.1 La programación dirigida a eventos

Para realizar las acciones necesarias de la aplicación el software de base utilizado emplea una programación dirigida a eventos, estructurada en diferentes tipos de lógica: lógica asociada a toda la aplicación (*Application Scripts*), lógica asociada a condiciones (*Condition Scripts*), lógica

asociada a cambios en los datos (*Data Change Scripts*), lógica asociada a las ventanas (*Windows Scripts*), lógica asociada a pulsaciones de teclas (*Key Scripts*, y *Touch Links*) y lógica asociada a pulsaciones del dispositivo apuntador, normalmente de tipo “ratón” (*Touch Links: user inputs, sliders, pushbuttons*).

En la Figura 7.1 se puede observar el diagrama de estado del código dirigido a eventos que corresponde al módulo de ejecución *Run-Time* del SCADA InTouch.

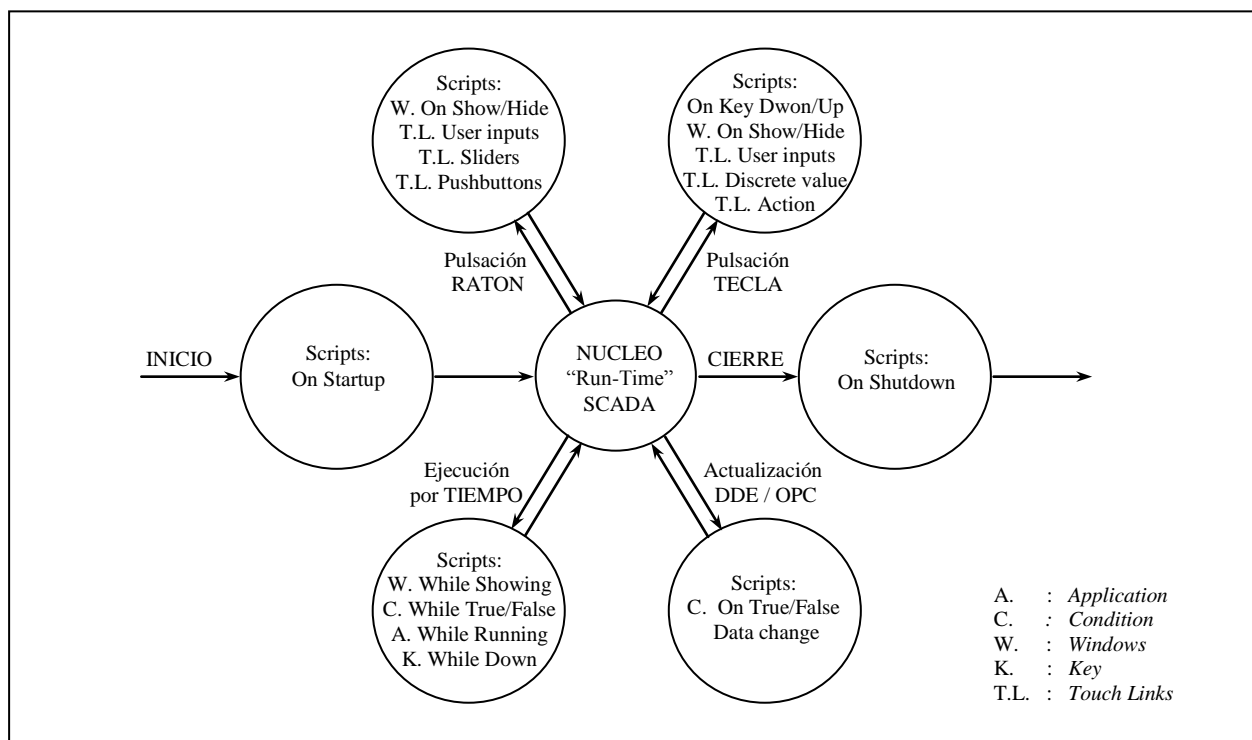


Figura 7.1: Diagrama de estados de la ejecución por eventos en el SCADA InTouch

Teniendo en cuenta esta forma de estructurar la programación, en la aplicación para el abastecimiento de agua de Granada se ha hecho la asignación de código que se muestra en la tabla de la Figura 7.2.

Scripts:		Acciones o funciones más importantes que se realizan:
<i>Application</i>	<i>On Startup</i>	Asignación de <i>path</i> para ejecutar programas externos, control conversaciones DDE para los distintos puestos de operador, inicializaciones en general.
	<i>While running</i>	Control espacio disco duro, cálculos varios (capacidad total en depósitos, caudal total distribución, caudal total salida sectorización)
	<i>On Shutdown</i>	Nada.
<i>Condition</i>	<i>On True</i> <i>While True/False</i> <i>On False</i>	A ciertas horas del día: Guarda el estado de variables (para mostrar en informes), puesta en hora y/o realizar <i>reset</i> a ciertas estaciones, cálculos varios para informes, almacena variables en ficheros, imprimir informes de forma automática, etc. Cada cierto tiempo: establece niveles máximos y mínimos según tramos horarios para los bombeos, genera alarmas para el sistema Spiral GSM 9000, actualiza variables en la ETAP, comunica variables entre estaciones, etc. Calculo del caudal impulsado por los bombeos, informa a las estaciones remotas del estado de la central, comprueba el estado de los autómatas de las estaciones remotas, calculo estadística de fallos de comunicaciones, <i>reset</i> de eventos, visualiza ventanas de avisos, etc.
<i>Window</i>	<i>On Show</i> <i>While Showing</i> <i>On Hide</i>	Inicializaciones en general (variables puntero, número de estación a la que pertenece, etc.), actualiza variables para animaciones, asignación de variables modificadas en la pantalla, cerrar la ventana cuando pasa cierto tiempo, cálculos mientras la ventana esta abierta, control en introducción de parámetros, etc.
<i>Key Scripts</i>	<i>On Key Down</i>	Cuando se pulsa la tecla F1 se reconocen las alarmas si se tiene el permiso adecuado, cuando se pulsa la tecla <i>ESCAPE</i> aparece/desaparece un menú de accesos importantes, etc.
	<i>While Down</i> <i>On KeyUp</i>	Nada.
<i>Touch Links</i>	<i>User inputs</i>	Nada (ya que se utilizan ventanas más fáciles y completas para la introducción de datos)
	<i>Sliders</i>	Se utiliza, por ejemplo, en la pantalla de históricos para marcar el punto de la curva cuyo valor se desea conocer, o para desplazar una línea de referencia, etc.
	<i>Pushbuttons</i>	Junto con los <i>conditional</i> y los <i>data change</i> son los <i>scripts</i> más utilizados. Prácticamente en todas las pantalla tenemos un botón o una zona en la que se puede pulsar para realizar funciones como: pasar a otra pantalla/ventana, cambiar el valor de una variable, realizar cálculos, leer/escribir en ficheros, etc.
<i>Data Change</i>		Cada cierto tiempo: comprueba si el servidor funciona correctamente, lectura y conversiones de los totalizadores, puesta a cero de variables, etc. Control de accesos de los operadores, cálculo de caudales mínimos diarios y detección de saltos bruscos de caudal, detección de anomalías en funcionamiento de bombeos y otros sistemas, genera secuencia consecutiva de variables en la pantalla de históricos, muestra menús al cambiar ciertas variables, se calcula si los niveles en los depósitos están subiendo o bajando, se sincroniza la hora de la central y las estaciones remotas, se llama a la pantalla del sector correspondiente según la zona pulsada en el mapa de la ciudad, controla el cambio de tarifas en los bombeos, etc.

Figura 7.2: Acciones de la aplicación asociadas a los distintos tipos de *Scripts* del SCADA InTouch

7.3.2 Comunicación con el exterior

El software SCADA utilizado incorpora dos mecanismos (DDE, OPC) para la actualización de las variables del proceso. La estructura para el intercambio de información se basa en el tradicional modo cliente/servidor de aplicaciones. En estos casos, como muestra la Figura 7.3, el cliente es el SCADA y el servidor serán los *drivers* que manejan los equipos concretos de adquisición y control.

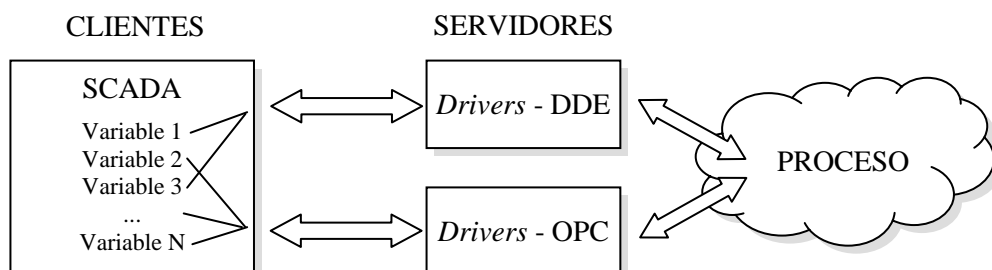


Figura 7.3: Estructura de intercambio de información con el exterior

Una de las ventajas más importante de este sistema de transferencia de información por parte del software de desarrollo, es la transparencia que se consigue al encapsular la gestión de las comunicaciones en un software adicional (*driver*), de tal forma que el ingeniero de la aplicación solo ha de especificar la dirección donde se encuentra la información correspondiente con cada variable de entrada/salida del proceso, en el formato adecuado según el equipo seleccionado.

El *driver* utilizado y la forma de direccionar las variables se trataron en el capítulo 5. Concretamente, para la aplicación de la red de abastecimiento de agua de la ciudad de Granada se ha utilizado el formato Spiral a la hora de direccionar las variables externas, con lo cual, independientemente del equipo al que se acceda (Siemens o SP-5000 en este caso), siempre se especifican las direcciones de la misma forma.

7.3.3 Diseño gráfico

Las pantallas sinópticas están constituidas por un fondo fijo y varias zonas activas que cambian dinámicamente, según los valores leídos en el proceso o en respuesta a las acciones del operador.

Toda la aplicación se basa en la utilización y ubicación fija de una serie de elementos comunes, como se puede apreciar en la Figura 7.4. De esta manera, en cualquier momento y situación, siempre estarán disponibles en la parte superior la misma **cabecera** y en la parte inferior la misma **botonera** (menú inferior), mediante la cual se puede acceder a las funciones más relevantes. La **parte media** de la pantalla, la mayor, es la usada para la representación de la información. Las actuaciones se realizan mediante **ventanas de control** que se abren en primer plano.

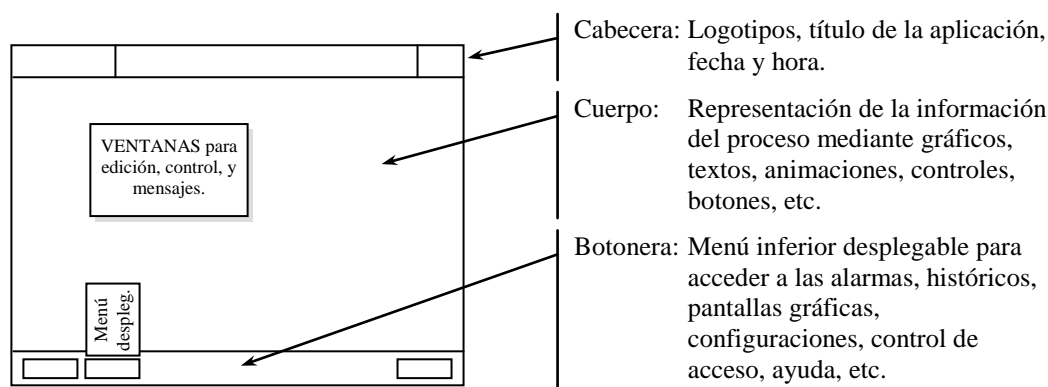


Figura 7.4: Elementos comunes en las pantallas de aplicación

En general los colores que se han utilizado son suaves y uniformes, ya que los colores fuertes (como el rojo y amarillo) se reservan para llamar la atención sobre el operador. En cuanto al cuerpo de la pantalla, la principal o aquella que se visualiza en primer lugar al iniciar el sistema, muestra un esquema general con la información más relevante del proceso con el objetivo de poder seleccionar sobre este la zona o subsistema del que se desea obtener más detalles (ver Figura 7.5). Si la central se encarga de controlar y supervisar procesos distintos o poco interrelacionados los accesos se pueden poner también en la botonera inferior.

También es importante que las ventanas de edición de parámetros y consignas, así como las de visualización de mensajes de avisos o alarmas, sean siempre las mismas y lo más uniformes posible. Y en cuanto a la representación de los sistemas, esta debe ser lo más parecida

a la realidad para lo cual se han de utilizar los medios que proporciona el software de diseño, como son: fotografías, animaciones, gráficos y esquemas similares a los sistemas reales.

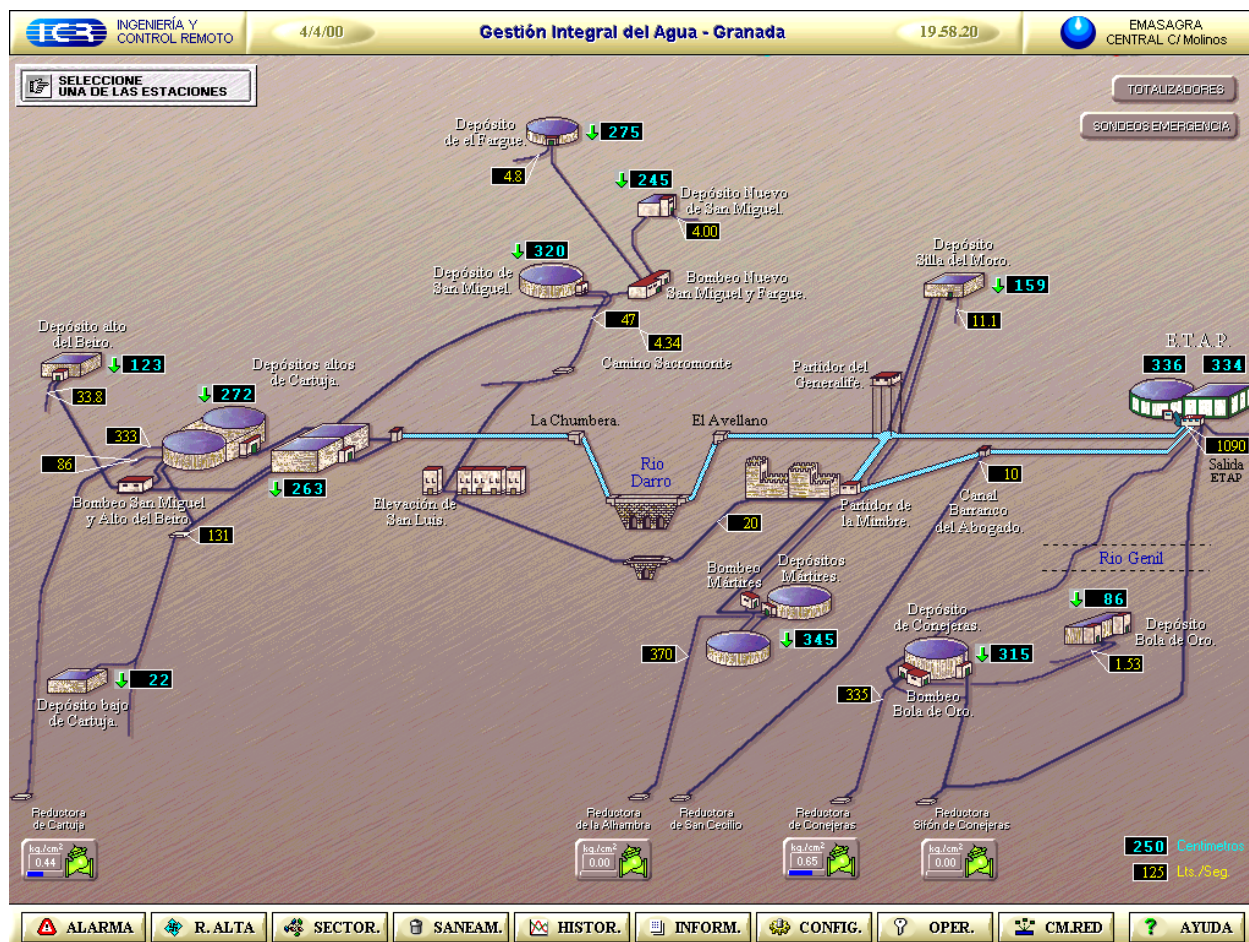
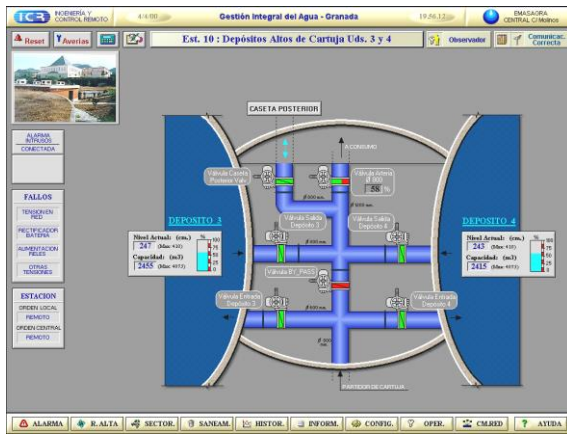
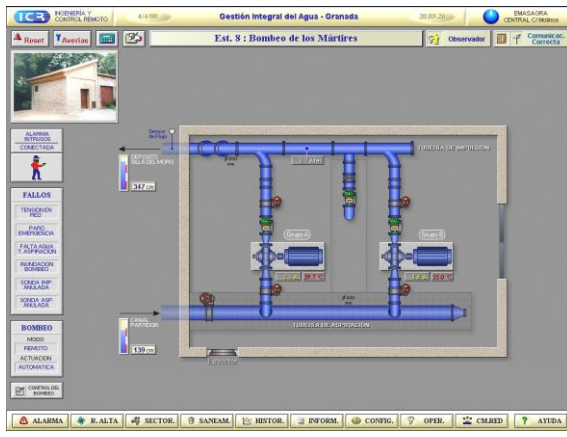
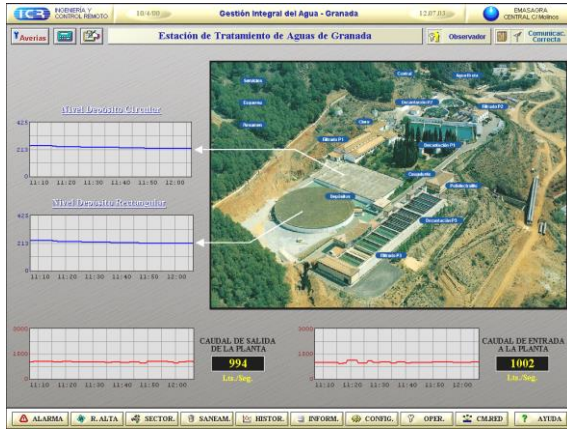


Figura 7.5: Pantalla principal de la aplicación para la red de abastecimiento de agua de Granada

En cuanto a la forma y niveles de acceso a las pantallas desarrolladas, en la tabla de la Figura 7.6 se muestran los caminos de acceso a cada una de ellas, así como un ejemplo representativo de pantallas gráficas o una referencia a otros apartados o capítulos de esta memoria donde encontrarlas.

Menú inferior	Acceso a:		Detalles y referencias	
Sumario alarmas	Grupos Histórico alarmas		En la sección 7.3.4	
Red de Alta	Depósitos	Beiro Altos de Cartuja Bajo de Cartuja San Miguel Nuevo S. Miguel El Fargue Mártires Silla del Moro Conejeras Bola de Oro		
	Bombeos	San Miguel Berio Nuevo S. Miguel Fargue Silla del Moro Bola de Oro		
	Partidores	Generalife Mártires	Pantallas similares a las de los depósitos	
	ETAP			
	Reductoras	Cartuja Mártires Conejeras Sifón	En el capítulo 8	

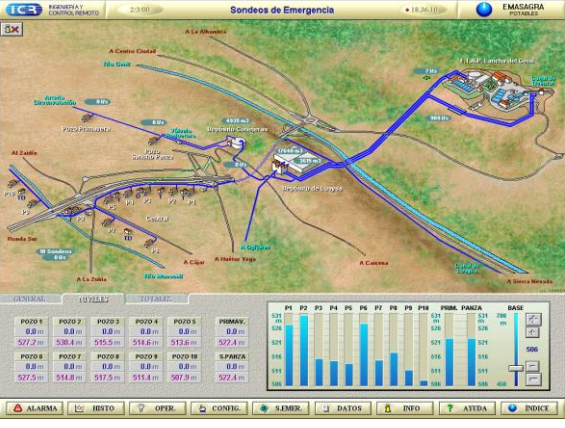
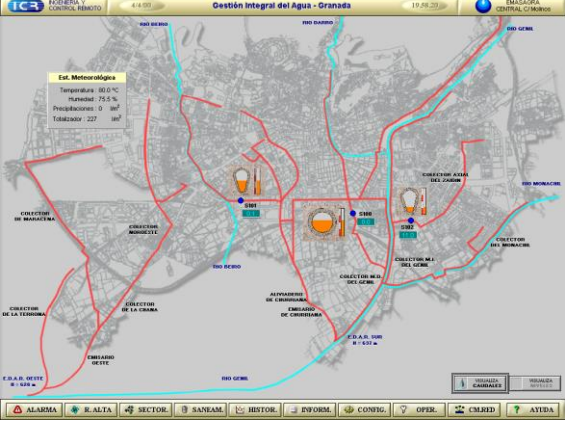
Menú inferior	Acceso a:		Detalles y referencias
	Sondeos emergencia		
Red de Baja	Datos generales		Se trata de una pantalla que recopila todos los datos (niveles y consumos) de la red de alta.
Saneamiento	Cuenca 1 Cuenca 2 Cuenca 3		
Históricos			En la sección 7.3.5
Informes	Aplicación	Diario Semanal Mensual	En la sección 7.3.6
	Excel	Consumos Bombeos	
Configuración	General		En la sección 7.3.7
	Niveles alarma		
	Caudales alarmas		
	Apertura max. válvulas		
Control acceso			En la sección 7.3.8
Ayudas	Generales		En la sección 7.3.9
	Aplicación		
	Agenda teléfonos		

Figura 7.6: Tabla de accesos a las pantallas de la aplicación

7.3.4 Tratamiento de las alarmas

Para la gestión de alarmas se han diseñado dos pantallas, sumario e histórico, más completas que los componentes estándar que proporciona el SCADA para ello.

Pulsando el botón *ALARMA* del menú inferior (y por lo tanto siempre accesible) se accede al sumario de alarmas, donde habrá una línea por cada suceso (ver Figura 7.7). Cada línea puede mostrarse en dos colores: amarillo para alarmas disparadas no reconocidas, y blanco para alarmas disparadas ya reconocidas por el operador.

FECHA	HORA	TIPO	DESCRIPCIÓN	NOMBRE	GRUPO	VALOR
14/08/99	16:05:50	ALM LOLO	Caudal Salida al Albaicín	CAUSAL_7	Partidor_Mimbres	0
14/08/99	16:05:50	ALM LOLO	Nivel Canal Albaicín	NCALBA_7	Partidor_Mimbres	0
14/08/99	16:05:50	ALM LOLO	Nivel Canal Estación Depuradora	NCDEPU_7	Partidor_Mimbres	0
14/08/99	16:05:50	ALM LOLO	Nivel Canal Partidor del Generalife	NCPAR_7	Partidor_Mimbres	0
14/08/99	16:05:50	ALM LOLO	Caudal Salida Dep. 1 y 2 Cartuja	CAUD12_1	Dep_Altos_Cartu	0
14/08/99	16:05:50	ALM LOLO	Nivel Canal Partidor Dep. 3 y 4	NCP34_10	Dep_Altos_Cartu	0
14/08/99	16:05:50	ALM LOLO	Nivel Canal Partidor Dep. 1 y 2	NCP12_10	Dep_Altos_Cartu	0
14/08/99	16:05:50	ALM LOLO	Nivel Dep. 2 en Alto Cartuja	ND2_10	Dep_Altos_Cartu	0
14/08/99	16:05:50	ALM LOLO	Nivel Dep. 1 en Alto Cartuja	ND1_10	Dep_Altos_Cartu	0

Figura 7.7: Pantalla de visualización del sumario de alarmas

Cuando aparece una nueva alarma en el sistema, el botón *ALARMA* comienza a parpadear en color rojo indicando la presencia de una alarma no reconocida.

Cuando se pulsa el botón *RECONOCIMIENTO*, la línea de alarma cambia a color blanco (alarma ya reconocida), y así continuará hasta que se restaure a estado normal, momento en el que desaparecerá del sumario de alarmas.

Cada línea del sumario de alarmas contiene la siguiente información: fecha, hora, tipo de alarma, descripción, nombre de la variable que genera la alarma, grupo (estación) a la que pertenece la alarma, y valor de la variable cuando genera la alarma.

Para que el sumario muestre las alarmas, dependiendo del grupo al que pertenecen, se dispone de una ventana de elección de grupo (ver Figura 7.8).



Figura 7.8: Ventana de elección de grupos de variables para visualizar alarmas

En la parte superior derecha del sumario de alarmas se dispone de un botón para pasar a la pantalla del histórico de alarmas, en la cual se puede visualizar la historia reciente del sistema en lo referente a registro de alarmas disparadas, reconocidas y restauradas. La distinción de las anotaciones que se realizan en el histórico de alarmas se hace, al igual que en el sumario, por colores. Y la información que se muestra es también semejante a la del sumario de alarmas.

Los datos referentes a las alarmas se almacenan en ficheros diarios, que además se pueden imprimir, pulsando el botón correspondiente, para tener una copia en papel.

7.3.5 Los históricos

Para visualizar la evolución de las variables analógicas en el tiempo también se ha diseñado un módulo específico. En la Figura 7.9 se muestra un ejemplo de pantalla de históricos a la que se accede pulsando el botón *HISTO* del menú inferior, y donde se puede especificar las variables a representar, la duración, el tipo de gráfica, la fecha y la hora de comienzo y el rango representado. Modificando estos parámetros se puede acceder a toda la información que se ha ido guardando a lo largo del tiempo y representarla como se desee. Además, en cualquier momento se puede obtener una copia impresa de lo que muestre la pantalla.

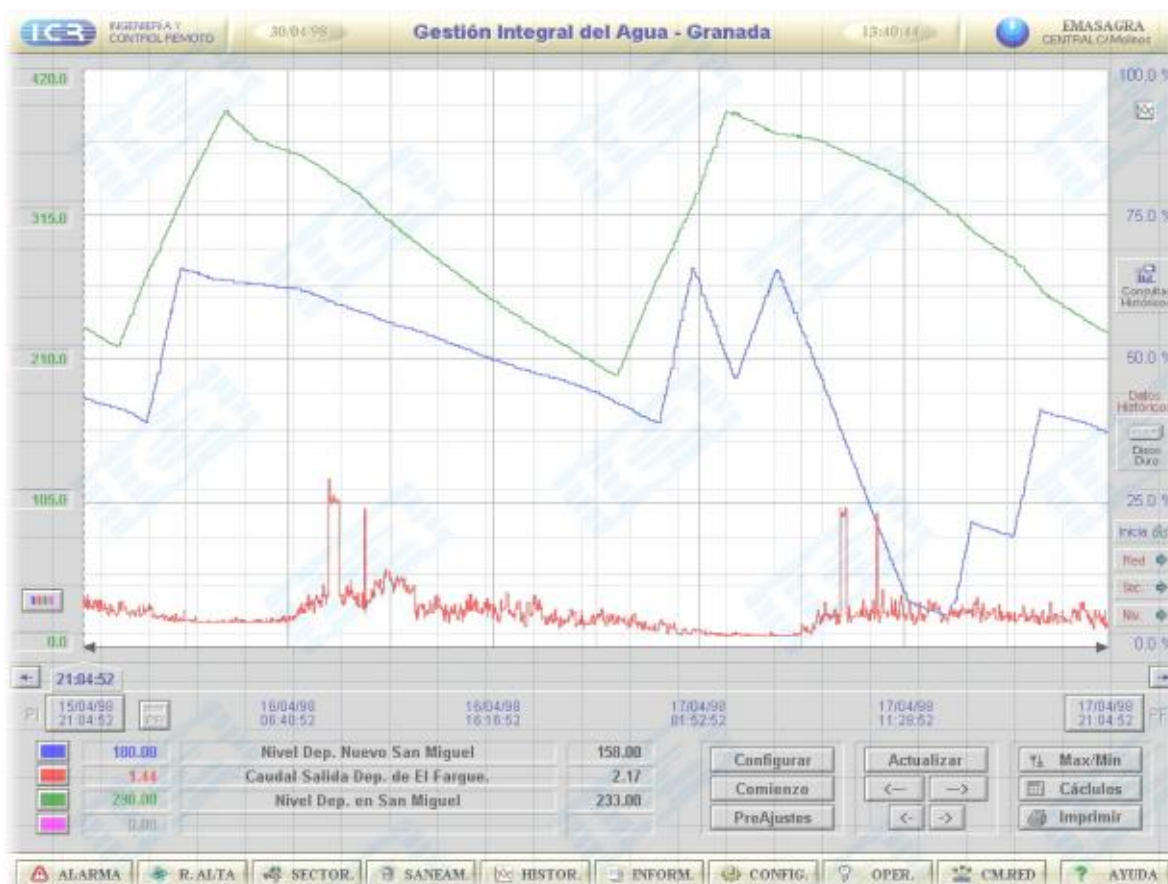


Figura 7.9: Pantalla de históricos diseñada en el SCADA

Sin duda, esta es una de las pantallas más utilizadas por parte de los operadores y personal más especializado. Con ella se puede ver la evolución de variables, como el nivel de los depósitos y canales, los caudales de consumo, y las presiones. También se pueden detectar fugas y fraudes, así como funcionamientos anómalos en grupos de bombeos, observar las operaciones de apertura/cierre porcentual de las válvulas por parte de los operadores, etc.

Si se desea tratar la información almacenada mediante otros programas (hojas de cálculo, bases de datos, programas estadísticos) se pueden generar ficheros para poder importarlos desde cualquiera de estos programas y así realizar gráficos, informes o estudios personalizados. Para ello se ha creado una ventana a la que se accede pulsando el botón *CONSULTAS* desde la pantalla de históricos, donde se pueden elegir las variables, la fecha de comienzo, la duración y la periodicidad de los datos a obtener, tal y como se muestra en la Figura 7.10.

NIVELES DE DEPÓSITOS			CAUDALES INSTANTANEOS			NIVELES DE CANALES		
ND1_10	Nivel Dep. 1 en Alto Cartuja	CAUD12_10	Caudal Salida Dep. 1 y 2 Cartuja	NCP12_10	Nivel Canal Partidor Dep. 1 y 2			
ND2_10	Nivel Dep. 2 en Alto Cartuja	CAUA600_10	Caudal Salida Arteria 600 Cartuja	NCP34_10	Nivel Canal Partidor Dep. 3 y 4			
ND3_10	Nivel Dep. 3 en Alto Cartuja	CAUA800_10	Caudal Salida Arteria 800 Cartuja	NCDAR_6	Nivel Canal Sifon del Darro			
ND4_10	Nivel Dep. 4 en Alto Cartuja	CAUSAL_3	Caudal Salida Dep. Alto Beiro	NCDEPU_6	Nivel Canal Estacion Depuradora			
NDEP_3	Nivel Dep. en Alto del Beiro	CAUSAL_5	Caudal Salida Dep. Silla Del Moro	NCALBA_7	Nivel Canal Albaicin			
NDEP_2	Nivel Dep. en Bajo de Cartuja	CAUSAL_7	Caudal Salida al Albaicin	NCDEPU_7	Nivel Canal Estacion Depuradora			

Figura 7.10: Configuraciones para las consultas de datos almacenados por el SCADA

7.3.6 Generación de informes

La aplicación genera dos tipos de informes: a) Informes desde la aplicación: diario, semanal y mensual de consumos, y b) Informes a partir de ficheros en formato de texto separados por coma: mensuales de consumos y bombeos.

En la Figura 7.11 se muestran informes del tipo a), y en la Figura 7.12 se muestra cómo se ha diseñado el informe de consumo, tipo b), utilizando la hoja de calculo EXCEL.

La ventaja principal de estos informes está en que se generan automáticamente, sin ningún trabajo para los operadores. Sin embargo, y con la experiencia adquirida, actualmente se está también volcando los datos en bruto adquiridos por la aplicación sobre una base de datos estándar para que así el operador pueda realizar los cálculos y los informes que necesite en cada momento. Se consigue así mayor integridad en la información y la posibilidad de cruzar más fácilmente estos datos con los obtenidos por otras fuentes (en nuestro caso, por la facturación), quedando los informes automáticos a medida destinados a consultas internas rápidas.

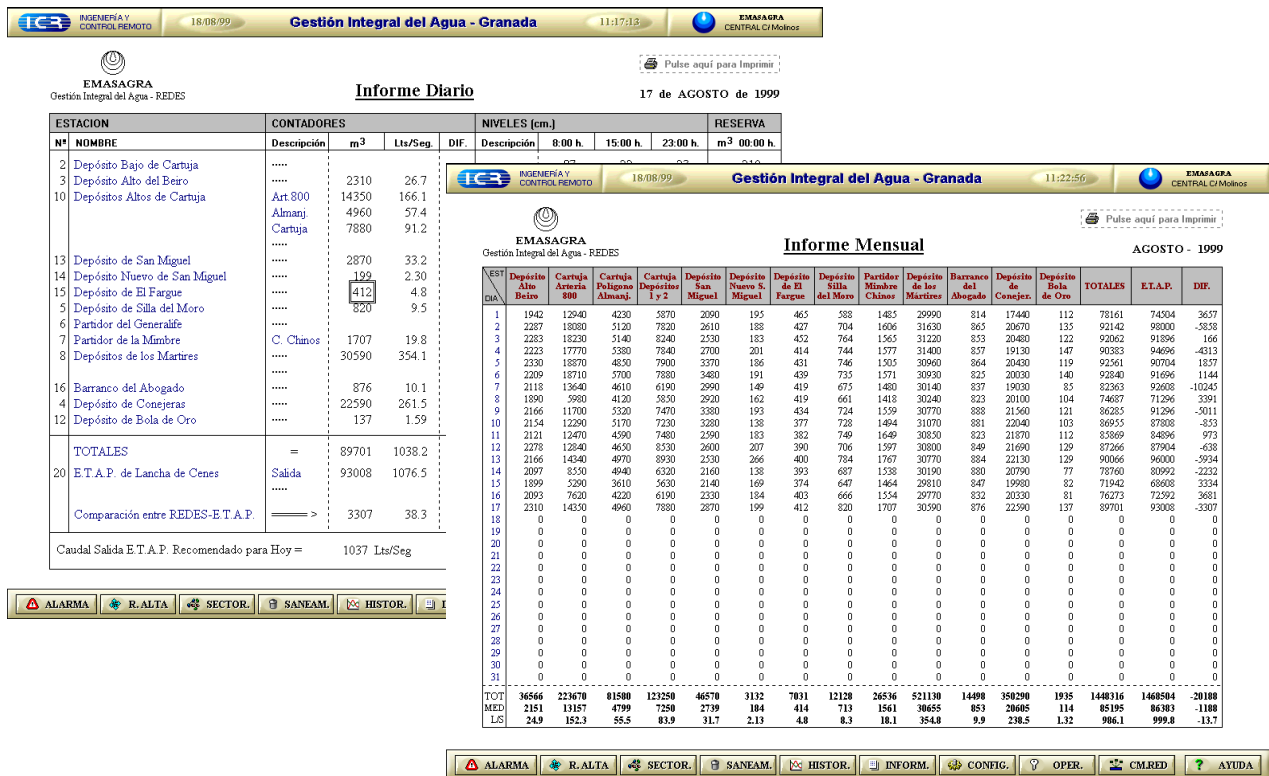


Figura 7.11: Informes generados por la aplicación SCADA

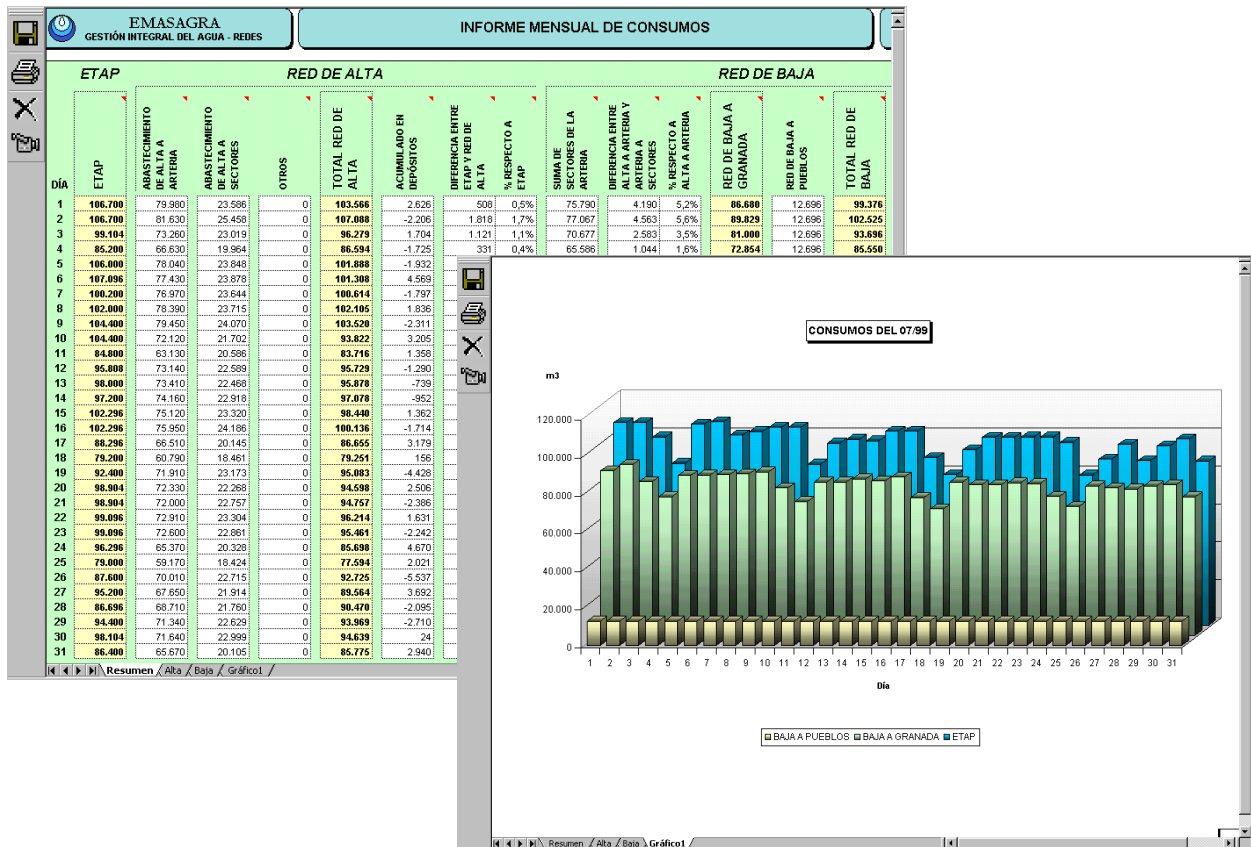


Figura 7.12: Informe mensual de consumo en formato EXCEL

7.3.7 Nivel de configuración

La metodología propuesta para el diseño de aplicaciones aboga por la máxima funcionalidad, para lo cual es necesario una gran flexibilidad y capacidad de configuración. Para ello se han desarrollado pantallas específicas de configuración como las que se muestran en la Figura 7.13, con las que se pueden parametrizar, entre otros, los siguientes elementos:

- Niveles máximos y mínimos de alarmas en depósitos, canales y caudales de salidas.
- Detección de saltos bruscos en los caudales de salida de depósitos y entrada a sectores.
- Detección de mínimos caudales diarios sobrepasados para la localización de fugas latentes.
- Limitación de la apertura máxima de aquellas válvulas consideradas críticas.
- Definición de las tarifas horarias contratadas y su asignación a estaciones de bombeo.
- Activación/desactivación de las comunicaciones con las estaciones remotas.
- Definición de tipo de nodo (servidor o cliente).
- Habilitar o no la impresión de informes, así como establecer el día en que se imprimirán automáticamente.

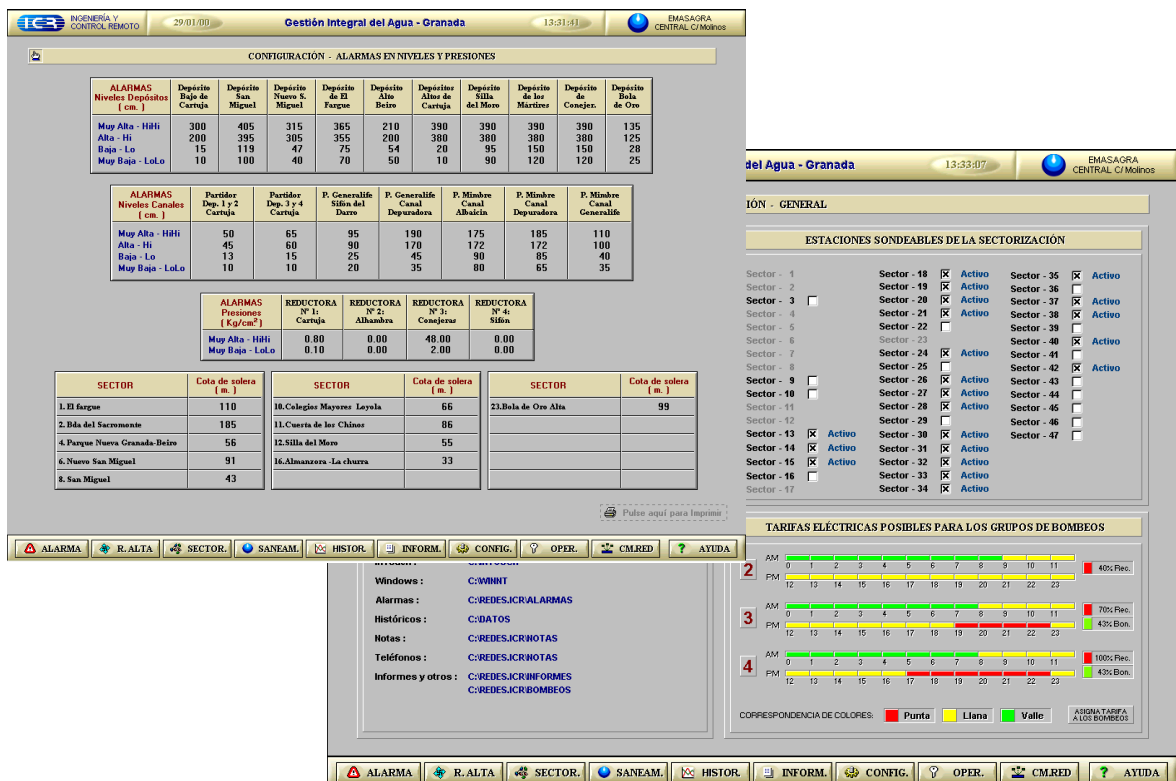


Figura 7.13: Pantallas típicas de configuración en la aplicación

7.3.8 Control de acceso

Como se comentó al hablar de las características generales de InTouch, este SCADA implementa la posibilidad de restringir el acceso a determinadas acciones y a determinados grupos de usuarios. Para que un usuario perteneciente a un grupo definido pueda operar sobre partes de la aplicación a las que sólo su grupo o superiores tienen acceso, debe introducir su nombre y su clave secreta de acceso; a partir de ese momento cualquier acción queda registrada con el nombre de ese operador, permitiendo un control posterior de las operaciones realizadas en caso de mal funcionamiento o negligencia.

Pulsando el botón *OPER* del menú inferior se muestra la ventana diseñada para poder abrir una sesión, ver Figura 7.14.



Figura 7.14: Ventana de control de acceso en la aplicación SCADA

Para evitar actuaciones no deseadas, si un operador no cierra su sesión cuando haya terminado su trabajo, la aplicación la cerrará automáticamente pasado cierto tiempo. Y cuando un usuario pretende realizar una operación para la cual no está autorizado, se muestra brevemente el mensaje de la Figura 7.15.



Figura 7.15: Mensaje acceso no autorizado en SCADA

7.3.9 Ayudas al operador

La principal ayuda al operador ha sido el diseño, mostrado en los apartados anteriores, orientado a facilitar la tarea a los usuarios. No obstante, se han generado además pantallas de ayuda interactivas, ver Figura 7.16 y Figura 7.17, para que el usuario pueda:

- Observar las correspondencias entre objetos gráficos y dispositivos físicos.
- Mostrar las animaciones que se han construido para resaltar ciertos estados del proceso.
- Practicar con los controles típicos diseñados.

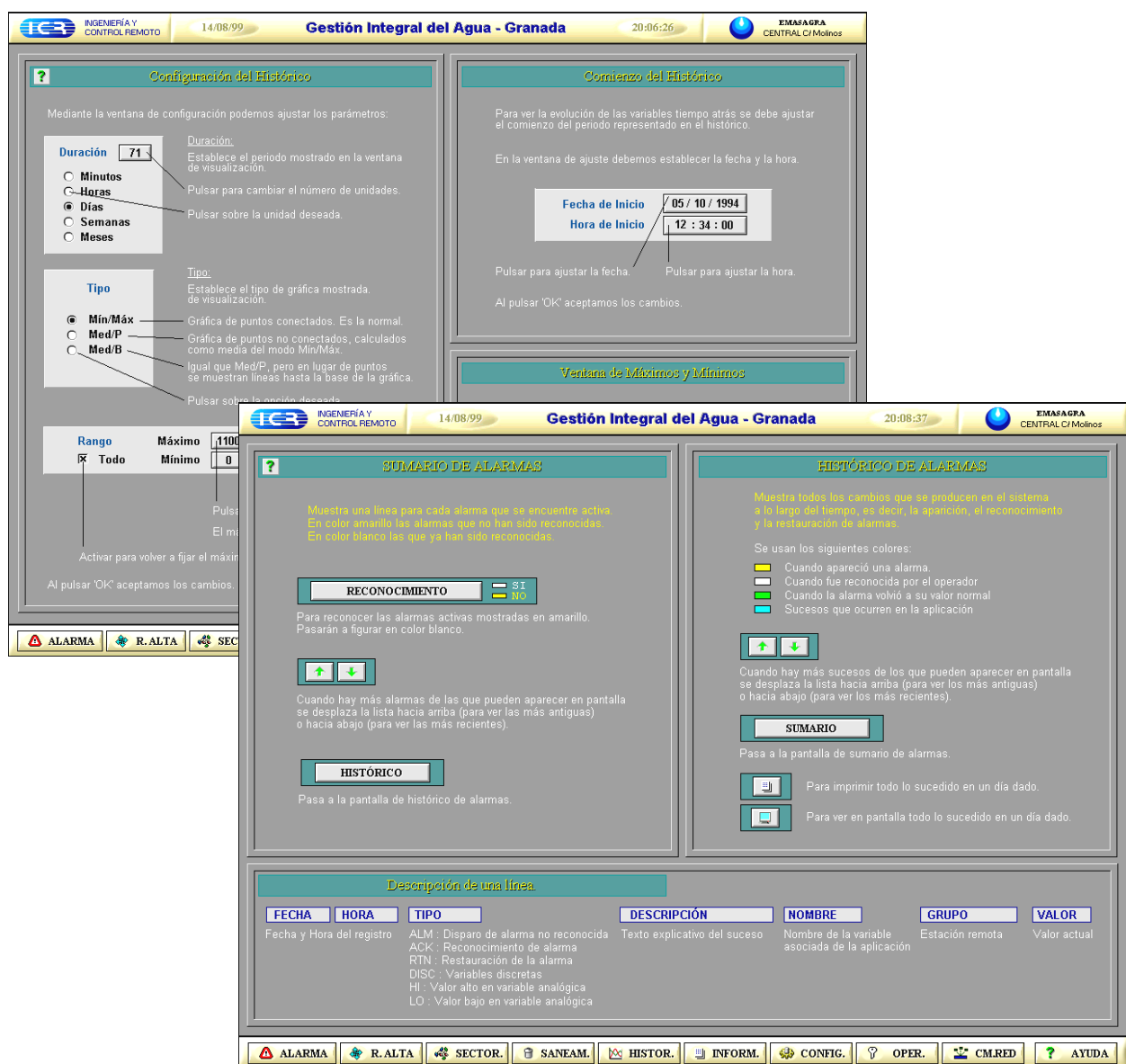


Figura 7.16: Ejemplos de pantallas de ayuda de elementos comunes a cualquier aplicación SCADA

Pulsando sobre el botón *AYUDA*, situado también en el menú inferior, se despliega una ventana para la elección de la pantalla de ayuda deseada. Existe ayuda para las pantallas de alarmas, para manejar y configurar la pantalla de históricos, para el control de acceso de los operarios, para familiarizarse y manejar las distintas ventanas de cambios de consignas enteras, reales o de fecha y hora, y para familiarizarse también con los detalles concretos de la aplicación (elementos de control, paneles informativos y de error comunes, etc.). Existe también la posibilidad de manejar una agenda de teléfonos.

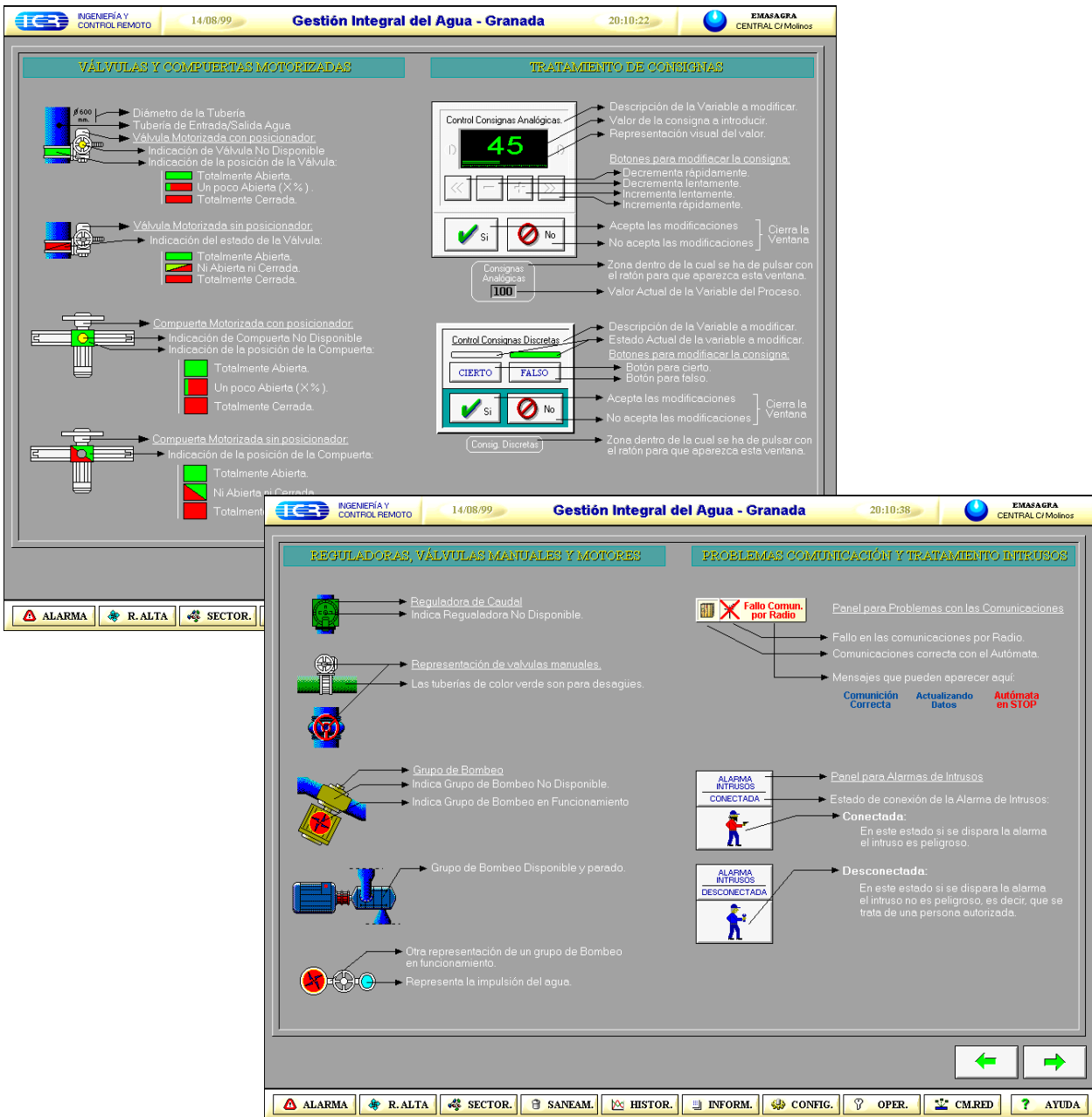


Figura 7.17: Ejemplos de pantallas de ayuda en la aplicación de la Gestión Integral del Agua en Granada

7.3.10 La aplicación en números

En la tabla de la Figura 7.18 se cuantifican los elementos gráficos, variables y módulos de programas del software de la estación central.

CONCEPTO	CANTIDAD
Pantallas completas (proceso, configuración, ayudas, etc.)	65
Ventanas de edición, control y mensajes	260
Variables E/S	2.589
Variables internas	2.454
Módulos de programa (<i>scripts</i>)	280

Figura 7.18: Recursos del software de la estación central

Las aplicaciones en las subcentrales de la ETAP y de las EDARs se han diseñado también utilizando esta misma metodología, con resultados similares. Como resumen de estas aplicaciones se da también la estimación en números (Figura 7.19), así como una serie de pantallas representativas (Figura 7.20 y Figura 7.21). Para finalizar, se proporciona una tabla global con el total de recursos que se han necesitado para el control y supervisión del ciclo integral del agua en la ciudad de Granada, tal y como se encuentra actualmente (Figura 7.22).

CONCEPTO	CANTIDAD
Pantallas completas (proceso, configuración, ayudas, etc.)	130
Ventanas edición, control y mensajes	198
Variables E/S	3.850
Variables internas	2.040
Total <i>scripts</i> (módulos de programa)	289

Figura 7.19: Recursos del software de la ETAP y las EDARs

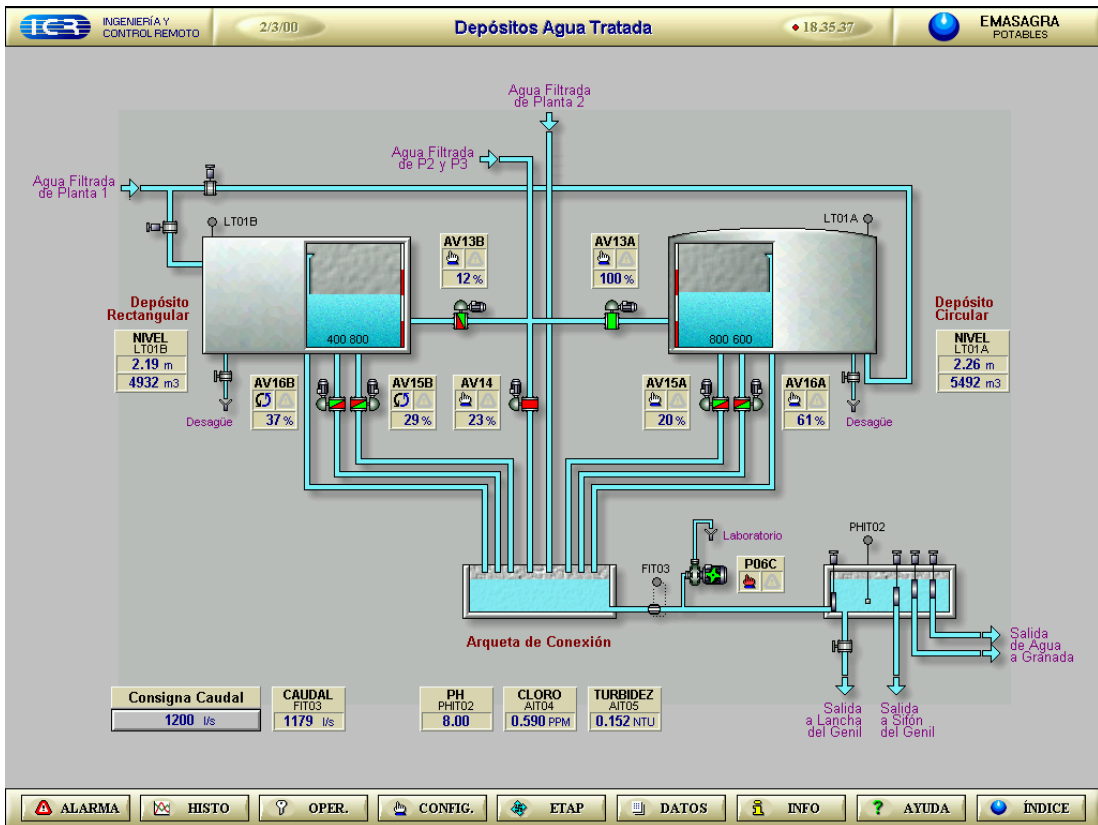
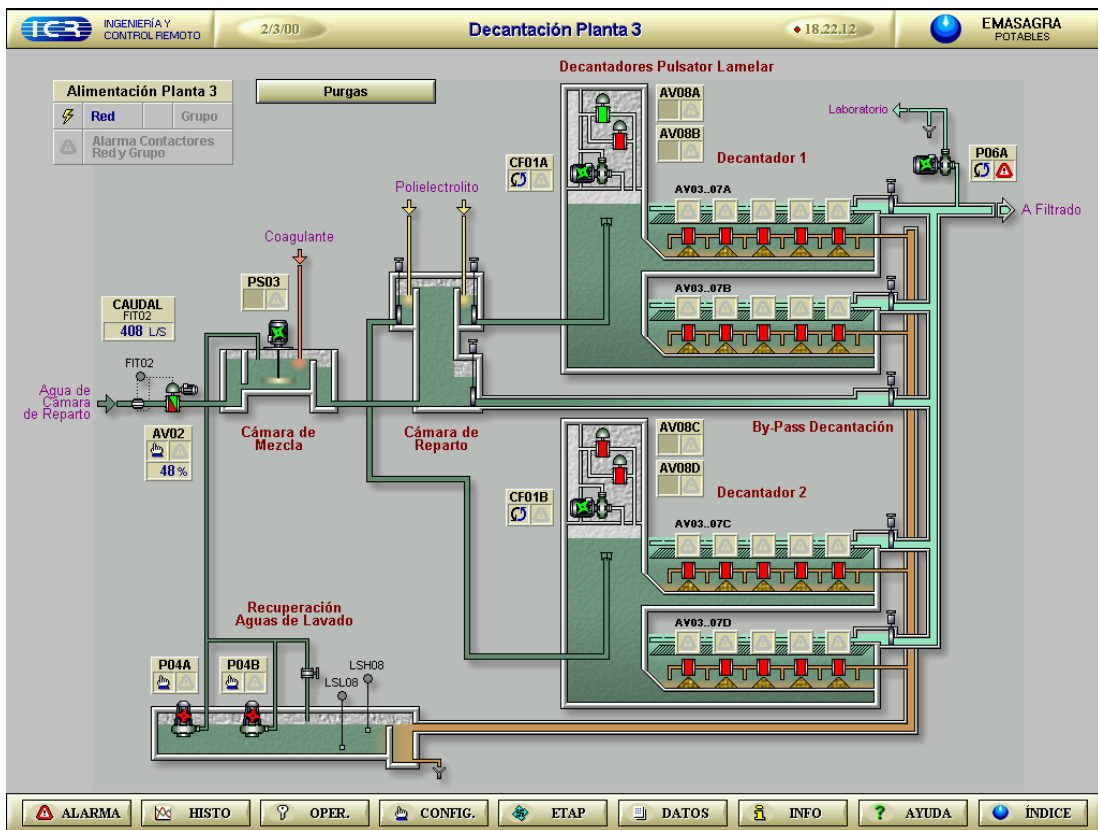


Figura 7.20: Algunas pantallas diseñadas para la ETAP

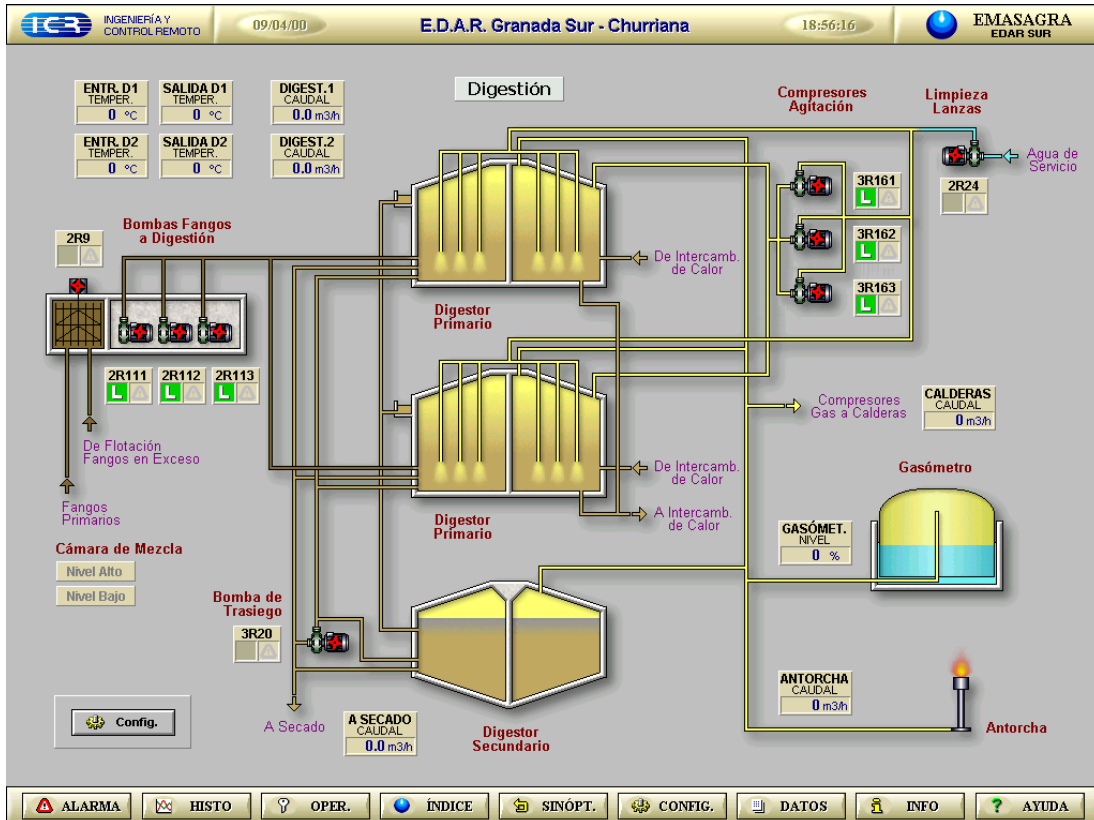
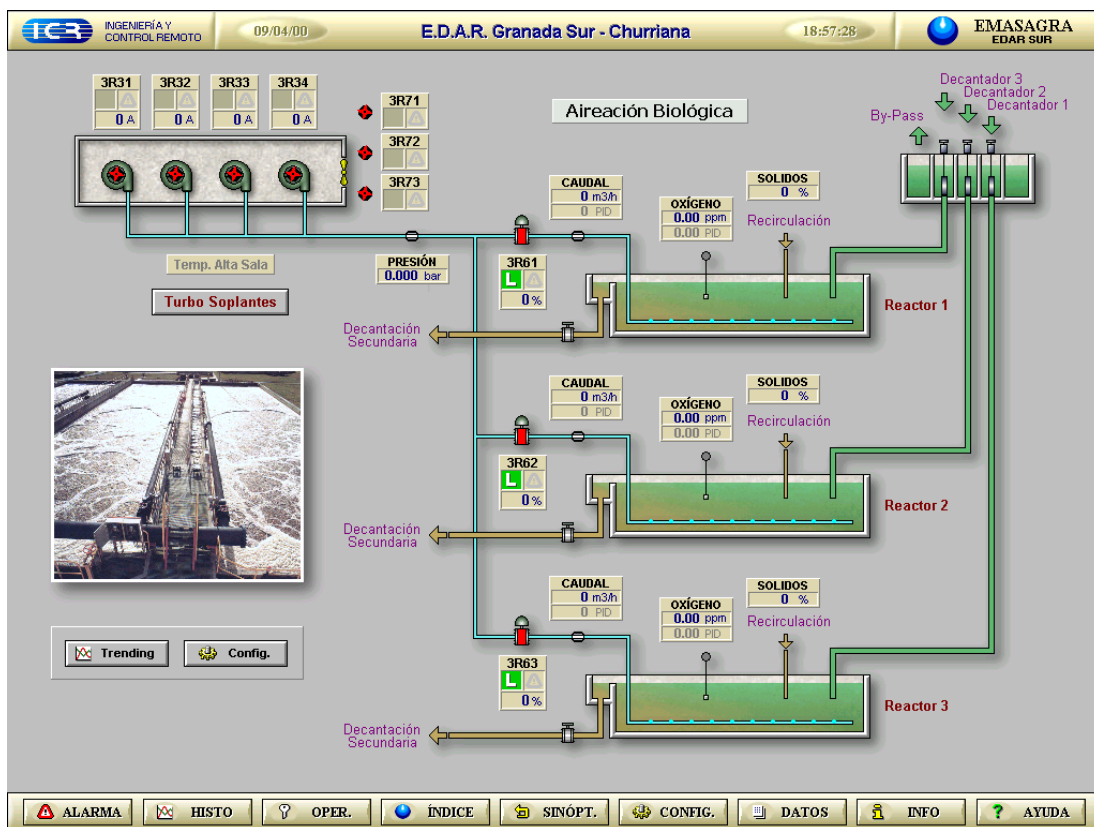


Figura 7.21: Algunas pantallas diseñadas para las EDARs

CONCEPTO	CANTIDAD
Pantallas completas (proceso, configuración, ayudas, etc.)	195
Ventanas edición, control y mensajes	458
Variables E/S	6.439
Variables internas	4.494
Total <i>scripts</i> (módulos de programa)	569

Figura 7.22: Recursos totales del software de control del ciclo integral del agua en Granada

7.4 Aplicaciones externas

Para ciertas tareas es más aconsejable utilizar programas independientes al SCADA porque permiten, entre otras cosas, utilizar el lenguaje de programación más acorde con cada aplicación, estructurar mejor los programas, no saturar las variables del SCADA y reutilizar software.

Concretamente se han implementado las siguientes aplicaciones externas:

- Spiral GSM 9000 (descrito en el Capítulo 4) para la supervisión y control de la aplicación utilizando telefonía móvil GSM-SMS.
- ICeR (descrito en el Capítulo 5) como gestor de comunicaciones con los sistemas de control distribuido.
- Predicción del consumo y distribución automática del caudal de salida de la ETAP entre los depósitos principales del abastecimiento (tratados en el siguiente capítulo sobre optimizaciones).

7.5 Conclusiones

Se ha descrito en este capítulo la metodología software empleada para lograr los objetivos propuestos. Se ha mostrado con un ejemplo las posibilidades que ofrece nuestro diseño, así como los recursos finales que se necesitan para el control óptimo de un proceso tan heterogéneo (por la diversidad que presenta) como es la automatización del ciclo integral del agua de una ciudad.

Una vez validado, este diseño ha sido aprobado para ser implantado por algunas de las principales empresas gestoras de agua de nuestro país, así como en diversas aplicaciones de control distribuido de similares características. En el apéndice A se muestra una lista detallada de las referencias realizadas en colaboración con la empresa ICR S.A.

Prueba del éxito de la metodología de software de integración de procesos descrita en este capítulo, está en el hecho de que una vez implantada, se produce un cambio radical en los procedimientos de gestión y explotación, y una rápida adaptación a los mismos.

Hemos de resaltar además las siguientes mejoras:

- La interfaz hombre-máquina (MMI) diseñada es homogénea, funcional y muy amigable, de manera que permite un manejo sencillo hasta para el operario más inexperto.
- El operador dispone de una visión global del estado de la red en tiempo real, sin más que observar la pantalla principal.
- Las alarmas llaman la atención tanto al operador de la central (intermitencia en rojo y registro en impresora) como al operador de mantenimiento que se encuentre desplazado realizando cualquier otra tarea (mediante el sistema Spiral GSM 9000).
- La pantalla de históricos permite, además del seguimiento de la evolución de variables como el nivel, la presión o el caudal, la detección de fugas y fraudes. Por ejemplo, el sistema avisa de forma automática al operador de eventos tales como cambios bruscos de variables o consumos anómalos con la posibilidad de realizar un estudio posterior más detallado de la evolución dinámica.
- Se generan informes automáticamente en papel y se almacenan en ficheros con un formato estándar para que el operador pueda generar informes más específicos.
- El control de acceso, al registrar también cualquier acción con el nombre del operador que la realiza, permite un seguimiento más controlado, a la vez que una mayor atención por parte del operario.
- Las pantallas de ayuda se pueden utilizar además para practicar con algunos de los controles más comunes.

- Con los módulos de optimización, no sólo se aprovechan mejor los recursos humanos, sino que además se obtienen ahorros económicos importantes, presiones más acordes con el consumo, localización de fugas, y distribución automática de caudales. Estas ventajas se describirán con más detalle en el capítulo siguiente.

Capítulo 8:

Optimizaciones en la red de distribución de agua

La industria del agua ha sufrido cambios radicales desde finales de los 80 y dichos cambios continúan sucediéndose rápidamente. Las compañías gestoras del agua han estado y seguirán estando bajo una presión creciente para reducir sus costes e incrementar el servicio, dentro de un marco legal cada vez más estricto y en el cual se necesita más información para cumplir y verificar la realización del servicio. La solución para lograr este objetivo pasa por la integración cada vez mayor de los sistemas automáticos junto con las optimizaciones dentro de las aplicaciones de control y supervisión de datos.

Este último capítulo está dedicado a la optimización de sistemas hidráulicos, y tiene como objetivo determinar de forma automática el estado de los elementos de control presentes en la red, estaciones de bombeo y válvulas reguladoras de presión y caudal, para que atiendan de forma eficiente y con un coste mínimo a cierta demanda hidráulica.

Después de la introducción y de describir los objetivos que se persiguen con la optimización en el ámbito de las redes de abastecimiento de agua (secciones 1 y 2), en las secciones siguientes se muestran los detalles de los métodos y algoritmos usados así como los resultados obtenidos en cada uno de ellos. Finalmente se presenta, como conclusiones, un resumen de todo el trabajo realizado y las posibles líneas futuras de investigación y desarrollo en este campo.

8.1 Introducción.

La estrategia orientada a la optimización de procesos y la red desarrollada, descrita en los capítulos anteriores, aportan un beneficio mucho mayor al permitir que una compañía suministre sus servicios de forma coordinada y eficiente.

Para ello, y una vez disponible la automatización total de la red, se han realizado las implementaciones necesarias en la aplicación que permiten mejorar (optimizar) la explotación del sistema, y así alcanzar los objetivos del control óptimo que se describen en la siguiente sección.

No obstante, hay que tener presente que las técnicas de optimización se han ido sofisticando cada vez más para resolver problemas también cada vez más complejos, pero llegados a este punto nos podemos plantear si el camino más adecuado en el futuro para resolver ciertos problemas es la consecución de algoritmos cada vez más sofisticados y robustos, o bien, conviene replantear el problema de nuevo desde el principio argumentando:

¿De qué sirve esforzarse por conseguir algoritmos cada vez más potentes y complicados, si la solución de ciertos problemas depende mucho más de su propia formulación que de la exactitud de los modelos matemáticos?

Esta última idea es la que nos ha impulsado a plantear la resolución y optimización de ciertos problemas, ya abordados con técnicas de programación clásicas, desde otra perspectiva pero con los mismo objetivos.

Por último, señalar que algunas de estas optimizaciones se han desarrollado con el propio software SCADA y otras, que utilizan técnicas o metodologías de programación más recientes y computacionalmente costosas, descritas en el Capítulo 2, se deben de realizar con programas externos, por necesitar mayores prestaciones en su implementación, y conectarse a la aplicación SCADA (vía DDE/OPC) para leer/modificar en tiempo real los parámetros del proceso necesarios para la optimización.

8.2 Objetivos del control óptimo

Los objetivos del control óptimo en sistemas hidráulicos se pueden englobar en los siguientes apartados:

- **Mínimo coste.** La reducción de costes está asociada fundamentalmente al ahorro que se puede conseguir en el consumo de electricidad de los grupos de bombeo. Durante años se ha invertido un considerable esfuerzo en la investigación y desarrollo de métodos matemáticos para resolver problemas de optimización en este sentido [Cou-95, Cou-92, Jac-91, Cre-88].
- **Control de fugas.** El método principal para obtener una mejora de la eficiencia es, por supuesto, el control de las fugas [Vel-95, Ger-95, Lig-93]. Un valor típico del porcentaje de agua no contabilizada se encuentra entre el 15% y el 40% del suministro total, y se concentra principalmente en las tuberías de pequeño diámetro del sistema de distribución. Puesto que el nivel de fugas puede ser relacionado con la presión del sistema, el control de la presión de distribución puede ser un modo valioso para reducir los costes de suministro.
- **Seguridad.** Uno de los beneficios esperados de un control automatizado es la mejora en la seguridad de las operaciones, ya que una de las causas más frecuentes de fallo en sistemas complejos es el error humano [Jar-93]. Como los sistemas de distribución son cada vez más complejos e interconectados, la tecnología de la información es el único modo de operar eficientemente.
- **Calidad de servicio.** La calidad del servicio se está convirtiendo en una preocupación creciente entre las empresas de suministro de agua, ya que estas deben de cumplir con estándares cada vez más exigentes en términos de calidad de agua y servicios. Los sistemas de apoyo en la toma de decisiones pueden ayudar a cumplir la normativa, ofreciendo herramientas valiosas para la comprensión, el análisis y la decisión entre las estrategias apropiadas, tanto durante una situación crítica como en la operación diaria del sistema [Jar-91].
- **Reducción del equipo técnico.** La supresión parcial o total de un turno de trabajo es una motivación significativa para implementar un control por computador. El tamaño del sistema de distribución y la estimación del ahorro pondrá de manifiesto si son aplicables la automatización y/o la optimización [All-88].

Para alcanzar los objetivos anteriores se ha desarrollado una metodología basada en herramientas y procedimientos funcionales que permite, para cada problema en particular, determinar los parámetros de los distintos elementos de control.

8.3 Consumo de energía en grupos de bombeos.

8.3.1 Introducción y objetivos

Los servicios de explotación de agua potable han tenido siempre como objetivo minimizar el coste del consumo de energía en los bombeos a depósitos de suministro de diferentes formas:

- En muchas ocasiones, sólo existen unas sondas que detectan la falta de agua en el depósito, por lo que el bombeo es "a la demanda directa", sin posibilidad de previsión ni de discriminación horaria.
- En otras ocasiones, se realiza una discriminación con reloj sobre varios grupos de sondas detectoras de nivel. Este procedimiento permite una programación horaria del funcionamiento de la bomba, con el correspondiente ahorro en el coste de la energía, pero requiere revisar periódicamente la posición de las sondas en función de los consumos de temporada.

En cualquiera de los dos casos anteriores siempre pueden surgir dos problemas:

- Si el depósito es excesivamente grande para la demanda, podemos bombear en horas valle, (incluidos sábados y domingos), con el consiguiente ahorro económico, pero con el inconveniente de pérdida de calidad, al desaparecer el efecto del cloro por exceso de permanencia del agua en el depósito, lo que obligaría a una cloración local, o
- Si el depósito es excesivamente pequeño, estamos obligados a bombear más por demanda que por discriminación horaria.

Cualquiera de las situaciones anteriores son difíciles de controlar sobre todo si el funcionamiento de las bombas y la demanda de consumo quedan sin registrar.

La solución que hemos implantado consiste en un método que permita el funcionamiento de los bombeos principalmente en horas valle, en segundo lugar en horas llanas, y en horas punta sólo si es realmente imprescindible. Hasta aquí, la aplicación es similar a las existentes con "sondas" o niveles de consigna con distintos periodos de discriminación horaria ya que los periodos horarios son los convencionales: horas valle, llana y punta¹.

La novedad consiste en que *los niveles se calculan dinámicamente para cada intervalo horario en función de la demanda del día anterior en esos intervalos horarios, con una*

¹ Intervalos horarios en los que las empresas eléctricas bonifican (horas valle) o penalizan (horas punta) a sus clientes para optimizar los recursos energéticos.

corrección en función de la demanda del día actual. Los detalles de este procedimiento se explican en el apartado siguiente, y los objetivos que se persiguen son:

- Bombear, con el mínimo de arranques posibles, el máximo de agua necesaria para el llenado del depósito fuera de las horas punta, y si es posible también fuera de las horas llanas, con el consiguiente ahorro económico.
- Evitar la acumulación innecesaria de agua en el depósito si éste es excesivamente grande, para impedir la pérdida de eficacia del cloro.
- Establecer niveles de seguridad adecuados para garantizar que los niveles de depósitos se encuentran entre los máximos y mínimos permitidos, que el bombeo no arranque si no hay agua suficiente, o que por fallos en la comunicación con la central siga bombeando hasta el nivel deseado.
- Proporcionar en tiempo real la máxima información al operador para ayuda a la toma de decisiones.

Construir una aplicación que cumpla estos objetivos se reduce a definir los algoritmos de cálculo que permitan prever los consumos y establecer los niveles de los diferentes periodos horarios de una forma dinámica de un día para otro. El jefe del servicio sólo debe proponer las consignas generales como, por ejemplo, el volumen mínimo a contener en el depósito (nivel mínimo de seguridad), el volumen máximo antes de rebosar (nivel de rebose), el porcentaje de volumen de seguridad a bombear, etc.

El funcionamiento de la bomba de elevación al depósito se adaptará a los criterios manuales y automáticos establecidos anteriormente. La única condición que se impone al sistema es el arranque del bombeo en horas llanas en el instante apropiado, para que al final de este tramo horario, con tarifa eléctrica más económica, el depósito quede lo más cercano posible al nivel de parada (ver Figura 8.1). Se puede mejorar este método, aplicando técnicas de búsquedas de trayectorias de bombeos con mínimo coste y adaptación continua, con el objetivo de calcular esta condición de arranque inicial teniendo en cuenta la evolución completa a lo largo de un día y así conseguir una mayor optimización.

8.3.2 Criterios de optimización

Los criterios que se han adoptado para el método propuesto no son los únicos, pero se ha observado que son suficientes para lograr un buen funcionamiento:

- 1°. Para cada día, se calcula el volumen consumido el día anterior y se establece éste, multiplicado por el porcentaje de seguridad, como el volumen a bombear el día siguiente.
- 2°. Con el dato anterior se establece:

ZONA HORARIA	NIVEL DE ARRANQUE:	NIVEL DE PARADA:
Horas valle	Nivel correspondiente al volumen consumido el día anterior.	Volumen anterior + porcentaje seguridad.
Horas llanas	Nivel correspondiente al volumen previsto a consumir a partir de la última hora llana (aseguramos que el depósito va a tener un volumen de agua suficiente para cubrir el resto del día, incluidas las horas punta).	Nivel correspondiente al volumen previsto a consumir durante el resto del día + porcentaje de seguridad.
Horas punta	Nivel correspondiente al volumen previsto a consumir a partir de la última hora punta (aseguramos que el depósito va a tener un volumen de agua suficiente para cubrir el resto de horas).	Nivel correspondiente al volumen previsto a consumir durante las horas punta + porcentaje de seguridad.

Figura 8.1: Tabla para el arranque y parada según zona horaria en la optimización de bombeos

- 3°. Los niveles de arranque se comparan siempre con el nivel correspondiente al volumen mínimo de seguridad a contener en el depósito, escogiendo siempre el mayor de los dos, con lo que nos aseguramos el arranque de la bomba en caso de cualquier eventualidad. De la misma forma, los niveles de parada se comparan siempre con el nivel máximo del depósito, escogiendo siempre el mínimo de los dos para que el depósito nunca rebose.

Como primeras consecuencias de los criterios establecidos tenemos que:

- 1°. Tiene prioridad el llenado del depósito en horas valle.
- 2°. Sea cual sea el nivel del depósito al inicio del día, al final del día siempre se concluye con el depósito cerca del volumen mínimo de seguridad, al menos.
- 3°. Sólo se bombea el volumen necesario para el consumo del día (con el margen de seguridad adecuado).
- 4°. Si un día atípico el sistema deja el depósito más lleno de lo normal, se corrige al día siguiente con un menor bombeo.

Además el sistema incorpora las siguientes funciones implementadas por software sin necesidad de instrumentación adicional:

- Evaluación del caudal de las bombas.
- Detección de la pérdida de rendimiento de la bomba o falta de flujo de caudal.

- Seguridad ante posibles fallos de comunicaciones con la estación remota del bombeo (el sistema tiene implementado un algoritmo que calcula el tiempo de bombeo hasta el llenado del depósito).

La implementación de este sistema permite ahorros económicos importantes a lo largo del año (ver sección 8.3.5), garantiza el abastecimiento, y evita que mantengamos en el depósito más agua de la estrictamente necesaria, lo que cubre la mayor parte de los objetivos propuestos.

8.3.3 Pantallas y controles diseñados

Las pantallas que se han diseñado con el SCADA, Figura 8.2, tratan de fijar los criterios de funcionamiento y la estandarización del manejo desde la central de cualquier bombeo en general, y conseguir el objetivo de mostrar la máxima información al operador en tiempo real.

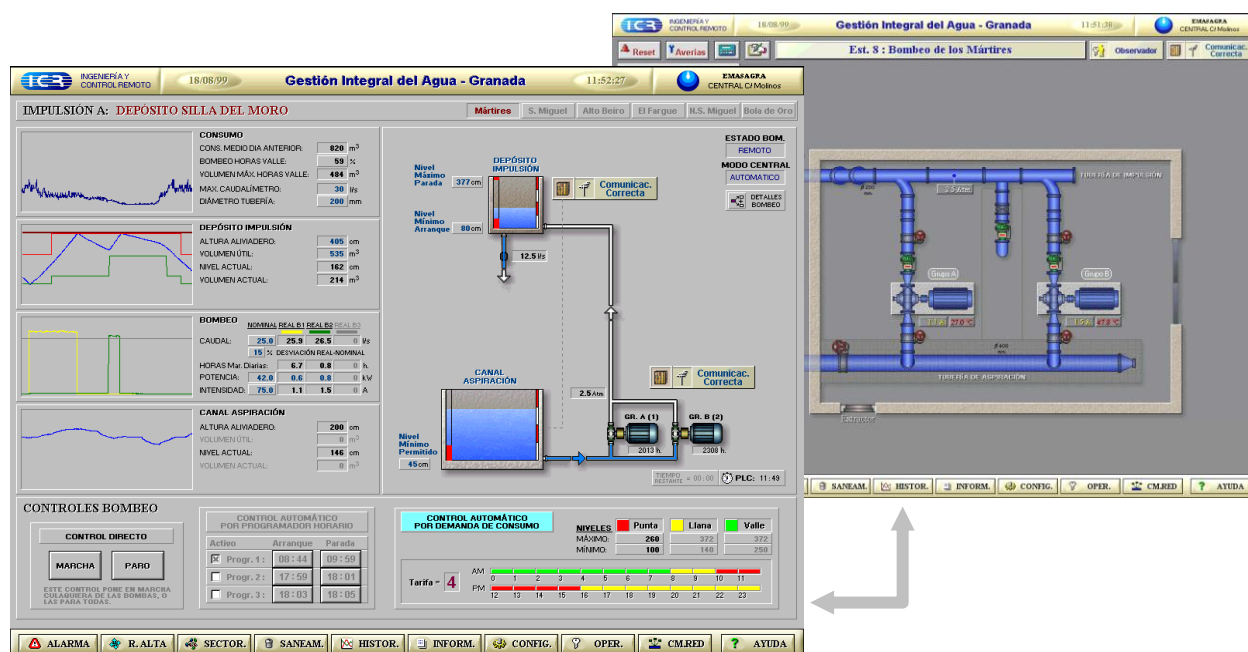


Figura 8.2: Pantallas para la optimización de bombeos

Desde la Central de Control se puede seleccionar tres modos de funcionamiento para cada bombeo:

- **Marcha manual:** permite poner en marcha o parar una bomba pulsando un solo botón.
- **Programa Horario:** permite arrancar una bomba según un programa horario establecido en la central (útil cuando las sondas de nivel no existen o están averiadas).
- **Programa por Demanda:** permite optimizar los costes de energía eléctrica aprovechando las tarifas eléctricas, ya que pondrá en marcha el bombeo dependiendo

de los parámetros que el operario le haya especificado para cada tramo horario. El sistema calcula también los caudales reales de las bombas y, en base a estos, el tiempo restante de bombeo necesario para alcanzar los volúmenes máximos de cada período, con objeto de evitar las paradas de bombeo por rotura en las comunicaciones.

Para conseguir el funcionamiento anterior hay que definir las tarifas existentes y hacer la asignación de tarifas a bombeos. Para ello se utilizan las ventanas que se muestran en la Figura 8.3.



Figura 8.3: Configuración y asignación de tarifas en la optimización de bombeos

Como ayuda al seguimiento del funcionamiento del bombeo y a la toma de decisiones se visualizan una serie de datos y tendencias de variables como muestra la Figura 8.4.

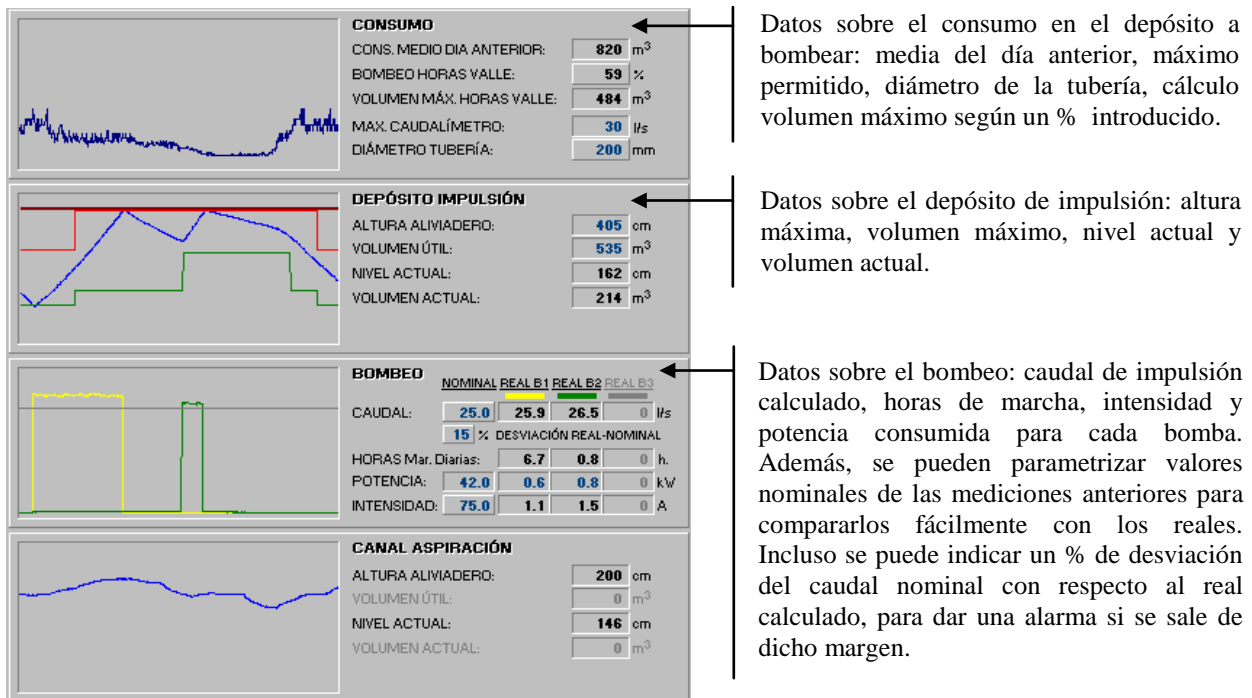


Figura 8.4: Información proporcionada en la pantalla de optimización de bombeos

Con toda esta información contenida en una única pantalla, el operador puede saber fácilmente cómo se ha comportado el bombeo durante el día anterior. Y para conocer el estado actual, también se proporciona una información gráfica del bombeo con datos en tiempo real de niveles, caudales de salida, estado de las comunicaciones y del funcionamiento de los PLCs implicados, tiempo restante de bombeo en caso de que fallen las comunicaciones, consignas de niveles máximos y mínimos permitidos tanto en el depósito de impulsión como en el canal o depósito de aspiración, presión en la impulsión, estado (remoto/local) y modo de funcionamiento (automático/manual), ver Figura 8.2.

8.3.4 Consideraciones sobre la implementación

Para hacer el sistema lo más robusto posible, el automatismo de arranque/parada de los bombeos está implementado localmente en cada PLC, de tal forma que si por cualquier causa la central no funciona correctamente, el estado de los bombeos no se ve afectado. Por lo tanto, lo único que se hace desde la central es parametrizar cada grupo de bombeo: modo de funcionamiento, asignación de tarifas eléctricas, y niveles máximos y mínimos de seguridad principalmente.

En cuanto a la distribución de la lógica del automatismo se debe de tener en cuenta que existen tres estaciones involucradas, estación de bombeo, estación central y estación depósito, en la mayoría de los casos interconectadas por radio. Por lo tanto, la información relativa al estado de funcionamiento y del nivel de la estación depósito necesaria para la estación bombeo a de pasar por la estación central. La seguridad y un funcionamiento óptimo en esta situación pasa por prever un algoritmo que calcule, según la expresión (8-1) el tiempo de marcha de la bomba (TRBFC) en caso de un fallo en las comunicaciones, con objeto de no parar el bombeo y que el depósito se siga llenando para así disponer de un mayor tiempo de respuesta.

$$TRBFC = \frac{(V_{max} - V_t)}{Q_s} \quad \begin{array}{l} V_{max}: \text{Volumen máximo del periodo horario en curso, m}^3 \\ V_t: \text{Volumen en el instante actual, m}^3 \\ Q_s: \text{Caudal de salida actual, m}^3/\text{h} \end{array} \quad (8-1)$$

El caudal de salida o caudal real (Q_s) se calculará cada vez que se produzca un bombeo según la expresión (8-2). Este caudal se presentará en la pantalla del sistema de control y se generará una alarma cuando se desvíe un % (configurable por el operador) respecto del caudal nominal proporcionado por el fabricante.

$$Q_s = \frac{(V_f - V_i) + (T_f - T_i)}{t} \tag{8-2}$$

V_f : Volumen final, en m³
 V_i : Volumen inicial, en m³
 T_f : Valor del totalizador de caudal de salida al final del período de bombeo, en m³
 T_i : Valor del totalizador de caudal de salida al inicio del período de bombeo, en m³
 t : Período de bombeo, en horas.

En cuanto a la programación realizada en los PLCs, a continuación se muestran los módulos de programa y sus variables resultantes, necesarias para generar la orden de marcha/paro del grupo de bombeo.

Módulo	Variables de salida
Obtención de parámetros tanto remotos (vía radio) como locales.	CD: Control Directo. MPCD: Marcha/Paro en Control Directo. MAUT: Modo Automático de funcionamiento. FCEN: Fallo en la Estación Central. FE9_10: Fallo de la Estación 9 (donde esta el depósito a bombear) para la 10 (donde se encuentra el bombeo). OMAG: Orden Mando Automático Grupo bombeo. ESTMR: Estación en Mando Remoto.
Control de seguridades	TRBFC: Tiempo Restante Bombeo por Fallo de Comunicaciones. BSNMIN: Boya de Seguridad por Nivel Mínimo del depósito a bombear. BSNMAX: Boya de Seguridad por Nivel Máximo del depósito a bombear. BSMINDA: Boya de Seguridad por nivel Mínimo del Depósito de Aspiración. AFDFG: Alarma Fallo Detector de Flujo del Grupo.
Comprobación de los tramos horarios activos y generación de la señal de arranque por programa horario.	MPH: Marcha por programa horario.
Generación de la señal de marcha según niveles y tarificación eléctrica.	MCNT: Marcha por niveles y tarifas eléctricas.

En la Figura 8.5, y utilizando la representación propia de los PLCs usados (diagrama de bloques funcionales de SIMATIC S5), se muestra gráficamente la condición de marcha/paro para cualquiera de los bombeos existentes en la red de distribución de agua de la ciudad de Granada, (expresión 8-3), en función de las variables definidas y generadas por los módulos anteriores.

$$\left(\begin{matrix} (CD \cdot MPCD) + \\ (\overline{MAUT} \cdot MPH \cdot \overline{CD}) + \\ (\overline{MAUT} \cdot MCNT \cdot \overline{CD} \cdot ((\overline{FCEN} \cdot \overline{FE9_10}) + (TRBFC > 0))) + \\ BSNMIN \end{matrix} \right) \cdot \left(\begin{matrix} \overline{BSMINDA} \cdot \overline{BSNMAX} \cdot \\ \overline{OMAG} \cdot \overline{ESTMR} \cdot \overline{AFDFG} \end{matrix} \right) \tag{8-3}$$

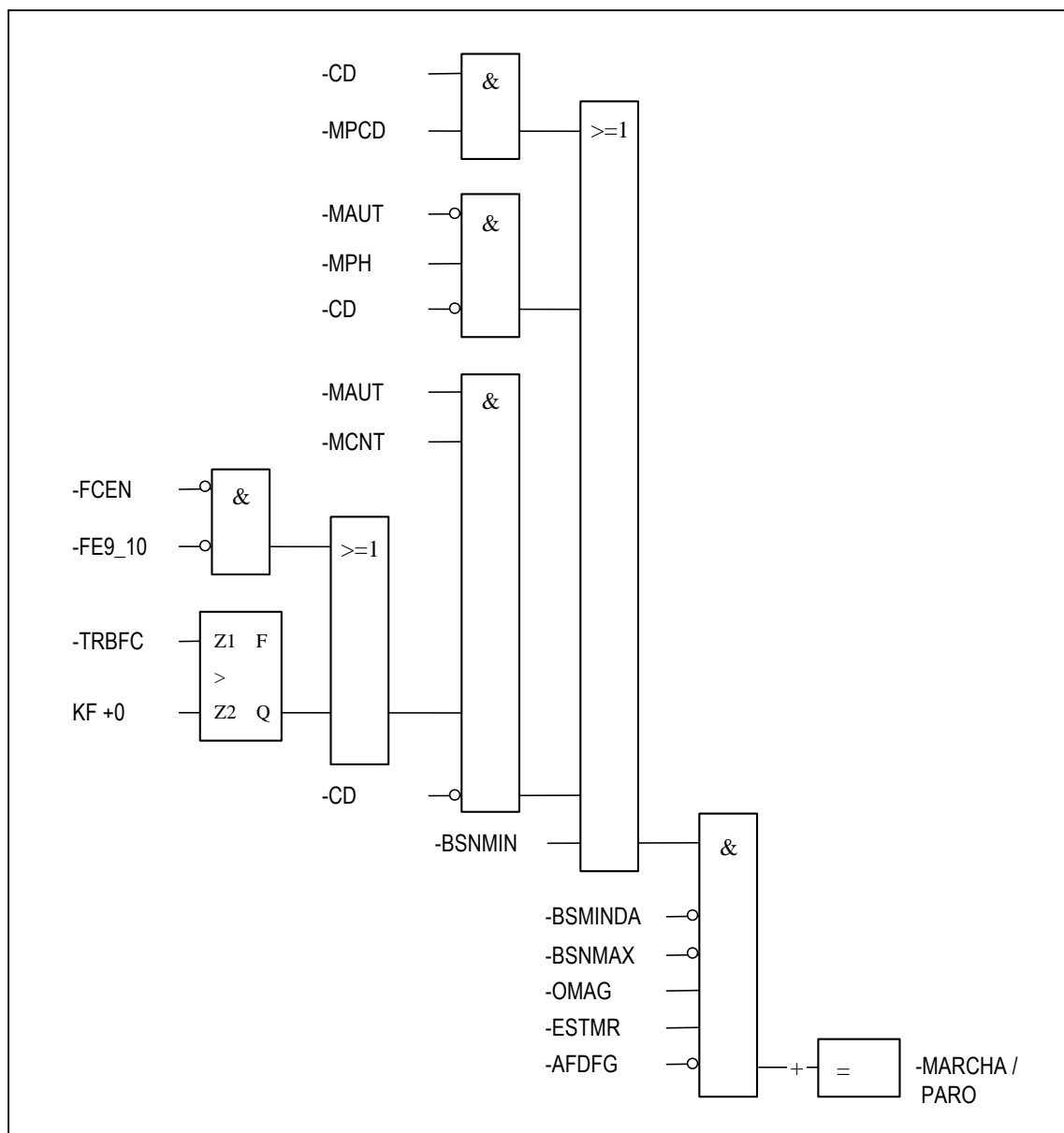


Figura 8.5: Diagrama de bloques funcionales SIMATIC S5 en la optimización de bombeos

8.3.5 Resultados obtenidos

Actualmente, este sistema está funcionando en 6 bombeos de la red de distribución de agua de la ciudad de Granada, mejorando el servicio en los siguientes aspectos:

- El control de los bombeos es uniforme y automático.
- Los operarios tienen más información para tomar decisiones.
- Se evitan paros innecesarios por posibles cortes transitorios en las comunicaciones.

- Se detectan un mayor número de fallos en el funcionamiento de las bombas.
- Se asegura el suministro las 24 horas del día.
- Se minimizan costes al intentar bombear toda el agua demandada en horas valle.

Como ejemplo del ahorro económico que se ha conseguido, en la Figura 8.6 se muestra una gráfica de la bonificación² aplicada a la facturación mensual en uno de los bombeos como consecuencia del mejor aprovechamiento de las horas valle y evitando las horas punta. En ella podemos observar que, desde la implantación a principios de 1998 del control automático en los bombeos con este sistema de optimización, se ha reducido el coste asociado al consumo de energía, ya que de estar en torno al 15% de bonificación ha pasado a más del 30% (que equivale aproximadamente a 1.100.000 ptas. anuales), lo cual significa un ahorro económico importante teniendo además en cuenta que se trata de 6 bombeos (Datos proporcionados por EMASAGRA).

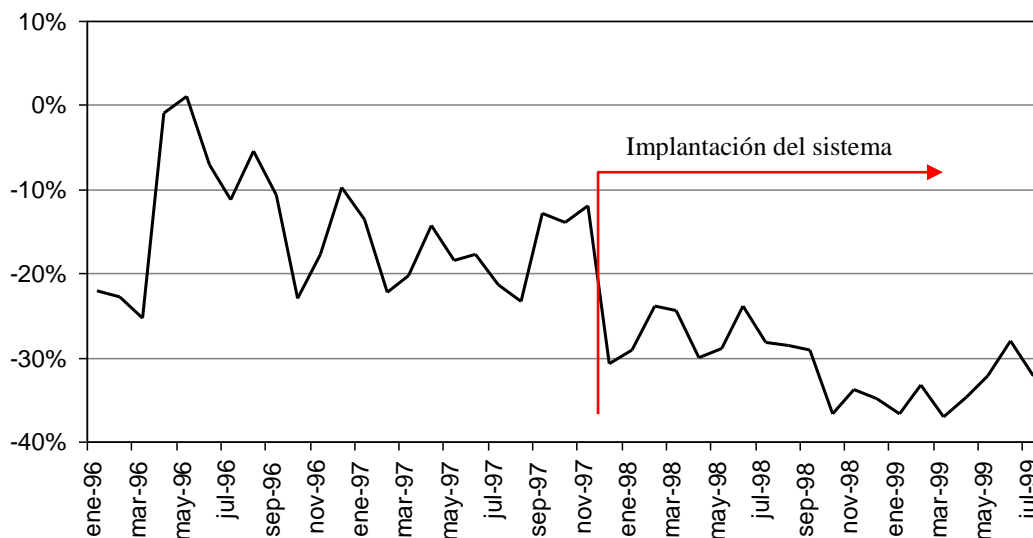


Figura 8.6: Gráfica del ahorro económico mediante la optimización en los grupos de bombeos

² Relación entre el ahorro por la utilización de la energía en horas valle y el coste de la energía total

8.4 Sectorización de la ciudad.

8.4.1 Introducción y objetivos

La justificada y creciente preocupación por mejorar el aprovechamiento del agua en los sistemas de abastecimiento exige aumentar la calidad y rendimiento de dichos sistemas [Vel-95]. En esta sección se presenta una técnica encaminada a controlar y reducir las pérdidas de agua e incrementar la eficiencia de los abastecimientos. Consiste en dividir la ciudad en zonas o sectores, según el área topográfica a que pertenezca, para realizar un análisis más exhaustivo del comportamiento de la red de distribución de agua (≈ 500 km. de tuberías en el caso de Granada), así como poder:

- Aislar un sector de forma inmediata en caso de que se detecte una fuga importante, mediante la actuación sobre la válvula motorizada de cabecera de dicho sector, y su puesta en carga una vez restablecida la avería de forma controlada.
- Detectar fugas, fraudes y consumos temporalmente anormales.
- Computar los caudales y presiones en informes diarios.
- Reducir la presión en aquellos sectores que por la cota a la que se encuentran necesitan un ajuste mayor que el resto.
- Minimizar las pérdidas nocturnas conmutando automáticamente a reductoras de presión instaladas en alguno de estos sectores.

8.4.2 Pantallas y controles diseñados

La pantalla principal (ver Figura 8.7) muestra de forma gráfica la división en sectores, las arterias principales del abastecimiento, los caudales, las posiciones de válvulas y las presiones que se miden, así como la detección de eventos en un sector concreto.

Los datos se obtienen utilizando el sistema distribuido de comunicaciones vía radio, descrito en el capítulo 4, que proporciona en tiempo real la información suministrada por cada sector, y que consta de:

- **Mediciones:** caudales, presiones y posiciones de válvulas/reductoras en todos los sectores, y en algunos de ellos otros parámetros como cloro y turbidez.
- **Eventos:** fallo del caudalímetro, fallo de tensión, fallo de maniobra de la válvula/reductora, apertura de puerta, alarma intrusos, etc.

- **Contadores:** se toman pulsos y se calcula en cada estación remota un totalizador diario y otro continuo.
- **Actuaciones:** el sistema también puede comandar la válvula que cierra, abre o conmuta (en algunos de los sectores) a otra válvula reductora de presión.

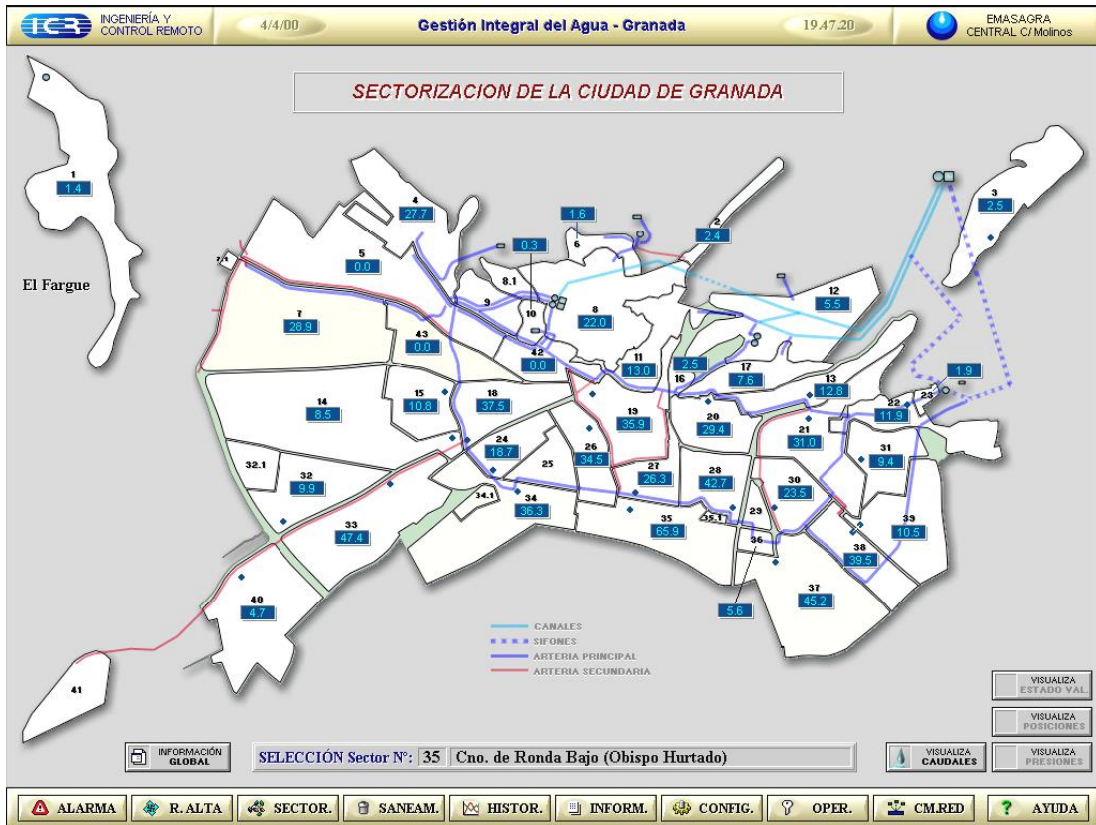


Figura 8.7: Pantalla principal para la sectorización de la ciudad de Granada

Cada sector individual se puede visualizar y controlar desde la central con una pantalla idéntica para todos los sectores, pero con las referencias, gráficos, fotografía, y datos del sector elegido, como se muestra en la Figura 8.8.

Desde esta pantalla se pueden acceder a los sectores que rodean al sector que se está visualizando, se puede reconocer la alarma de puerta abierta, ver si hay fallos de tensión o del caudalímetro, se muestran los valores diario y continuo del totalizador, se puede hacer un reset del totalizador continuo (reflejando la fecha y la hora de dicho reset), se muestran también los valores analógicos de consumo y presión, así como el esquema de las válvulas de cabecera del sector. También se pueden configurar parámetros como el nombre del sector, y los valores de alarmas de mínimo y máximo del caudal y de la presión.

La ciudad se ha dividido al final en 45 sectores, de los cuales:

- 24 se controlan con 22 estaciones remotas especialmente diseñadas para este fin,
- 15 se corresponden con estaciones de la red de alta, y
- 6 aportan datos utilizando sistemas de adquisición portátiles (*logger*). Para incorporar los consumos obtenidos de esta forma, se ha implementado un programa que a partir del fichero generado por el *logger*, actualiza la base de datos, y calcula los consumos diarios para incorporarlos en las columnas correspondientes del informe.

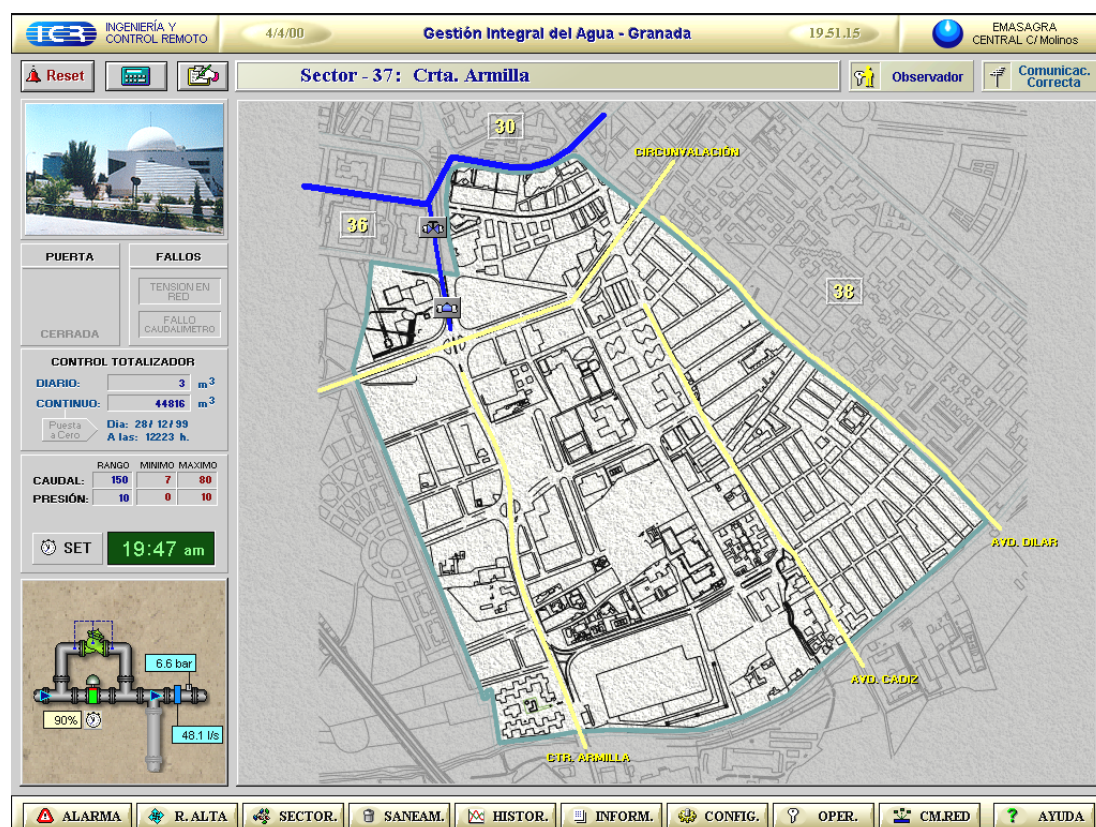


Figura 8.8: Detalles de un sector particular de la ciudad de Granada

Las ventajas más importantes que se consiguen con esta interfaz son:

- Disponibilidad de los datos de los sectores en tiempo real.
- Uniformidad en el tratamiento de la información.
- Actuación manual y automatizada desde la central.
- Cálculos de los consumos diarios para cada sector de forma automática. Esta información la podemos visualizar de varias formas:

1. Pulsando sobre el icono correspondiente (en forma de caudalímetro), se muestran las curvas de consumo y de presión de las últimas 24 horas del sector que se está visualizando.
2. Si se desea hacer un estudio más exhaustivo de la evolución de las variables se puede utilizar el módulo de visualización de históricos que incorpora la aplicación.
3. Y por último, también se puede acceder al informe de consumos diarios (en formato EXCEL) de cualquier mes anterior o del mes en curso.

8.4.3 Resultados obtenidos

- Se ha diseñado un interfaz de operador sencillo y a la vez funcional.
- Se han detectado numerosas fugas, como se muestra en la Figura 8.9, que han supuesto un ahorro económico y de agua muy importante (sólo en el año 1999, y gracias a este sistema, se han ahorrado aproximadamente 2 millones de m³ de agua, el equivalente a casi un mes de consumo o más exactamente un 6% del consumo total anual). A esto también hay que añadirle la reducción en cuanto a la aplicación de planes costosos de detección directa de fugas (mediante micrófonos sonda, geófonos acústicos o electroacústicos, equipos de correlación acústica, etc.) [Fol-95].
- Se han detectado fraudes o consumos no contabilizados que han supuesto también un ahorro económico significativo.
- Se realiza un seguimiento de saltos en el consumo (regadíos, comercios, etc.), así como un control más preciso de la presión.
- Permite un ajuste más fino de la presión según la zona (cota) de la ciudad. Como consecuencia inmediata se consigue un ahorro en el consumo y en pérdidas por fugas, ya que menor presión es directamente proporcional a menores pérdidas en las fugas existentes [Lam-94, Ger-95, Sav-95]. En la Figura 8.10 se puede observar el efecto que produce en la presión y en el consumo el cierre automático de la válvula principal de suministro al sector y la entrada en funcionamiento de una válvula reductora de presión asociada con la anterior (ver Figura 8.8).
- Con los datos que proporciona este sistema y las posibilidades de control que se ofrece sobre la red de distribución de agua de la ciudad de Granada, está permitiendo poder aplicar, calibrar y testear una nueva metodología para la construcción de controladores neuro-difusos con capacidad de aprendizaje y adaptación continua, y cuyos detalles se dan en la siguiente sección.

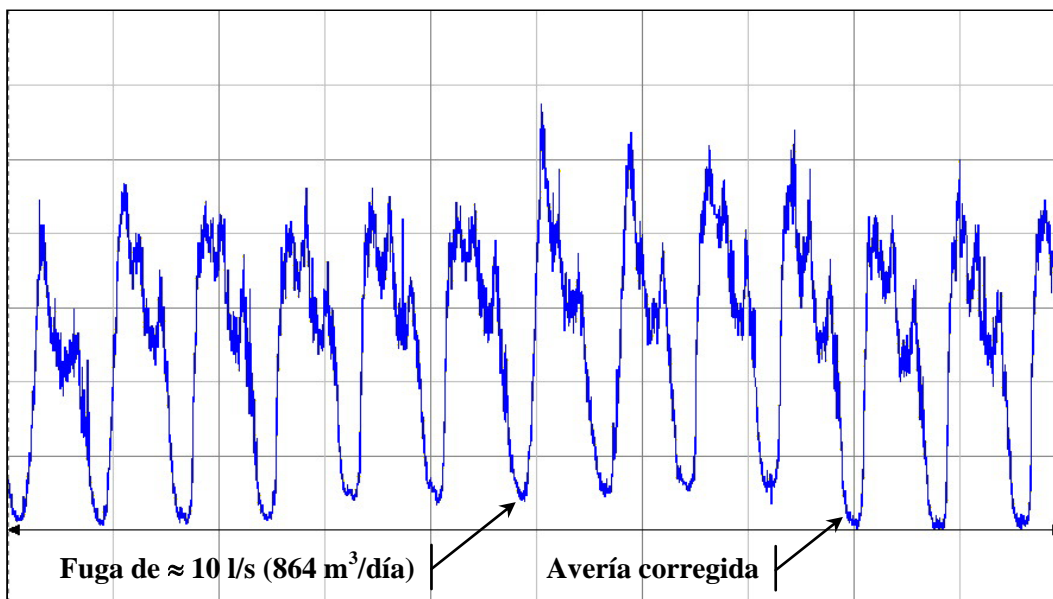


Figura 8.9: Ejemplo de detección de fuga

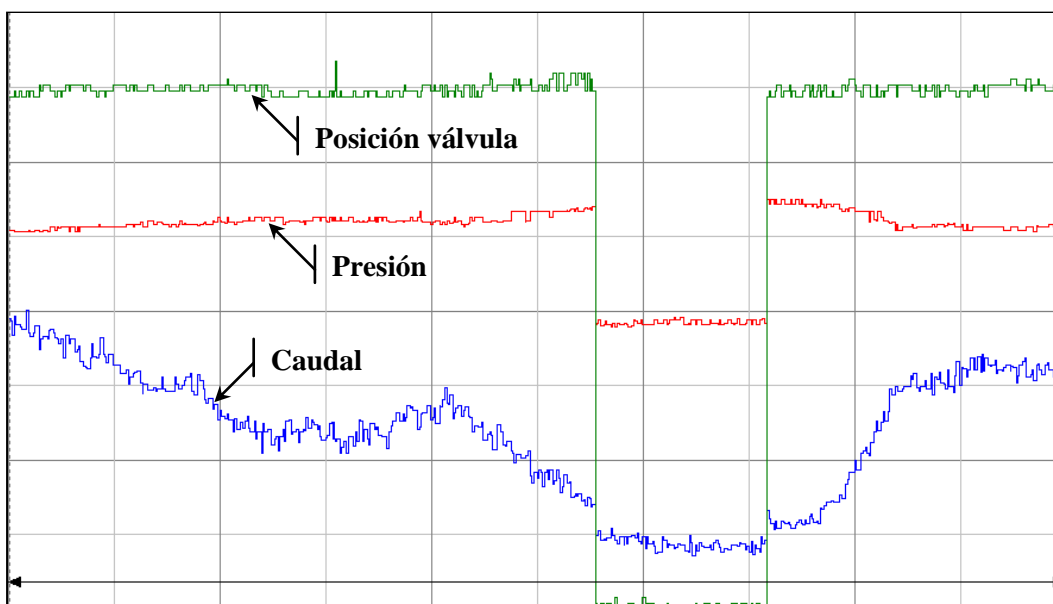


Figura 8.10: Efecto de reducción del consumo y de la presión en un sector de la ciudad

8.4.4 Líneas actuales y futuras derivadas del sistema

Uno de los problemas más relevantes en la construcción de controladores de sistemas completos, y en particular de sistemas de distribución de agua, es el diseño de modelos fiables del sistema. En la actualidad, es conocido que la eficiencia del controlador depende en gran medida de la exactitud del modelo matemático utilizado para un conjunto heterogéneo de posibles estados del mismo.

La simulación o estimación del estado del sistema real precisa de la obtención y calibración de un modelo previo de la planta. Dicho modelo debe comportarse como el sistema real, para de esta forma facilitar la localización e identificación de medidas erróneas, estudiar los efectos que producen cambios en las variables de entrada del proceso, ayudar al entrenamiento y toma de decisiones de los operadores humanos, y predecir posibles estados futuros del sistema.

Una de las tareas más laboriosas en la construcción de uno de estos modelos es el calibrado del amplio número de parámetros que involucra su diseño [Dat-94]. En nuestro caso, sólo el modelado de las presiones y caudales en los distintos puntos de la red conlleva la solución de sistemas complejos de ecuaciones no lineales y con elevado número de variables. Las técnicas basadas en modelos analíticos [Rah-80, Wal-83], tienen una restringida utilidad en sistemas reales, puesto que tan solo son eficientes en redes de reducido tamaño. Las tendencias más prometedoras en el campo de la modelización hacen uso de técnicas no convencionales como pueden ser los AGs, las RNAs y la lógica difusa, descritas en el capítulo 2 de esta memoria dedicado a fundamentos.

En este sentido, utilizando los datos y la capacidad de actuación del sistema de telecontrol que hemos desarrollado, se está calibrando un modelo simplificado de la red de distribución de agua de Granada, mediante un sistema neuro-difuso [Pom-99], para usarlo en lugar de la planta real, y así poder ajustar un controlador difuso con capacidad de aprendizaje y adaptación continua. El control de una red de distribución de agua es una actividad compleja, ya que posee una gran cantidad de variables de entrada y salida generalmente interrelacionadas, en general no existe un único criterio de decisión y estos son expresados en términos lingüísticos (susceptibles por tanto de ser representados mediante lógica difusa), y es además un sistema no-lineal.

El objetivo final es obtener un sistema que en una primera fase (y en la que nos encontramos actualmente) pueda servir de ayuda a la predicción y la toma de decisiones del operador humano, y que en una fase posterior podría formar parte de un controlador completamente autónomo y distribuido de redes de abastecimiento de agua de ciudades.

8.5 Control de reductoras.

8.5.1 Introducción y objetivos

La red de suministro de la ciudad de Granada, debido a los desniveles que presenta, requiere diferenciar entre la red de alta y la red de baja con las pertinentes válvulas reductoras de presión (VRP). Actualmente existen cuatro estaciones para el control de la presión en la arteria principal. Cada una de ellas incorpora dos VRP, una para caudales altos (régimen diurno, de 400 mm de diámetro) y otra para caudales bajos (régimen nocturno, de 150 mm de diámetro). Una VRP mantiene la presión constante a su salida (aguas abajo) abriéndose ante caudales elevados y cerrándose para caudales pequeños. Estas válvulas (URA-CONTROL) son de control hidráulico, están accionadas por un diafragma, utilizan un servomotor que actúa girando un tornillo (piloto) para la regulación precisa de la presión, y disponen de una serie de electroválvulas de apertura y cierre gobernadas por un PLC para el control automático.

El control de las VRP debe permitir:

- Programar una reducción de presiones en función de las horas del día, lo que puede producir un ahorro importante de agua por la noche, sobre todo en épocas de sequía. Es bien conocido que los porcentajes de fugas se incrementan considerablemente con la presencia de altas presiones en la red, y que la relación entre el nivel de fugas y la presión no es lineal [Lam-94].
- Conmutar automáticamente entre las dos reductoras que forman el sistema, para así aprovechar mejor el régimen de funcionamiento nominal de estas, y conseguir que la regulación se adapte a la diferencia entre los caudales diurnos y nocturnos.

La posibilidad de controlar desde la central (vía radio) la presión en la arteria principal de la red de baja mediante esta automatización de las VRP, nos permite además, junto con la división de la red en zonas según el área topográfica a que pertenezca, avanzar en la optimización de este proceso, así como ensayar y aplicar nuevas técnicas de modelización y control, ya indicadas en la sección anterior. Concretamente, se ha observado que uno de los puntos más críticos y que afectan notablemente al funcionamiento de todo el sistema son los valores de consignas de presión que se establezcan para estas válvulas. Por tanto, otro objetivo que se persigue, es la determinación automática de esas consignas para optimizar aun más el consumo en la ciudad.

8.5.2 Pantallas y controles diseñados

En la Figura 8.11 se muestra la pantalla diseñada para supervisar y controlar desde la central las estaciones reductoras de presión.

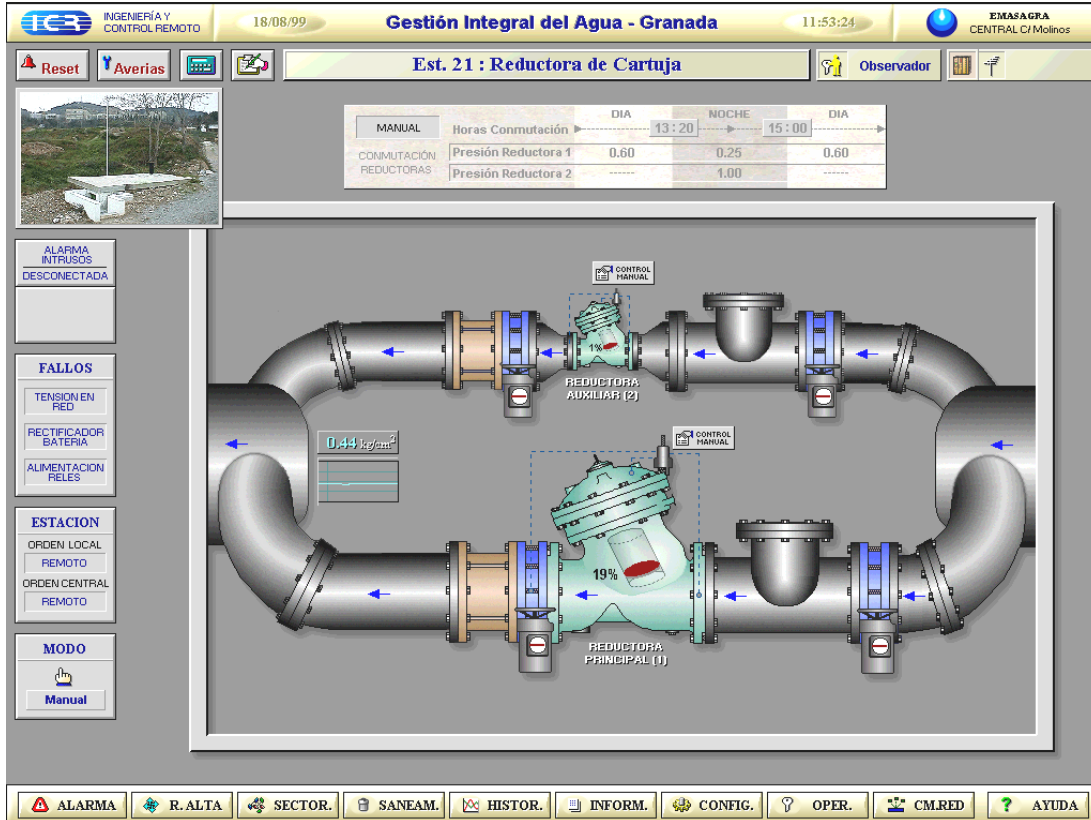


Figura 8.11: Pantalla general para las estaciones reductoras de presión

La válvula reductora de presión junto con sus detalles de visualización y control se puede apreciar en la

Figura 8.12:

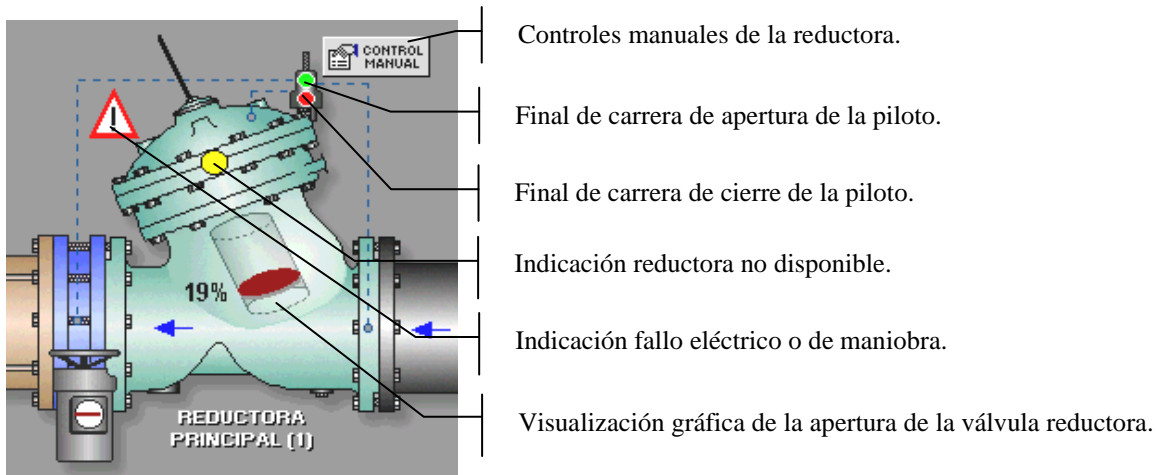


Figura 8.12: Detalles de la VRP

El control sobre las estaciones reductoras se puede efectuar tanto en modo manual como en modo automático. En ambos modos se puede activar o desactivar la regulación y cambiar la consigna y la histéresis de presión de las dos VRP (principal y auxiliar).

Estas modificaciones se hacen sobre la ventana de control, (ver Figura 8.13), que aparece pulsando sobre el botón etiquetado como *CONTROL MANUAL*.



Consigna de presión aguas a bajo.

Histéresis para la regulación.

Botones para parar y activar la regulación de la presión.

Visualización gráfica de la posición de la válvula reductora y de la presión, así como de sus niveles de alarmas.

Figura 8.13: Ventana de control manual para las VRP

Hay que tener presente que cuando la regulación está activada el PLC intenta mantener la presión de salida (aguas a bajo) entre los siguientes márgenes:

$$PRESIÓN SALIDA = CONSIGNA PRESIÓN \pm HISTÉRESIS$$

En modo manual, no se hace la conmutación de las válvulas reductoras y en la ventana de control anterior se activan los botones y textos relacionados con este modo. Estos botones se usan para abrir o cerrar completamente la VRP. En este modo manual se pueden aplicar también pulsos pequeños sobre la piloto para que aumente o disminuya la presión en pequeños incrementos.

En modo automático, la conmutación entre las dos VRP se hace de forma automática según una programación horaria y unas presiones establecidas mediante el control de la Figura 8.14, que sólo se activa en este modo.

		DIA	NOCHE	DIA
AUTOMÁTICA	Horas Conmutación	01:30	07:00	
CONMUTACIÓN	Presión Reductora 1	0.70	0.25	0.70
REDUCTORAS	Presión Reductora 2	-----	0.40	-----

Figura 8.14: Ventana de control automático para las VRP

El control automático, y según la configuración que se muestra en la Figura 8.14, consiste en lo siguiente:

“Durante el día estará regulando la reductora principal (1) manteniendo la presión a 0,70 kg/cm². Cuando llega la 1:30 h. de la noche se produce la conmutación automática a la reductora auxiliar (2) y regulará la presión a 0.40 kg/cm², hasta las 7:00 h. de la mañana que se volverá a conmutar a la reductora 1”.

Como es lógico, tanto los valores de consigna de presión como las horas de las conmutaciones se pueden configurar por el operario con el permiso adecuado (supervisor).

8.5.3 Consideraciones sobre la implementación

En primer lugar hay que destacar que la manipulación de una VRP, como las existentes en la red de distribución de la ciudad de Granada, es una tarea muy crítica, ya que puede ocasionar trastornos y daños muy cuantiosos si se abre o cierra inadecuadamente. Por esta razón la implementación de este automatismo ha de ser muy fiable y robusto, y para ello:

- Se ha utilizado un PLC como sistema de control distribuido, encargado de ejecutar localmente los algoritmos de regulación y conmutación, así como supervisar las ordenes tanto manuales como automáticas.
- En cuanto a la programación se han añadido muchas seguridades: filtrados de señales analógicas para eliminar ruidos, control fino por pulsos de duración variable, tiempos de espera y control de estabilidad en cada maniobra, y detección de fallos de maniobra.
- El algoritmo propuesto para la conmutación entre las dos VRP está diseñado para que en caso de que se produzca algún problema (fallo eléctrico o de programa), la válvula reductora principal se quede siempre en una posición adecuada para que hidráulicamente el sistema siga funcionando en unas condiciones factibles. Es decir, se consigue una conmutación más segura si se utiliza un automatismo híbrido, en el cual el control mediante PLC se apoya además en el comportamiento hidráulico del sistema.

En cuanto a la implementación realizada en los autómatas, en la Figura 8.15 se muestra un diagrama de bloques de la parte dedicada a la manipulación, conmutación y regulación de las VRP.

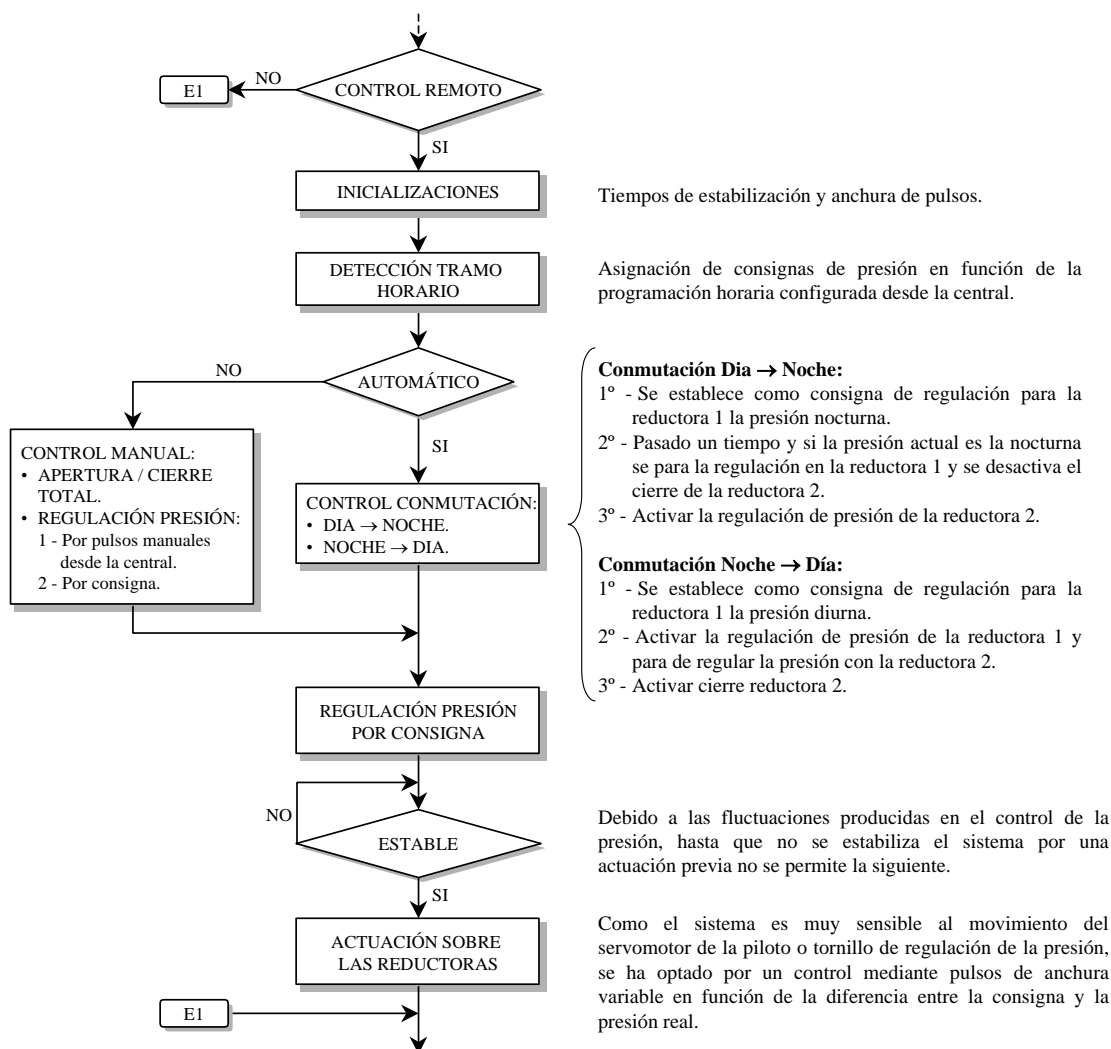


Figura 8.15: Diagrama de bloques para el control de las VRP

El comportamiento hidráulico en el que se basa el módulo de control de conmutación del diagrama de bloques de la figura anterior es el siguiente:

- En la conmutación día → noche, después del paso 2º, la VRP principal (1) se cerrará sola (por efecto hidráulico) puesto que la presión fijada de la VRP pequeña (2) es superior que la de la VRP 1.
- En la conmutación noche → día, después del paso 2º la VRP 2 se cerrará sola (también por efecto hidráulico) al ser más grande, y por tanto dominar, la presión de la VRP 1.

8.5.4 Resultados obtenidos

- Se ha diseñado un interfaz de operador sencillo y a la vez funcional.
- Se ha podido observar y comparar, por primera vez en la red de distribución de agua de Granada, la relación entre el caudal, la presión y la posición de la VRP para distintos regímenes de funcionamiento, en los puntos principales de inyección de la arteria principal.
- Se puede controlar y manipular de forma segura un sistema muy crítico dentro una red de distribución de agua: la reducción de la presión en la arteria principal mediante VRP.
- Al poder manipular desde el centro de control las presiones de salida de todas las reductoras, se puede conseguir manipular la presión (subir o bajar) en la arteria principal de forma gradual y segura.
- Dado que se pueden comandar desde el centro de control todos los dispositivos, se aprovecha la disponibilidad de las dos reductoras, regulando una u otra en función del caudal que demanda en cada momento la ciudad. Antes de la implantación del sistema, siempre funcionaba la reductora mayor, incluso fuera de su régimen nominal, con el consiguiente deterioro en la VRP que esto provocaba.
- Si se detecta una rotura importante, se puede cerrar automáticamente la válvula reductora asociada, con el consiguiente ahorro en costes y tiempo en la atención de la avería, así como en seguridad.
- Se ofrecen las herramientas necesarias para poder conseguir un ahorro en el consumo y en las pérdidas por fugas y roturas mediante bajadas en la presión de la red.
- Con los datos y el control que proporciona este sistema sobre la red de distribución de agua de la ciudad de Granada, se amplían las posibilidades de seguir trabajando para aplicar nuevos algoritmos que consigan una mayor optimización en la misma línea indicada en la sección 8.4.4.

8.6 Predicción del volumen de agua diario.

8.6.1 Introducción y objetivos

Como se indicó en el Capítulo 3, el hecho de la existencia de canales abiertos en la red de distribución de Granada hace que se plantee un problema de retardo en las conducciones. Concretamente, desde los embalses hasta la ETAP, los tiempos de retardo son de unas dos horas, y desde la ETAP hacia los depósitos de distribución, los tiempos de retardo son de unos 40 minutos, dependiendo siempre de los caudales en servicio.

La capacidad total de almacenamiento en depósitos es de unos 45.000 m³ mientras que la demanda media es de unos 90.000 m³. Esta deficiencia de capacidad de almacenamiento, junto con el tiempo de retardo en los canales de abastecimiento y distribución, hace que se necesite prestar una atención permanente al estado de los depósitos para poder adelantarse lo suficiente a las fluctuaciones de demanda y solicitar a los embalses los cambios de caudal necesarios, evitando así, el llenado o vaciado excesivo de los depósitos.

Por estos motivos se consideró conveniente:

- Disponer de un sistema de predicción de la demanda del día siguiente, con objeto de solicitar los volúmenes de agua necesarios a los embalses en una sola ocasión al día, y
- Analizar la evolución diaria para, con los volúmenes almacenados y caudales demandados en tiempo real y en determinados momentos del día, realizar las correcciones pertinentes en la solicitud de caudales a los embalses (los operadores de los embalses sólo están disponibles de 8:00 a 14:00 en días laborables exclusivamente).

8.6.2 Análisis de la demanda y selección de las variables

La demanda diaria de agua de una población tiene fluctuaciones normales que vienen provocadas por diversos motivos como pueden ser la variabilidad de la población, la climatología, las festividades o simplemente los diferentes días de la semana [Gis-93]. También posee una importante componente de aleatoriedad debido a averías, eventos deportivos o turísticos, y otras situaciones no determinadas.

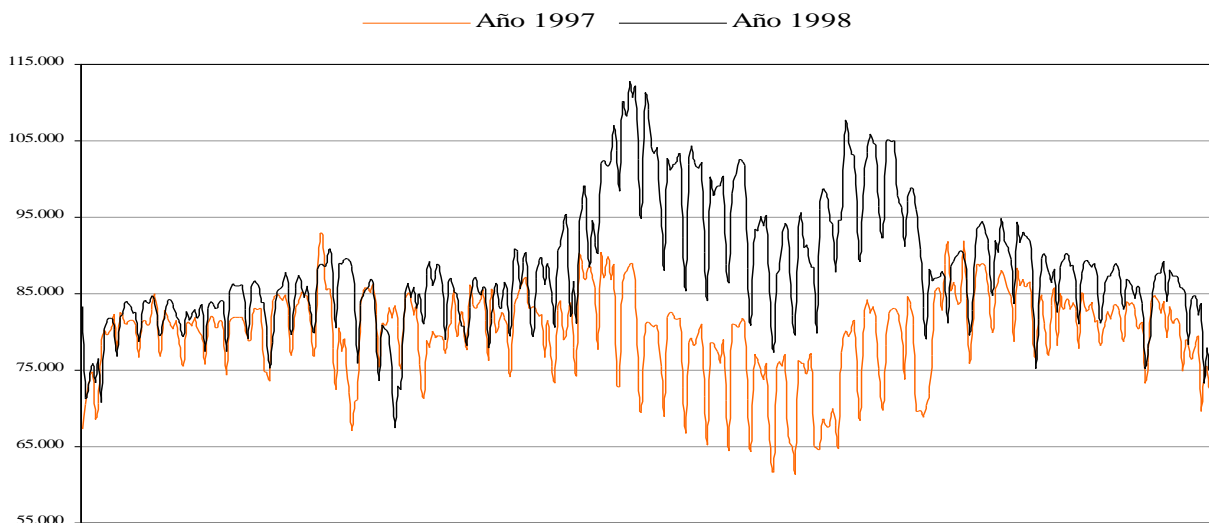


Figura 8.16: Consumo anual de la ciudad durante dos años ($\text{m}^3/\text{día}$)

En la Figura 8.16 se pueden observar todas estas fluctuaciones: se aprecia claramente la incidencia de los festivos con un valle muy pronunciado, en el período de verano un crecimiento del consumo al principio como consecuencia del incremento de temperatura, un valle importante con motivo de las vacaciones (mes de agosto), y otro crecimiento al comenzar el período lectivo (una gran parte de la población de Granada es universitaria). El desplazamiento entre las dos curvas (años 1997 y 1998) se debe a la relajación de las costumbres después del período de sequía y a la incorporación de otros municipios a la red de abastecimiento en estos años.

En los servicios de explotación de una red concreta se afrontan estos problemas de distinta forma y de una manera un poco particular, lo que provoca una importante componente de incertidumbre en la previsión del volumen demandado.

Una vez analizado el problema, y dado que se va a utilizar un procedimiento basado en RNAs, se plantea la necesidad de seleccionar las variables que pueden dar información del volumen a consumir. En primer lugar es obvia la dependencia de la demanda con los días de la semana (domingos y festivos parecen tener la misma incidencia). En segundo lugar se pensó que la estacionalidad podría influir (los meses de Julio, Agosto y Septiembre, suelen tener un consumo diferente al resto de los meses), y se decidió elegir como segunda variable el mes en curso. Sin embargo, los resultados obtenidos indicaron que para periodos de entrenamiento no

superiores a un año, esta variable de entrada no aportaba ninguna mejora en los resultados, así que finalmente el mes no se ha incluido como entrada en el proceso de aprendizaje de la red.

También se ha estudiado la relación entre las temperaturas diarias y los caudales; sin embargo, como la incidencia de la temperatura está implícita en el consumo, se decidió no utilizar tampoco esta variable directamente.

En este punto se nos plantea un problema respecto a la selección de los datos de entrada: si tomamos éstos con su valor absoluto, sucederá que para temporadas en las que la tendencia de consumos sea al alza o a la baja respecto a los datos de entrenamiento, al no poder extrapolar, la herramienta puede perder utilidad. Por éste motivo, se ha decidido normalizar los datos de forma que se evite esta dependencia, de ahí que se utilice el caudal del día actual respecto al caudal promedio de los 7 días anteriores. Así obtenemos como variable incógnita o de salida:

$$Razón(i) = \frac{Q(i)}{\sum_{j=i-8}^{i-1} Q(j) / 7} \quad Q(i) = \text{Caudal total del día.} \quad (8-4)$$

Por otra parte, como resultado de una serie de pruebas realizadas, se consideró necesario introducir otra variable más que tuviese en cuenta la tendencia de los caudales en los días previos:

$$Tendencia(i) = \frac{\sum_{j=i-8}^i Q(j) / 7}{\sum_{j=i-15}^{i-1} Q(j) / 14} \quad (8-5)$$

Esta variable, expresión (8-5), indica la tendencia de la demanda (ascendente antes del verano y descendente después del verano). No obstante, también se ha considerado necesario agregar una tercera variable de entrada, la pendiente de la tendencia anterior, para mostrarle a la red si dicha tendencia es positiva o negativa.

En resumen, las variables del problema las dividimos en dos conjuntos:

- Variables de entrada a la red:
 1. **D**, día de la semana, de domingo a sábado codificados del 1 al 7, (los festivos se han marcado también como 1),
 2. **T**, tendencia, relación entre la media de los 7 días anteriores con respecto a la media de los 14 días anteriores al día problema,

3. **P**, pendiente, +1 para tendencias crecientes y -1 para tendencias decrecientes.
- Resultado obtenido con la red:
 1. **R**, razón de demanda diaria, relación entre el caudal del día a predecir con respecto a los 7 días anteriores.

El resultado obtenido, R, permite calcular el caudal previsto para el día siguiente sin más que multiplicarlo por la media de los 7 días previos.

8.6.3 Elección de la herramienta de cálculo

Existen múltiples métodos para la resolución de este problema tales como el análisis de series temporales y modelos derivados del ARIMA, análisis factorial, o técnicas de alisamiento exponencial. En nuestro caso, como herramienta, se han escogido las Redes Neuronales [Wid-90, Hil-95].

En diversas ocasiones se han realizado estudios de previsión de caudales basados en métodos deterministas o estadísticos que intentan extrapolar soluciones a partir de los datos obtenidos históricamente. En nuestro caso se pretende encontrar las soluciones, más como resultado de un aprendizaje previo del comportamiento de los consumos diarios, que de la evolución estadística de los datos. Con los métodos de predicción utilizados se obtienen resultados más empíricos que deterministas, tal y como haría un operador experimentado.

Utilizando las variables seleccionadas en la sección anterior se han probado distintos tipos de redes neuronales usando la herramienta MATLAB, y se ha comprobado que la que mejores resultados proporciona es la red *backpropagation* (BPN), una de las más utilizadas hoy en día, con aprendizaje supervisado y conexiones hacia adelante (*feedforward*) [Rum-88].

La principal característica de esta red BPN consiste en su capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para “aprender” la relación que existe entre un conjunto de patrones dados como ejemplo y sus salidas correspondientes. Así se puede aplicar esa misma relación, después del entrenamiento, a nuevos valores de entrada que el sistema no ha visto nunca, dando una salida correcta si la nueva entrada es parecida a las presentadas durante el aprendizaje. A esta característica se le denomina generalización.

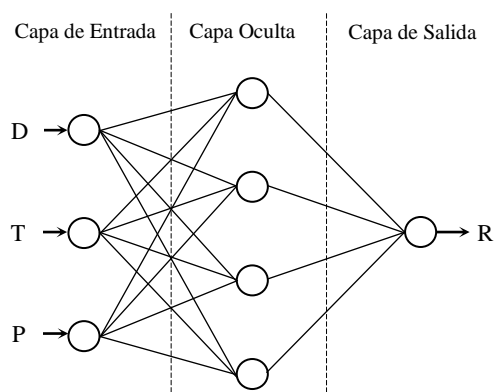


Figura 8.17: Esquema de la red neuronal utilizada para la predicción

En nuestro caso, como el problema de predicción de demandas se reduce a un problema de clasificación de patrones (a un conjunto de entradas corresponde una clase de salidas), se ha empleado un modelo de red neuronal del tipo BPN, con tres neuronas en la capa de entrada, cuatro neuronas en la capa oculta y una neurona en la capa de salida [Val-99]. En la Figura 8.17 se puede ver el esquema de la red utilizada para obtener los resultados que se comentan en la siguiente sección.

En el método propuesto se ha seguido el siguiente procedimiento, que se ha de aplicar para cada día:

1. Selección de los datos correspondientes a un periodo de días anteriores al día problema.
2. Inicialización y entrenamiento de la red varias veces, y selección de los pesos y umbrales (W_i , B_i) para la mejor solución de los entrenamientos (menor error).
3. Utilización de la red ya entrenada para que nos proporcione el valor de salida.
4. Con el resultado obtenido, razón de demanda, se multiplica por la media de los últimos 7 días anteriores para calcular el volumen que se va a consumir durante el día.

Respecto al periodo o número de días de entrenamiento de la red (paso 1 del procedimiento), se han probado cuatro opciones: tres meses, seis meses, un año y dos años. De estas opciones, se comprobó que la de tres meses y la de un año proporcionaban una desviación típica menor que las otras dos. Y de las dos mejores opciones se ha elegido la de un año por ser la que integra todas las posibles fluctuaciones del caudal.

8.6.4 Resultados obtenidos

Para comprobar la bondad de la herramienta utilizada se ha simulado, usando el procedimiento descrito en la sección anterior, la predicción del caudal para cada día de un periodo de un año (1998), del cual se conocen los caudales reales. En la Figura 8.18 se puede observar la curva resultado de la simulación respecto de los datos reales. Como se puede ver en la gráfica, el seguimiento de los domingos y festivos (los valles más profundos) es muy ajustado, y además, la tendencia correspondiente a las estaciones del año, así como los periodos vacacionales (verano, semana santa y otros) quedan bien reflejados en la predicción.

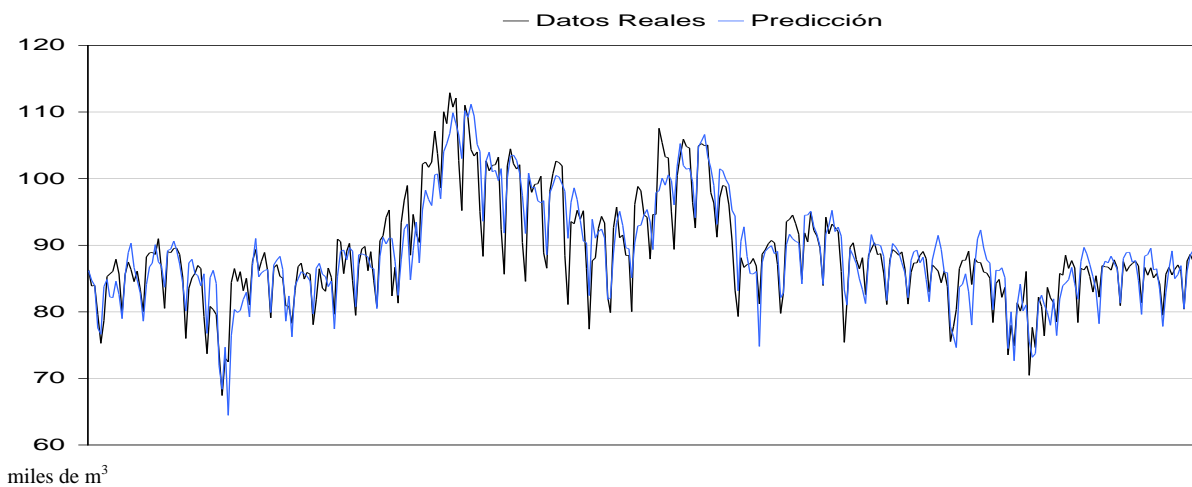


Figura 8.18: Comparativa entre datos reales y predicciones para un año completo (1998)

Como resumen queremos resaltar que la desviación típica de este conjunto de datos es del 3,71 %. Es decir, el 68% de los resultados está por debajo de este error, como se ilustra en la Figura 8.19.

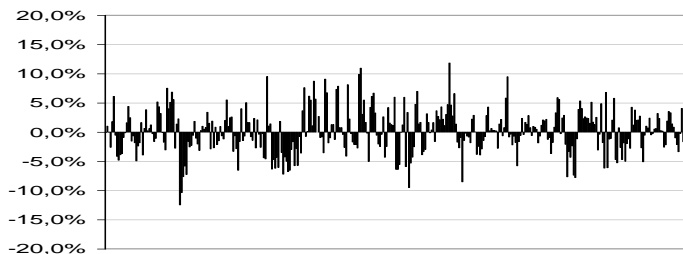


Figura 8.19: Error entre el caudal calculado y el real para el año 1998

8.7 Regulación automática de los depósitos de la red de alta.

8.7.1 Introducción y objetivos

La optimización global de los sistemas de abastecimiento de agua presenta diversas facetas [Ali-98]. Por una parte están los problemas relativos a la optimización de los recursos en las estaciones de tratamiento de agua potable, por otra parte la optimización de las redes de transporte y distribución, y finalmente la optimización del consumo.

En cuanto a los diversos problemas de optimización de las redes de transporte, esta última sección se incluye en el contexto de la determinación de estrategias de control que minimicen los costes de explotación, ajustándose a las restricciones físicas del sistema y asegurando los niveles de demanda. No se consideran aquí otros problemas relativos a la optimización del transporte y distribución mediante el diseño de una red, con características y topologías capaces de minimizar los costes de implementación y funcionamiento.

El problema concreto que se aborda consiste en determinar la forma de distribuir el agua proveniente de las estaciones potabilizadoras o de los bombeos entre los depósitos disponibles en el sistema de distribución, de forma que se satisfaga la demanda en todo momento, sin que se supere el volumen máximo de cada depósito, y con el menor movimiento de válvulas posible (en adelante RAD, Regulación Automática de los Depósitos). Los datos experimentales utilizados, tanto en lo que se refiere a las características del sistema como a los niveles de demanda, corresponden a la red de transporte hidráulico de la ciudad de Granada.

La estrategia seguida para resolver el problema considera un nuevo procedimiento propuesto dentro del ámbito de la neuro-programación dinámica [Dam-00c, Dam-00d] (en adelante NPD) descrita en el Capítulo 2. Concretamente se basa en el uso de un algoritmo genético que permite determinar los puntos de funcionamiento aceptables en cada uno de los instantes de tiempo en los que hay que actuar sobre las válvulas del sistema para distribuir el caudal proveniente de la estación potabilizadora. Este conjunto de puntos factibles de funcionamiento del sistema, es el que se utiliza en un algoritmo de programación dinámica para tomar las decisiones de control de llenado de los depósitos.

Puesto que el número de puntos de funcionamiento factibles suele ser muy elevado, el tiempo de cómputo necesario para la toma de decisiones óptimas puede resultar

inaceptablemente alto para su aplicación al sistema con la suficiente antelación. Para ajustarse a las restricciones temporales es necesario trabajar con un subconjunto del total de soluciones factibles, por lo que se obtendrá una solución subóptima.

Como veremos, el uso del procesamiento en paralelo para la resolución de este problema se justifica no sólo porque puede reducir el tiempo necesario para obtener una solución al problema sino también porque, al permitir utilizar más soluciones factibles, contribuye a mejorar la calidad del óptimo encontrado en un tiempo similar al que se necesita si se utiliza un sólo procesador [Dam-99b, Dam-00b].

8.7.2 Descripción del procedimiento NPD propuesto

El procedimiento que proponemos aquí, basado en la NPD, se puede aplicar en general cuando para cada etapa es necesario seleccionar de entre un espacio continuo los posibles estados y controles de un sistema, y donde además las entradas son continuas y desconocidas, pero hasta cierto punto predecibles.

Una descripción general de la programación dinámica y la introducción a la técnica de NPD se hizo en la sección 2.4.3. En la mayoría de procedimientos de NPD propuestos con anterioridad ([Tes-92, Cri-96, Zha-96, Tes-97, Tsi-97]) se parte de una secuencia de acciones inicial, a partir de la cual se determina un valor de la función de coste futura y se obtienen mejoras sobre esa política inicial aplicando cambios en la misma que se aceptan si mejora el valor actual de la función de coste futuro. En algunos trabajos ([Tes-92, Cri-96, Zha-96, Tsi-97]) la función de coste futura se obtiene mediante aproximación funcional. Estos procedimientos necesitan mucho tiempo de entrenamiento (*'off-line'*) sobre un conjunto de posibles trayectorias de control, cuyo número suele ser muy elevado, y no son adecuados para la toma de decisiones en tiempo real. En otros trabajos [Tes-97], se realizan simulaciones Monte Carlo de posibles trayectorias de control y a partir de ellas se estiman los valores esperados necesarios para determinar la función de coste futuro. En el procedimiento que planteamos se utiliza una estrategia similar a la usada en [Tes-97], realizando simulaciones Monte Carlo de trayectorias de control, pero incluimos la posibilidad de reducir el espacio de trayectorias posibles si se utiliza un procedimiento que permita predecir los valores futuros en las entradas al sistema. El uso de los valores de entradas predichos, junto con la eliminación de soluciones factibles poco prometedoras en cada etapa del procedimiento (esto sí se utiliza en [Tes-97] por ejemplo) supone

una reducción en el número de trayectorias posibles y mejora la calidad de las soluciones, al tiempo que contribuye a reducir el tiempo puesto que la predicción se puede realizar concurrentemente con el resto de tareas del procedimiento propuesto. En nuestro procedimiento, la aproximación funcional se usa para obtener un modelo del sistema que permite realizar las simulaciones, en el caso de que no exista un modelo matemático previo. Este modelo se puede ir refinando a medida que se dispone de más patrones extraídos del sistema en funcionamiento.

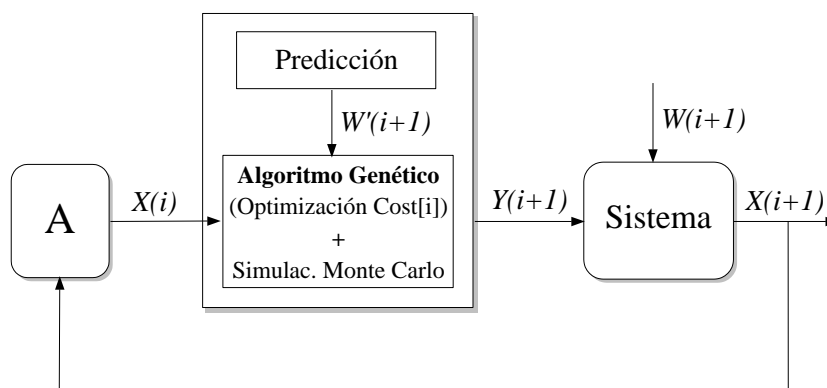


Figura 8.20: Diagrama de bloques del procedimiento NPD propuesto

La Figura 8.20 proporciona un esquema de módulos del procedimiento de neuro-programación dinámica propuesto, donde:

- $Cost(i)$ es la función de coste usada para determinar los puntos factibles de funcionamiento del sistema en el periodo i .
- $W(i)$ es el vector de entrada al sistema para el periodo i .
- $X(i)$ es el vector de salida (o estado) del sistema en el periodo i .
- $Y(i)$ es el vector de control para el periodo i .

La salida del módulo de predicción se usa como entrada al módulo de optimización, el cual determina un conjunto de puntos factible de funcionamiento mediante un algoritmo genético, y selecciona de entre estos el mejor para aplicarlo en la etapa siguiente. La selección del control óptimo se realiza mediante simulación Monte Carlo, como veremos a continuación

El procedimiento recibe el vector de estado (o de salidas) del sistema en la etapa i , $X(i)$ y debe generar el vector de las señales de control, $Y(i+1)$, para la etapa siguiente con la suficiente antelación. Para ello, el procedimiento implementa una serie de iteraciones que corresponden a las etapas futuras, en las que utiliza el modelo aproximado de sistema para simular su

comportamiento cuando se aplican vectores de control posibles y los valores previstos por el módulo de predicción para las entradas, en esas etapas, $W'(i)$. En cada iteración, i , los valores de los estados, $X(i)$, obtenidos mediante simulación, y el valor predicho para la entrada $W'(i)$ definen una función de coste, $Cost(i)$ que se utiliza para determinar los puntos de funcionamiento factibles para el sistema en la etapa $i+1$. Se trata de una serie de problemas de optimización multimodal que se deben resolver por parte del módulo de optimización. Una vez se han obtenido una serie de puntos factibles de funcionamiento para la etapa $i+1$, $Y(i+1)$, es necesario seleccionar los mejores para la siguiente iteración, para lo que hay que tener en cuenta el valor del coste inmediato asociado a cada solución factible, más un coste asociado a la evolución futura del sistema. Este coste viene representado por la función de coste futuro (*cost-to-go*) que se determina en el correspondiente módulo. Este proceso se repite hasta que, al finalizar las iteraciones del procedimiento, el módulo de coste futuro elige como señal $Y(i+1)$ aquella a partir de la que se haya construido la trayectoria de soluciones con un menor valor para la función de coste futuro.

Podemos ver que el procedimiento de neuro-programación dinámica que proponemos integra procedimientos de aproximación funcional, predicción, y optimización. Para resolver los problemas de aproximación funcional relacionados con el modelado se pueden utilizar técnicas neuronales y de lógica difusa. Para los problemas de optimización se han utilizado metaheurísticas de optimización estocásticas, especialmente, los procedimientos basados en la computación evolutiva, que permiten determinar los puntos de funcionamiento factibles del sistema. Y para la predicción de los valores de las entradas futuras al sistema se utilizan también técnicas basadas en redes neuronales. En este caso, tal y como se ha hecho, el problema de NPD se podría tratar como un problema determinista que actúa en el sentido de intervalos crecientes (en el sentido del tiempo), siendo bastante útil en problemas de tiempo real [Ber-96].

8.7.2.1 Descripción del algoritmo genético utilizado

Para determinar los valores de control factibles se utiliza un procedimiento híbrido que mezcla un algoritmo genético y un procedimiento de optimización local. Se combinan los algoritmos genéticos y los métodos de descenso en gradiente para aprovechar sus características y evitar sus inconvenientes.

Como ya indicamos en la sección 2.4.4, los algoritmos genéticos representan una alternativa a los métodos clásicos de optimización en problemas complejos con muchos óptimos locales, donde dichos métodos clásicos no son eficientes. Y además, los AGs presentan un compromiso entre explotación y exploración de soluciones, con búsquedas paralelas a través de los individuos de la población. Sin embargo, si la función a optimizar presenta un buen comportamiento entorno a un mínimo, el método de descenso en gradiente clásico permite una explotación eficiente de la información local, mejorando así la lenta convergencia que presentan los algoritmos genéticos.

Por consiguiente, en cada generación del algoritmo y sobre los individuos de la población, se aplican tanto los operadores genéticos de cruce y mutación como varias iteraciones del procedimiento de descenso en gradiente. Se evalúa la población y se elige al mejor individuo. De esta forma, el algoritmo genético mejora las soluciones obtenidas por el procedimiento de descenso en gradiente y además se incrementa la velocidad de convergencia. Como el algoritmo genético trabaja con las soluciones encontradas en la optimización local, se puede considerar que el algoritmo híbrido tiene características "Lamarckianas" [Ren-96].

Cada individuo de la población se corresponde con un vector de control, $Y(i)=(y_1(i), \dots, y_n(i))$, y cada uno de sus componentes, $y_k(i)$, codificado como un número real, representa un gen de ese individuo. Las características de los operadores de mutación y cruce [Gol-89, Jan-91], se describen a continuación:

Mutación: para cada generación se seleccionan dos parámetros, p y q , de forma aleatoria. El parámetro p puede valer +1 o -1 con igual probabilidad, mientras que q toma un valores en el intervalo [0,1] con una distribución uniforme. El gen mutado toma el valor:

$$\begin{aligned} y'_k &= y_k + (y_{max} - y_k)(1 - q^{[1-(t/T)]^s}) \quad \text{if } p=+1 \text{ o} \\ y'_k &= y_k + (y_k - y_{min})(1 - q^{[1-(t/T)]^s}) \quad \text{if } p=-1. \end{aligned} \quad (8-6)$$

En esta expresión (8-6), q determina indirectamente el alcance de la modificación, y_{min} y y_{max} son los límites inferior y superior del gen y_k respectivamente, T es el número de generaciones a aplicar la mutación, y s es el factor de decrecimiento de la probabilidad de mutación.

Cruce: Se aplican, como en [Ren-96], dos tipos de cruce, discreto y continuo, con probabilidades p_1 y p_2 respectivamente. Con el cruce discreto se obtiene un descendiente

tomando aleatoriamente un gen de uno u otro ancestro (o padre). El cruce continuo calcula los valores de alguno genes como la media de los valores de los correspondientes genes padres, mientras que el resto de los genes toman los mismos valores que uno de los padres.

Y en cuanto a las iteraciones del procedimiento de descenso en gradiente los genes se modifican según la siguiente expresión (8-7):

$$\begin{aligned} y'_k &= y_k + P && \text{if } \text{Cost}(y_1, \dots, y_k + P, \dots, y_n) < \text{Cost}(y_1, \dots, y_k - P, \dots, y_n) \text{ o} \\ y'_k &= y_k - P && \text{if } \text{Cost}(y_1, \dots, y_k + P, \dots, y_n) \geq \text{Cost}(y_1, \dots, y_k - P, \dots, y_n) \end{aligned} \quad (8-7)$$

donde P es la perturbación que se aplica a un componente dado de la solución.

8.7.2.2 La función coste futuro óptimo (*cost-to-go*)

Una vez obtenido un conjunto de puntos factibles para la etapa en curso ($i+1$), $\{Y(i+1)_j, j=1, \dots, M_{i+1}\}$, se necesita seleccionar el mejor entre ellos para aplicarlo en la etapa siguiente. El mejor de los $Y(i+1)_j$ se calcula teniendo en cuenta el coste inmediato asociado con su aplicación, y la evolución futura del sistema.

La implementación de esta parte del procedimiento de optimización se realiza mediante simulación Monte Carlo. Esta simulación usa las predicciones de los vectores de entrada para la etapa siguiente, $W'(i+2), \dots, W'(N-1), W'(N)$, del modelo disponible en la etapa i . Utilizando estas entradas predichas se puede estimar el conjunto de valores factibles para el control en la siguiente etapa $\{Y(i+2)_j, j=1, \dots, M_{j+2}\}, \dots, \{Y(N)_j, j=1, \dots, M_N\}$, y generar trayectorias de soluciones (como se muestra en la Figura 8.21) para aproximar la función coste futuro (*cost-to-go*) de la etapa $i+1$, utilizando la expresión (8-8).

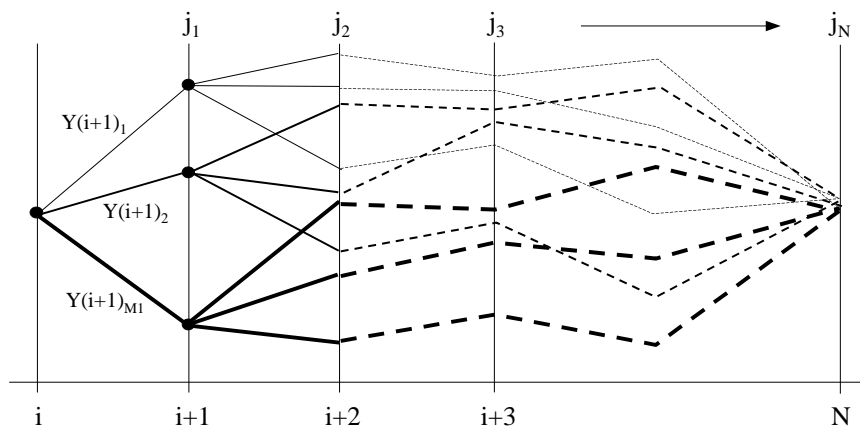


Figura 8.21: Generación de trayectorias del procedimiento NPD propuesto

$$J'(X(i+1)_j) = (1/K)(d(X(i+1)_j,1)+d(X(i+1)_j,2)+ \dots + d(X(i+1)_j,K)) \quad (8-8)$$

donde K es el número de trayectorias simuladas, y:

$$d(X(i+1)_{j1},m) = g(X(i+1)_{j1},Y(i+2)_{j2},X(i+2)_{j2}) + g(X(i+2)_{j2},Y(i+3)_{j3},X(i+3)_{j3}) + \dots \quad (8-9) \\ + \dots + g(X(N-1)_{jN-1},Y(N)_{jN},X(N)_{jN})$$

es el coste acumulado hasta alcanzar el estado final en la última etapa para la trayectoria m . Se asume que diferentes trayectorias simuladas son estadísticamente independientes.

Es posible calcular $J'(X(i+1)_j)$ iterativamente comenzando con $J'(X(i+1)_j)=0$, y usando [Bel-62]:

$$J'(X(i+1)_j) = J'(X(i+1)_j) + G_m * (d(X(i+1)_j,m) - J'(X(i+1)_j)) \quad (8-10)$$

con $G_m=1/m$, $m=1,2,\dots,K$. Una vez calculados los valores de $J'(X(i+1)_j)$ para $j=1,2,\dots,M_{i+1}$, se ordenan, y se selecciona para la siguiente etapa el control $X(i+1)_j$ ($j=1,\dots,M_{i+1}$) con menor valor de la función coste futuro aproximada.

8.7.3 Aplicación al control de la RAD

El problema concreto, donde hemos aplicado el procedimiento anterior, consiste en determinar la distribución del caudal que proporcionan las estaciones potabilizadoras (ETAP) entre los tres depósitos principales del sistema de distribución de agua de Granada, de forma que se satisfaga la demanda, que los depósitos ni se derramen ni se vacíen, y que no queden con un volumen de agua inferior a un mínimo establecido. En el Capítulo 3 se muestran esquemas del sistema en el caso de la red de distribución de aguas de Granada [Dam-99a].

La ETAP suministra un caudal que debe distribuirse entre los depósitos. El sistema de control ha de determinar el caudal que debe pasar a cada depósito, de forma que se produzcan los menores movimientos de las válvulas de entrada, que el caudal que suministre la ETAP sea el mínimo necesario y, salvo en situaciones muy excepcionales, se mantenga constante a lo largo del funcionamiento del sistema.

También se impone la restricción de que el contenido de los depósitos al comienzo y al final de cada día sea el mismo. Esta restricción constituye una condición de continuidad que

permite plantear el problema de control con un horizonte finito de 24 horas. Además, tiene sentido establecer esta condición puesto que los aspectos esenciales de consumo se suelen repetir diariamente, y se tienen valores mínimos para el consumo nocturno. Se han tomado volúmenes iniciales y finales iguales a la mitad de los volúmenes totales porque esto supone elegir una situación intermedia que permita al sistema reaccionar a tiempo en caso de que la demanda suba o se reduzca muy rápidamente. En el primer caso, se asegura que hay una reserva para mantener el suministro sin aumentar el caudal que proporciona la ETAP, y en el segundo caso, se asegura un cierto volumen libre en los depósitos para que el agua que proporciona la ETAP no ocasione desbordamientos.

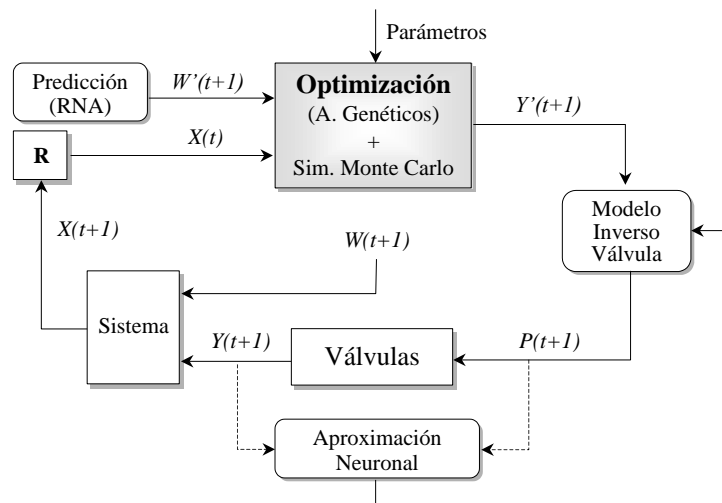


Figura 8.22: Esquema de bloques del procedimiento para la RAD

La Figura 8.22 muestra las variables del problema y esquematiza su uso en el procedimiento de control óptimo propuesto. En cada intervalo de tiempo t , se tiene en cuenta el nivel de los depósitos en ese momento, $X(t)$, y el valor previsto de la demanda en el periodo siguiente, $W'(t+1)$, para determinar el caudal que debe proporcionarse a cada depósito en dicho periodo, $Y'(t+1)$. La variable $X(t)$ es un vector $X(t)=(x[1][t], \dots, x[n][t])$ cuyos componentes son los niveles de los n depósitos del sistema en el instante t ; $W'(t+1)=(w[1][t+1], \dots, w[n][t+1])$ es el vector cuyos componentes representan los volúmenes de demanda previstos en cada uno de los n depósitos a lo largo del intervalo de tiempo siguiente (entre el instante t y el $t+1$); e $Y'(t+1)=(y[1][t+1], \dots, y[n][t+1])$ es el vector cuyos componentes corresponden al volumen de agua que debe pasar a cada uno de los n depósitos, también a lo largo del intervalo de tiempo siguiente. Como se ha indicado, el problema a resolver consiste en determinar el valor de los

componentes de $Y'(t)$, para cada uno de los instantes de tiempo considerados, de forma que la suma de los componentes del vector $Y'(t)$ sea igual al caudal suministrado desde la ETAP, satisfaciendo la demanda de agua en cada depósito y sin que éstos se desborden.

En cada intervalo de tiempo t se utiliza el vector $W'(t+1)$ con las previsiones de consumo para el siguiente intervalo. Estos valores se pueden estimar de formas muy diversas. En nuestro caso, la predicción de los niveles de consumo se ha realizado mediante un procedimiento basado en el uso de redes neuronales [Sal-99]. En cualquier caso, tal y como se muestra en la Figura 8.22, al final del intervalo de tiempo, el valor de la demanda real, $W(t+1)$, es el que determina los valores de los niveles de los depósitos, que se tendrán en cuenta para el nuevo intervalo de tiempo. Algo similar sucede con el valor de $Y'(t+1)$ obtenido por el procedimiento. A partir de dicho valor se determinan las posiciones de las válvulas para que sean los caudales determinados los que entren en cada depósito. Sin embargo, debido a la posible imprecisión en el calibrado de las válvulas, las interacciones que se producen entre los distintos caudales, dado el régimen de funcionamiento de la red, harán que se tengan unos caudales en cada depósito que difieren respecto de los calculados. Así, los volúmenes de agua que finalmente se introducen en cada depósito se indican en la Figura 8.22 como $Y(t+1)$, y son los que se tienen en cuenta para determinar los niveles de los depósitos en el siguiente intervalo de tiempo. En la Figura 8.22 también se representa un módulo que implementa una red neuronal entrenada durante el funcionamiento del sistema, para aproximar el comportamiento de las válvulas (también se podría haber planteado para esto un controlador difuso). De esta forma, se puede mejorar el conocimiento del modelo inverso de las válvulas y se reducirán las diferencias entre $Y'(t)$ e $Y(t)$. Este modelo neuronal evita tener que disponer de un modelo hidráulico del sistema como el que se utiliza en otras aproximaciones al control óptimo [Ali-98]. En adelante, y para simplificar la notación, no se hará distinción entre los valores $W(t)$ y $W'(t)$, o entre $Y(t)$ y $Y'(t)$.

8.7.3.1 Función de coste para el algoritmo genético

En la función de coste, las magnitudes y los parámetros que se utilizan, tal y como se han descrito en la sección anterior y aparecen en la Figura 8.22, son los siguientes:

$x[i][j]$ Volumen de agua en el depósito j ($j=1,\dots,n$) en el intervalo de tiempo i ($i=1,\dots,m$).

$w[i][j]$ Volumen de agua que se ha previsto que se consumirá desde el depósito j en el intervalo de tiempo i .

$y[i][j]$ Volumen de agua que se suministra al depósito j en el intervalo de tiempo i .

- C Volumen de agua suministrado por la ETAP en cada intervalo de tiempo (consigna, que en principio se considerará constante para todos los intervalos de tiempo).
- $V[j]$ Volumen máximo del depósito j .
- $L[j]$ Nivel deseado en el depósito j . Como se indicó en la sección anterior, se trata de un parámetro que se fija para permitir que se pueda responder a situaciones de poca o mucha demanda y evitar que un depósito se derrame o se vacíe, respectivamente.
- $H(a)$ Es una función tal que $H(a)=a$ si $a>0$ y $H(a)=0$ en caso contrario.

La función de coste para el intervalo de tiempo i (entre el instante $i-1$ y el instante i), $Cost[i]$, se obtiene a partir de la suma de cuatro términos,

$$Cost[i] = Cost1[i] + Cost2[i] + Cost3[i] + Cost4[i] \quad (8-11)$$

cuya forma y significado son:

- Cost1:** Permite imponer la condición de que la suma de las cantidades de agua suministradas a cada depósito es igual a la cantidad de agua total suministrada por la ETAP.

$$Cost1[i] = A \times \left| C - \sum_{j=1}^n y[i][j] \right| \quad (8-12)$$

- Cost2:** Representa la condición de que el volumen de agua en cada depósito no supere la capacidad máxima del mismo (no se derrame el depósito).

$$Cost2[i] = B \times \sum_{j=1}^n H(x[i-1][j] + y[i][j] - w[i][j] - V[j]) \quad (8-13)$$

- Cost3:** Corresponde la condición de que el volumen de agua en cada depósito sea mayor que cero (no se quede vacío).

$$Cost3[i] = D \times \sum_{j=1}^n H(-(x[i-1][j] + y[i][j] - w[i][j])) \quad (8-14)$$

- Cost4:** Permite imponer una condición que orienta la búsqueda de soluciones de forma que el nivel de agua en los depósitos se mantenga a unos valores prefijados. Se supone que estos niveles son aquellos que permiten que el sistema reaccione adecuadamente ante demandas muy elevadas o muy bajas, evitando que se vacíe o se desborde el depósito. La influencia de este término va aumentando a medida que se acerca el final del periodo diario considerado, ya que como se puede ver depende del índice del período elevado a una constante k .

$$Cost4[i] = E \times i^k \times \sum_{j=1}^n |x[i-1][j] + y[i][j] - w[i][j] - L[j]| \quad (8-15)$$

Los parámetros A , B , C , D , E deben ajustarse para que existan soluciones aceptables al problema.

Tal y como se ha indicado, el parámetro C corresponde al volumen de agua suministrado en cada intervalo de tiempo por la ETAP. La condición de continuidad que existe entre intervalos de tiempo consecutivos es:

$$x[i][j] = x[i-1][j] + y[i][j] - w[i][j] \quad (8-16)$$

Así pues, para cada intervalo de tiempo i , los mínimos de la función $Cost(i)$ constituyen los volúmenes de agua que, en el mencionado intervalo, deben entrar en cada depósito para que el agua proveniente de la ETAP se distribuya de forma que los depósitos ni se desborden, ni se vacíen, dado el nivel de demanda existente. A medida que el intervalo de tiempo se acerca al final del horizonte considerado, tiene más importancia en la función de coste el término que penaliza la desviación en el nivel de los depósitos respecto de la situación inicial de los mismos, que a su vez es la situación inicial del siguiente horizonte. En nuestro caso, el horizonte utilizado es de un día, los intervalos de tiempo son de una hora, y se considera que los niveles de los depósitos deben tener unos valores prefijados a las 0 horas de cada día.

La minimización de la función de coste $Cost(i)$ para un intervalo i dado, se realiza mediante un algoritmo genético híbrido basado en el descrito en [Ren-96], como ya hemos descrito en la sección 8.7.2.1.

8.7.3.2 Solución al problema de optimización.

Si el algoritmo genético se aplica sobre la función $Cost(1)$, se obtiene una forma de distribuir el agua entre los depósitos que, junto con el consumo registrado en ese periodo, determina el nivel de los depósitos para el siguiente intervalo. Aplicando nuevamente el algoritmo genético sobre la función $Cost(2)$ obtenida, se puede determinar una forma de distribuir el agua para el segundo intervalo de tiempo. Así, encontrando una solución para las sucesivas funciones de coste obtenidas, $Cost(3)$, ..., $Cost(24)$, se define una trayectoria de control factible, ya que en cada intervalo se define la forma de distribuir el agua entre los depósitos cumpliendo las restricciones físicas y los niveles de demanda, y al final del horizonte proporcionará los niveles deseados en los depósitos. Sin embargo, es sólo una de las posibles trayectorias, y no necesariamente la óptima.

En este trabajo se ha utilizado un procedimiento aproximado que empieza por determinar la función de coste $Cost(1)$. Se obtiene un número máximo, M , de valores óptimos de esta función de coste, y a partir de cada una de esas soluciones se construye una trayectoria de control

para estimar la función coste futuro (*cost-to-go*) correspondiente a cada uno de los valores factibles. Para ello, el módulo de predicción genera una estimación del agua a consumir en la siguiente etapa, $W'(2), \dots, W'(24)$. Y usando estos valores, se calculan iterativamente las funciones de costes $Cost(2), Cost(3), \dots, Cost(24)$ para cada uno de los puntos factibles calculados en la optimización de $Cost(1)$, y se definen K trayectorias para cada uno de esos puntos. De esta forma, en el primer intervalo se obtienen $M \cdot K$ trayectorias a partir de las M soluciones de $Cost(1)$ y la función coste futuro para cada control factible calculados como se ha indicado en la sección 8.7.2. La función de coste g considera los cambios de la posición de las válvulas. Los valores de $g(X(i)_{j0}, Y(i+1)_{j1}, X(i+1)_{j1})$ decrecen con la distancia entre $Y(i)$ y $Y(i+1)$, porque esto implica cambios menores en las posiciones de las válvulas.

En este procedimiento, la longitud de las trayectorias de control que hay que determinar en cada intervalo de tiempo va disminuyendo a medida que se van analizando los sucesivos periodos. En la primera etapa hay que procesar 23 periodos para construir las trayectorias de control, en la segunda hay que procesar 22, y así sucesivamente. De esta forma, a medida que se avanza en el periodo a procesar, puede incrementarse el número de trayectorias, K , que se simulan. Este número de trayectorias viene limitado por el tiempo de cálculo máximo, que está determinado por el tiempo necesario para aplicar la acción de control. A medida que se procesan más trayectorias crecen las prestaciones del procedimiento de control.

El procesamiento paralelo [Cul-99] resulta beneficioso porque permite procesar un mayor número de trayectorias en un tiempo prefijado, mejorando el comportamiento del procedimiento de control. Además, también permite reducir el tiempo necesario para procesar un número prefijado de trayectorias. En la siguiente sección se proporcionan los resultados experimentales que ilustran estas posibilidades.

8.7.4 Resultados Obtenidos

En la Figura 8.23 y Figura 8.24 se pueden observar algunas de las trayectorias que se generan actualmente como resultado de las operaciones manuales realizadas por los operadores sobre las válvulas (Man1, Man2, Man3), así como un ejemplo de una trayectoria obtenida con este procedimiento de control (Controlador). Se ha utilizado como entrada un fichero de datos reales con la evolución de los depósitos y consumos para un día típico, para poder simular los valores necesarios en cada periodo de cálculo.

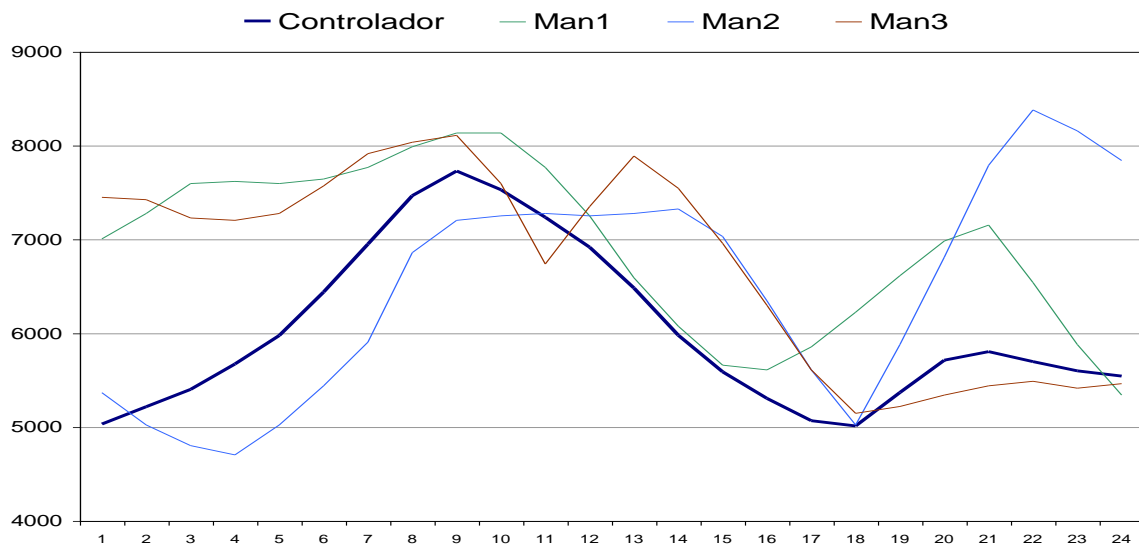


Figura 8.23: Comparativa entre trayectorias manuales y calculada para el nivel del depósito 1 (24 h.) en la RAD

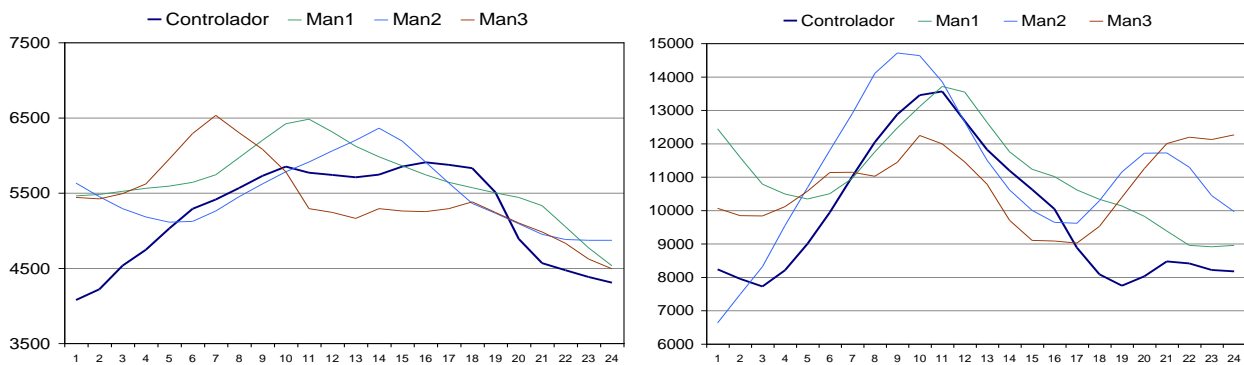


Figura 8.24: Comparativas entre trayectorias manuales y calculadas para el nivel de los depósitos 2 y 3 en la RAD

Se puede apreciar, observando la evolución de los niveles de los depósitos, que los resultados obtenidos con este método son similares e incluso más uniformes y suavizados que los actuales.

Además, como ya hemos comentado anteriormente, los resultados se pueden mejorar utilizando procesamiento en paralelo, y para demostrarlo, se ha implementado también el procedimiento NPD propuesto en PVM³ [Gei-94] y se ha usado un 'cluster' de 9 PCs con Pentium II a 333 MHz conectados con Fast Ethernet (100 Mbs) [And-95].

³ PVM = *Parallel Virtual Machine*. Biblioteca pública de primitivas de paso de mensajes que permite que una red heterogénea de estaciones de trabajo (o de computadores en general) pueda ser utilizada como un computador paralelo de memoria distribuida (Máquina virtual paralela).

En primer lugar se ha observado la mejora en la calidad de la solución obtenida al aumentar el número de procesadores que intervienen en la construcción de trayectorias de control, manteniendo constante el tiempo de procesamiento en cada procesador. En la Figura 8.25, las curvas de la gráfica superior indican la variación de los volúmenes, $Y(t)$, que pasan a cada uno de los tres depósitos, y las de la inferior a los niveles, $X(t)$, de dichos depósitos. Como se puede observar, la solución obtenida cuando se utilizan 8 procesadores presenta una menor variación en cuanto a los volúmenes de agua que se introducen en los depósitos, comparada con las variaciones que se producen cuando se ha utilizado únicamente un procesador.

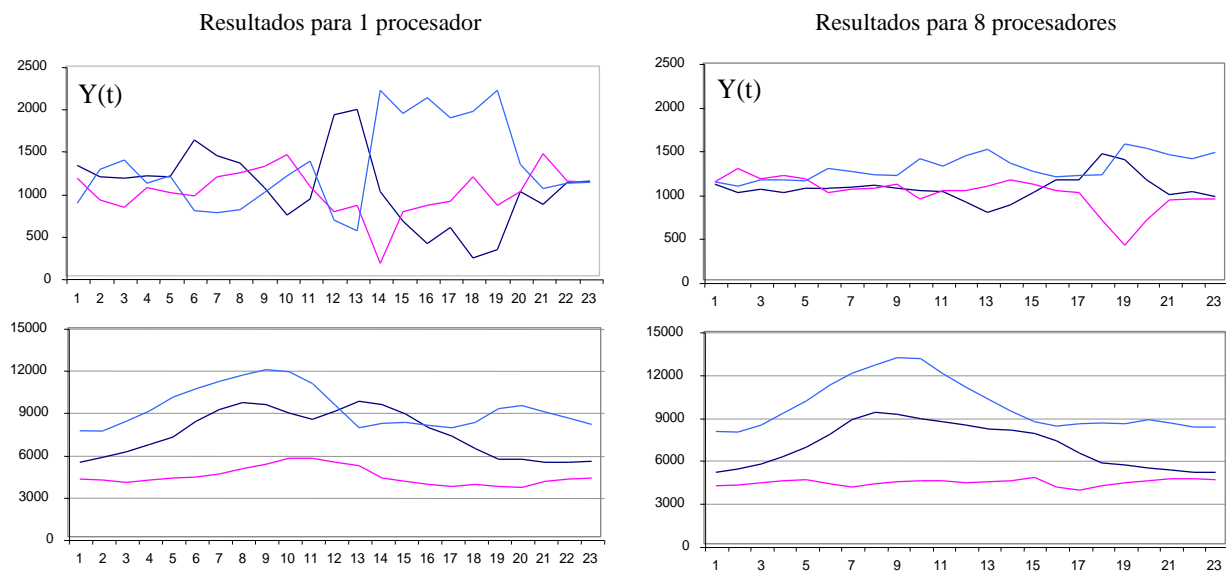


Figura 8.25: Mejora de los resultados utilizando procesamiento paralelo en la RAD

En la Figura 8.26 se muestran las ganancias de velocidad cuando se distribuye un número fijo de trayectorias a construir entre los procesadores disponibles. Se ha fijado el número de trayectorias a 9, y se han utilizado 1, 2, 3, 5, y 9 procesadores (respectivamente 9, 5, 3, 2, y 1 trayectorias calculadas como máximo en algún procesador). Las eficacias obtenidas son bastante satisfactorias, ya que están entre 0.8 y 0.9. Los valores más bajos de la eficacia se producen cuando el número de trayectorias que se evalúan no es múltiplo del número de procesadores. Se puede reducir este efecto haciendo una distribución del trabajo con una granularidad más fina, como por ejemplo paralelizando el algoritmo genético utilizado en la determinación de las soluciones factibles que definen las trayectorias de control.

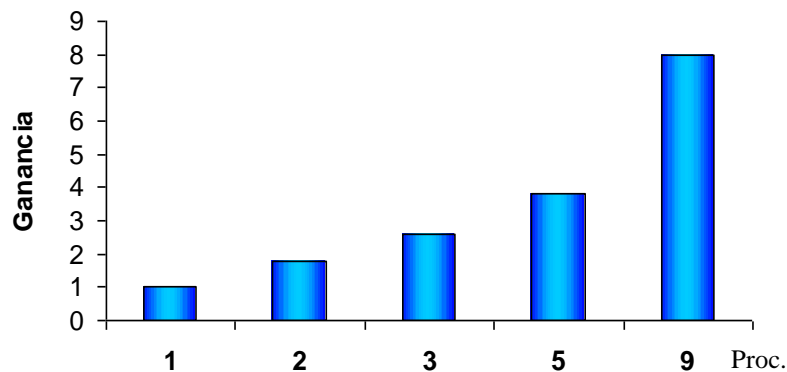


Figura 8.26: Ganancia de velocidad en función del n° de procesadores para 9 trayectorias en la RAD

Y por último, en la tabla de la Figura 8.27 se muestran los resultados, para diferentes números de trayectorias y procesadores (valores medios obtenidos de cinco ejecuciones para cada caso), que dan una idea de los tiempos de cálculo que necesita este procedimiento, así como las mejoras que se obtiene al utilizar la implementación paralela del algoritmo.

Procesadores\Trayectorias	8 Tray.	16 Tray.	32 Tray.
1 Proc.	42.0 min.	74.7 min.	145.3 min.
4 Proc.	13.9 min.	25.6 min.	48.5 min.
8 Proc.	7.8 min.	11.1 min.	20.1 min.

Figura 8.27: Tiempos de cálculo para diferentes n° de trayectoria y procesadores en la RAD

8.7.5 Líneas actuales y futuras de investigación

Como hemos comentado al comienzo del capítulo, entre los objetivos principales de las empresas de gestión del agua están la reducción de costes y el incremento de la calidad de sus servicios, dentro de un marco legal estricto en el que cada vez son más necesarios los sistemas de información y planificación para el apoyo en las decisiones, de forma que se pueda cumplir y verificar la realización del servicio de forma óptima [Jar-93].

Dentro de los problemas relativos a la optimización integral de los sistemas de abastecimiento de agua nos hemos centrado en la determinación de estrategias de planificación y control que minimizan los costes de explotación, ajustándose a las restricciones físicas del sistema y asegurando la demanda. Actualmente y en el futuro vamos a continuar investigando en esta línea que se ha abierto gracias a las herramientas proporcionadas y los resultados obtenidos

a raíz de este trabajo. Es decir, estaremos involucrados en aplicaciones relacionadas con la modelización y operación de las redes de distribución de agua.

El objetivo de la modelización es disponer de una forma eficiente de conocer la respuesta del sistema ante una entrada o acción determinada. El problema de generar un modelo se puede abordar mediante técnicas de aproximación funcional a los patrones de entrada/salida de los que ofrece el sistema desarrollado, utilizando modelos basados en redes neuronales y en lógica difusa (como se ha indicado en la sección 8.4.4).

En cuanto a la operación sobre las redes, los procedimientos de optimización basados en meta-heurísticas, como los algoritmos evolutivos, constituyen técnicas adecuadas para abordar la alta no-linealidad de las funciones de coste asociadas a estos problemas: no se necesita asumir ninguna hipótesis de linealización, ni calcular derivadas parciales, y se evitan las inestabilidades numéricas asociadas a la inversión de matrices. En nuestro trabajo se ha abordado, utilizando neuro-programación dinámica, el problema de distribuir entre una serie de depósitos un caudal prefijado de forma que se satisfaga la demanda existente sin que se derramen los depósitos.

En este sentido el trabajo que hemos presentado, supone una primera aproximación a la metodología propuesta. Los resultados que se han obtenido muestran la utilidad del uso de técnicas de predicción de series dentro del procedimiento de neuro-programación dinámica, y las mejoras en la calidad de soluciones y en el tiempo de respuesta que ofrece el procesamiento paralelo. No obstante, el procedimiento puede hacerse más eficaz si se aborda el problema de la optimización multimodal, y se incluyen procedimientos de distribución dinámica de la carga de trabajo entre los procesadores disponibles. Usualmente, en las primeras etapas del proceso hay que simular un mayor número de etapas en las trayectorias de control hasta que se alcanza el final del período de control (24 horas). De esta forma, a medida que se consideran etapas más próximas al final se puede ir incrementando el número de trayectorias simuladas, ramificando las que se están considerando. Esta es otra mejora interesante que habrá que incluir. Por otra parte, el procedimiento de neuro-programación dinámica propuesto debe extenderse a otras aplicaciones dentro de los sistemas de distribución hidráulica. Concretamente, pensamos utilizarlo también para optimizar aún más el consumo energético de los bombeos.

Y por último, queremos indicar también que el acceso a través de Internet a las plataformas y a los procedimientos desarrollados en este trabajo es otro de los objetivos que tenemos planteado, y para lo cual se ha solicitado recientemente un proyecto de investigación.

Concretamente, se trata de permitir, a través de las correspondientes interfaces, la ejecución de las aplicaciones en las mismas plataformas que se configurarán. Debe permitirse el acceso a los recursos por cualquier usuario autorizado, desde cualquier equipo conectado a Internet, y ha de ser transparente a la plataforma utilizada. La forma más sencilla sería acceder a través de un navegador (*browser*), desde una página *web*, ya que la *web* es la interfaz perfecta para las aplicaciones que se ejecutan en una plataforma de *metacomputación* [Gen-99], al poderse aprovechar toda la colección de aplicaciones, protocolos, y librerías que facilitan el acceso a la red a través de los navegadores.

En los entornos propios de la ciencia y la tecnología, casi siempre ligados a universidades y centros públicos de investigación, la *metacomputación* (cuyo propósito es permitir utilizar sistemas de altas prestaciones conectados a través de una *LAN* o una *WAN*) se ha experimentado con cierto éxito, pero también ofrece muy buenas perspectivas en la industria, donde la idea es proporcionar a las medianas y pequeñas empresas servidores de aplicaciones de altas prestaciones para sus problemas intensivos en cómputo. Los usuarios accederían a las aplicaciones a través de un navegador, sin necesidad de realizar procesos de instalación de software ni de actualización de versiones. En lugar de pagar por todas las características de un programa (también por las que no se van a utilizar) el usuario abonaría una cuota cada vez que se conecta al servidor en función de las opciones que utiliza [Nea-99, Haw-99]. Además todos los usuarios tienen siempre la última versión de la aplicación, y si se detecta un error sólo hay que repararlo en el servidor. Así, un entorno de *metacomputación* puede verse como una faceta más del campo del comercio electrónico [Byw-99].

Así pues, se pretende configurar un servidor de cómputo de altas prestaciones distribuido (constituido por computadores y *clusters* conectados mediante Internet/Intranet), para resolver problemas de optimización complejos, en nuestro caso relacionados con la planificación y predicción en sistemas de distribución de aguas.

8.8 Conclusiones

En este último capítulo se han propuesto una serie de herramientas y procedimientos para la optimización integral de los sistemas de abastecimiento de agua. Al comienzo hemos introducido los objetivos del control óptimo para dichos sistemas, y a continuación, en el resto de las secciones, se han analizado cada una de las contribuciones realizadas hasta ahora, que han mejorado o pueden mejorar la explotación de este recurso natural. También se han planteado a lo largo del capítulo las líneas actuales y futuras de investigación surgidas de los trabajos realizados y descritos en torno a las distintas facetas de la optimización que se han abordado.

En la sección 8.3 se ha tratado el problema de la optimización de los bombeos de la red de abastecimiento de agua de la ciudad de Granada, y se ha mostrado tanto el diseño realizado en la aplicación SCADA como la metodología seguida para conseguir los objetivos de minimización de los costes cumpliendo los requisitos de demanda y satisfaciendo las restricciones operacionales. Concretamente, con esta primera aproximación se ha conseguido un ahorro extra del 15% en los costes económicos de la energía.

En la sección 8.4 se ha mostrado cómo el diseño de una herramienta adecuada para el sistema de telecontrol de cada uno de los sectores en que se ha dividido la ciudad, permite mejoras muy significativas en aspectos tan importantes para empresas encargadas de la explotación del agua como detección de fugas y saltos bruscos en el caudal, y presiones más acordes con la topografía de la zona. Se han dado datos concretos del ahorro conseguido (solo en detección de fugas se consigue el 6% del consumo total en 1999) así como de las posibilidades que ofrece el sistema.

En la sección 8.5 se describe la automatización de uno de los puntos más críticos en las redes de distribución de agua, la inyección de caudal en la arteria principal de la ciudad utilizando válvulas reductoras de presión. Debido a que en estos sistemas la seguridad es uno de los puntos clave, se ha propuesto un algoritmo de control híbrido apoyándose en el comportamiento hidráulico, para hacer frente a la inestabilidad y no-linealidad del sistema. También se han mostrado las ventajas y posibilidades que ofrece el control automático a distancia del sistema propuesto, fundamentalmente en cuanto a la regulación dinámica de la presión en toda la arteria principal.

Finalmente, en las secciones 8.6 y 8.7 se trata el problema de la predicción del consumo y de la regulación automática de depósitos de forma más teórica, utilizando nuevas metodologías apoyadas más en la experiencia que aportan los datos proporcionados por el sistema diseñado, que en complejos y costosos modelos matemáticos basados en la dinámica de fluidos. En cuanto a la predicción del consumo se han utilizado redes neuronales como paradigma de cálculo sencillo de implementar y generalizable, y se han obtenido resultados muy satisfactorios, con una desviación típica del 3,7 %.

Y por último, para la regulación automática de depósitos se ha propuesto una nueva y completa metodología (basada en la neuro-programación dinámica) para acometer problemas de optimización en sistemas dinámicos, que permite manejar funciones de coste no lineales y variables de tipo continuo o discreto, tener en cuenta las influencias por el efecto y coste de una orden, y modelar las interacciones entre los estados del sistema. Hemos aplicado dicho procedimiento a este problema de distribución automática, y se han obtenido, en simulaciones sucesivas, resultados que demuestran su buen comportamiento. También se ha justificado el uso del procesamiento paralelo en este tipo de aplicaciones donde se utiliza la programación dinámica, y se han comprobado sus beneficios en cuanto a las mejoras en velocidad y resultados obtenidos al permitir mayores capacidades de procesamiento.

Capítulo 9:

Conclusiones y líneas de investigación futuras

El objetivo de la tesis doctoral se puede resumir de forma muy sencilla de la siguiente forma: *"desarrollar técnicas y elementos que contribuyan al diseño, integración y optimización de sistemas de control distribuido aplicados a procesos industriales en general, y en particular en el ámbito de las redes de abastecimiento de agua de ciudades"*. Los resultados concretos de la materialización del trabajo realizado corresponden a la automatización completa de todo el ciclo integral del agua de la ciudad de Granada.

La metodología de integración que hemos propuesto ha implicado las siguientes realizaciones, que constituyen las principales aportaciones del trabajo:

- Estudio de las especificaciones de los sistemas de abastecimiento y saneamiento de agua, y determinación de las necesidades de control para este supuesto real, con el objetivo de adoptar una arquitectura de red de comunicaciones y dispositivos de control adecuada, y a la vez completamente flexible y generalizable a diversos procesos de telemedida y telecontrol. En este contexto se ha justificado el uso tanto de las comunicaciones por radio como del sistema SPIRAL.
- Definición de una programación homogénea para los sistemas de control distribuido utilizados en estaciones remotas, que consigue facilitar las tareas de desarrollo y mantenimiento posterior.

- Desarrollo de un sistema para poder manipular a distancia, vía telefonía móvil GSM-SMS, una aplicación de control y supervisión, con la cual se ha conseguido una mayor operatividad y un aprovechamiento más eficiente de los recursos humanos.
- Concepción y diseño de un sistema de control distribuido de propósito general y adecuado a las necesidades de supervisión y control remoto de redes amplias como es el caso de la red hidráulica de distribución de agua potable de la ciudad de Granada. El sistema Spiral-5000 propuesto es un producto totalmente terminado y operativo, modular, fácil de configurar y de programar, y con una relación coste/prestaciones superior a otras opciones existentes.
- Como herramienta especialmente concebida para la integración de sistemas se ha diseñado e implementado un gestor de comunicaciones orientado al control y supervisión de sistemas distribuidos, basado en los conceptos de modularidad y estandarización, y con una gran variedad de posibilidades en cuanto a modos de acceso, topologías de interconexión, medios de comunicación, manejo de dispositivos de distinto tipo, y optimización del tamaño de las tramas y tiempos de sondeo. Se ha mostrado la utilización de este sistema tanto en simulaciones como en aplicaciones reales, y se han obtenido resultados que demuestran las ventajas que supone su instalación.
- Se ha utilizado eficientemente un entorno abierto basado en SCADA, para diseñar aplicaciones de supervisión y control óptimo de un proceso tan heterogéneo como es la automatización del ciclo integral del agua de una ciudad.
- En el último capítulo se han propuesto una serie de herramientas y procedimientos de optimización en la explotación de este recurso natural:
 - 1) En primer lugar se ha analizado el problema de optimización de los bombeos de la red de abastecimiento de agua de la ciudad de Granada, y se ha diseñado una aplicación con la metodología seguida para conseguir los objetivos de minimización de costes cumpliendo los requisitos de demanda y satisfaciendo las restricciones operativas. Con esta primera aproximación se ha conseguido un ahorro extra del 15% en los costes económicos de la energía.
 - 2) En el mismo capítulo, se ha mostrado cómo el telecontrol de cada uno de los sectores en que se ha dividido la ciudad, permite mejoras muy significativas en aspectos tan importantes como detección de fugas y saltos bruscos en el caudal, así como en la regulación de presiones. Se han proporcionado datos concretos del ahorro conseguido así como de las posibilidades que ofrece el sistema. Concretamente, sólo en el año

1999, y gracias a este sistema, se han ahorrado aproximadamente 2 millones de m³ de agua, el equivalente a casi un mes de consumo.

- 3) También se describe la automatización de uno de los puntos más críticos en las redes de distribución de agua, la inyección de caudal en la arteria principal de la ciudad utilizando válvulas reductoras de presión. Se ha propuesto un algoritmo híbrido de control basado en el comportamiento hidráulico, para hacer frente a la inestabilidad y no-linealidad del sistema y conseguir una alta seguridad. También se han mostrado las ventajas y posibilidades que ofrece el control automático a distancia de estas estaciones, fundamentalmente en cuanto a la regulación dinámica de la presión en toda la arteria principal.
- 4) Finalmente, se ha tratado de forma teórica el problema de la predicción del consumo y de la regulación automática de depósitos, utilizando nuevas metodologías apoyadas más en la experiencia que aportan los datos proporcionados por el sistema diseñado, que en complejos modelos matemáticos basados en la dinámica de fluidos.
 - En cuanto a la predicción del consumo se han utilizado redes neuronales como paradigma de cálculo sencillo de implementar y generalizable, obteniéndose resultados muy satisfactorios, con una desviación típica del 3,7 %.
 - Y en la resolución del problema de la regulación automática de los depósitos se ha propuesto un nuevo y completo procedimiento basado en neuro-programación dinámica para acometer problemas de optimización en sistemas dinámicos, que permite manejar funciones de coste no lineales y variables de tipo continuo o discreto, tener en cuenta las influencias por el efecto y coste de una orden, y modelar las interacciones entre los estados del sistema. Se ha aplicado dicho procedimiento a este problema de distribución automática, y hemos obtenido, en simulaciones sucesivas, resultados que demuestran su buen comportamiento. También se ha justificado el uso del procesamiento paralelo en este tipo de aplicaciones donde se utiliza la programación dinámica, y se han comprobado sus beneficios en cuanto a las mejoras en velocidad y resultados.

El diseño de los sistemas propuestos en esta memoria se puede utilizar también en la automatización de otros procesos del mismo o de distinto tipo. En concreto, usando esta misma metodología y en colaboración con las empresas ICR S.A. y Transtelecom S.A., se han realizado, entre otras, los sistemas de control distribuido que se muestran con detalle en el apéndice A, y que resumimos a continuación:

- Telecontrol de redes de abastecimiento de agua en Córdoba, Huelva, Jaén, León, Málaga, Almería, Valencia, Gran Canarias y Burgos.
- Telecontrol de redes de saneamiento de agua en Almería, Valencia, Alicante y Málaga.
- Supervisión y control de estaciones de tratamiento agua potable en Alicante, Huelva, Almería, Málaga y Córdoba.

- Supervisión y control de estaciones depuradoras de aguas residuales en Almería, León, Málaga, Gran Canaria, Jaén y Melilla.
- Telecontrol de redes de riego en Almería, Madrid, Jaén, Huelva y Murcia.
- Supervisión y control de la estación de esquí de Sierra Nevada en Granada.
- Supervisión y control de procesos de fabricación de papel y aluminio (TORRAS PAPEL en Granada, CELUPAL S.A en Cádiz, o INESPAL en Alicante).
- Telecontrol de faros de El Ferrol en la Coruña, del distrito de Shanghai en China, y de Cap-Bengut en Argelia.

En cuanto a las líneas de investigación futuras, actualmente se está trabajando en varios frentes, todos ellos derivados de este trabajo:

- Un campo abierto es el de la calibración de un modelo simplificado de la red de distribución de agua de Granada, mediante un sistema neuro-difuso, para usarlo en lugar de la planta real, y así poder ajustar un controlador difuso con capacidad de aprendizaje y adaptación continua.
- Al mismo tiempo, se continuará mejorando el procedimiento NPD propuesto, incluyendo optimización multimodal y distribución dinámica de la carga. Se pretende también extender este procedimiento a otras aplicaciones dentro y fuera de los sistemas de distribución hidráulica; concretamente estamos pensando en utilizarlo para optimizar aún más el consumo energético de los bombeos. Recientemente se ha solicitado financiación para un nuevo proyecto de investigación con el que se pretende, entre otras cosas, llevar a la práctica estos procedimientos de optimización, difundirlos y utilizarlos vía Internet/Intranet.
- El objetivo final es agrupar todos los sistemas de optimización desarrollados para obtener un único sistema experto global de redes de abastecimiento de agua de ciudades, completamente autónomo y en tiempo real, tal y como se muestra en la Figura 9.1.

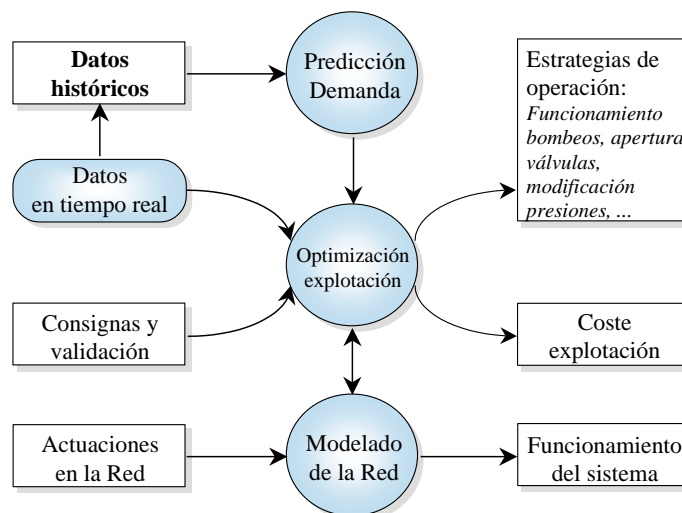


Figura 9.1: Sistema de operación automática en tiempo real en redes de abastecimiento de agua

Apéndice A:

Listado de sistemas de control distribuido

Realizados en cooperación con la empresa Ingeniería y Control Remoto S.A., donde se han utilizado todos o algunos de los sistemas que se han descrito en la memoria.

1. Telecontrol de la red de abastecimiento de agua de la ciudad de Alcalá la Real, Jaén

Telecontrol vía radio de un total de 16 estaciones remotas que conforman los principales bombeos y depósitos que abastecen a la ciudad de Alcalá. El control y supervisión se realiza mediante autómatas de Siemens (95U), dos ordenadores en red local, software SCADA InTouch, y driver ICeR.

2. Control Microcentral presa de Canales, Granada

Control automático de caudales de salida de la presa en función de la demanda solicitada para riego o abastecimiento. El control y supervisión se realiza mediante un autómata de Siemens (115U), un ordenador, software SCADA InTouch, y driver ICeR.

3. Telecontrol de la red de abastecimiento de agua de la ciudad de Antequera, Málaga

Telecontrol vía radio de un total de 8 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la ciudad de Antequera. El control y supervisión se realiza mediante autómatas Hitachi y Siemens, un ordenador, software SCADA InTouch, y driver ICeR.

4. Telecontrol de la red de abastecimiento de agua y de la ETAP del Bajo Almanzora, Almería

Telecontrol vía radio de un total de 26 estaciones remotas que conforman la red de abastecimiento para toda la comarca del Almanzora (20 Pueblos y una extensión de 400 Km), y control automático de la planta de tratamiento de agua potable. El control y supervisión se realiza mediante autómatas de Siemens (115U y 95U), un ordenador, software SCADA InTouch, y driver ICeR.

5. Telecontrol de la red de abastecimiento de agua de la ciudad de Huelma, Jaén

Telecontrol vía radio de un total de 5 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la ciudad de Huelma. El control y supervisión se realiza mediante autómatas de Siemens (95U), un ordenador, software SCADA InTouch, y driver ICeR.

6. Telecontrol de la red de abastecimiento de agua de la comarca de Benamargosa, Málaga

Telecontrol vía radio de un total de 11 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la comarca de Benamargosa (9 municipios). El control y supervisión se realiza mediante autómatas de Siemens (95U) y Spiral 3000¹, un ordenador, software SCADA InTouch, y driver ICeR.

7. Telecontrol de la red de abastecimiento de agua de la ciudad de Velez Málaga, Málaga

Telecontrol vía radio de un total de 8 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la ciudad de Velez Málaga. El control y supervisión se realiza mediante autómatas de Siemens (95U) y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

8. Telecontrol de la red de abastecimiento de agua del Sureste, Isla de Gran Canaria

Telecontrol vía radio de un total de 7 estaciones remotas que conforman las impulsiones y depósitos principales que abastecen a la comunidad del Sureste (6 municipios). El control y supervisión se realiza mediante autómatas de Siemens (95U), un ordenador, software SCADA InTouch, y driver ICeR.

9. Telecontrol de la estación de aforo El Guaro, Málaga

Telecontrol vía radio de un total de 2 estaciones remotas que conforman el bombeo y depósito principales que abastecen a la comarca del Guaro. El control y supervisión se realiza mediante autómatas de Siemens y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

10. Telecontrol de la reutilización para riego de aguas residuales del Bajo Andarax, Almería

Telecontrol vía radio de un total de 5 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la comarca de riego del bajo Andarax. El control y supervisión se realiza mediante autómatas de Siemens y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

11. Telecontrol de la red de abastecimiento de agua de la comarca de Ponferrada, León

Telecontrol vía radio de un total de 9 estaciones remotas que conforman los bombeos y depósitos principales que abastecen a la comarca de Ponferrada (8 municipios). El control y supervisión se realiza mediante autómatas de Siemens (95U), un ordenador, software SCADA InTouch, y driver ICeR.

12. Control y supervisión de la ETAP de la Viñuela, Málaga

Control automático de la planta de tratamiento de agua potable. Realizado mediante un autómata General Electric. El control y supervisión se realiza mediante un autómata General Electric, un ordenador, software SCADA InTouch, y driver ICeR (vía radio recibe además toda la información de la red de abastecimiento para toda la comarca Benamargosa y ciudad de Velez Málaga).

¹ Spiral 3000 es un sistema de control distribuido, desarrollado también en nuestro departamento, anterior y similar al SP-5000, pero con menos prestaciones.

13. Telecontrol de la red de abastecimiento de agua de la ciudad de Torremolinos, Málaga

Telecontrol vía radio de un total de 11 estaciones remotas que conforman los bombeos, depósitos principales que abastecen a la ciudad de Torremolinos, y 3 estaciones de telecontrol para los bombeos de aguas residuales. El control y supervisión se realiza mediante autómatas de Siemens (115U/95U), un ordenador, software SCADA InTouch, y driver ICeR.

14. Telecontrol de la EDAR y los bombeos del aeropuerto de Granada

Telecontrol vía radio de un total de 3 estaciones remotas para la depuradora de aguas residuales y los bombeos de agua potable del aeropuerto de Granada. El control y supervisión se realiza mediante autómatas de Siemens (95U), un ordenador, software SCADA InTouch, y driver ICeR.

15. Telecontrol de la red de abastecimiento de agua de Monachil, Granada

Telecontrol vía radio de un total de 6 estaciones remotas que conforman las impulsiones y depósitos principales que abastecen a las urbanizaciones de la estación de Esquí de Sierra Nevada. El control y supervisión se realiza mediante autómatas de Siemens (95U), un ordenador, software SCADA InTouch, y driver ICeR.

16. Telecontrol de la red de riego de la comunidad de regantes de Balerma, Almería

Telecontrol vía radio de un total de 9 estaciones remotas que conforman los bombeos, depósitos principales que abastecen a la comunidad de regantes. El control y supervisión se realiza mediante autómatas de Alfa-Laval, un ordenador, software SCADA InTouch, y driver ICeR.

17. Control y Supervisión de procesos de la Factoría Torras Papel en Motril, Granada

Control y supervisión de procesos como la fabricación de almidón, línea de recorte, refino, aguas blancas y aguas recuperadas, entre otros. El control y supervisión se realiza mediante autómatas de Siemens (115U/95U) conectados en red (Sinec-L1), ordenadores en red, y software SCADA InTouch.

18. Telecontrol de la red de abastecimiento de agua de la Contraviesa, Granada

Telecontrol vía radio de un total de 25 estaciones remotas que conforman los bombeos, depósitos principales que abastecen a la comarca de la Contraviesa (12 municipios). El control y supervisión se realiza mediante autómatas de Siemens (95U) y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

19. Telecontrol integral de la estación de Esquí de Sierra Nevada, Granada

Automatización y telecontrol de procesos como el sistema de producción de nieve, circuito cerrado de televisión, iluminación pistas, control de megafonía, paneles de información, estaciones meteorológicas, y centros de transformación. El control y supervisión se realiza mediante autómatas de Siemens (135U/115U/95U) y SP-5000, ordenadores en red, software SCADA InTouch, y driver ICeR.

20. Telecontrol de la red de abastecimiento de agua de la comarca del Condado, Huelva

Telecontrol vía radio de un total de 26 estaciones remotas que conforman los bombeos, depósitos principales que abastecen a la comarca del Condado (15 municipios). El control y supervisión se realiza mediante autómatas de Siemens (95U) y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

21. Control y supervisión de la EDAR de Almería

Control automático de la planta de depuración de aguas residuales de Almería. El control y supervisión se realiza mediante autómatas de Siemens (115U) conectados en red (Sinec-L1), un ordenador, y software SCADA InTouch.

22. Automatización y Supervisión de la ETAP de la Pedrera, Alicante

Control automático de la planta de tratamiento de agua potable de la Pedrera en Alicante. El control y supervisión se realiza mediante autómatas de Omron (C-200H) en red, un ordenador, software SCADA InTouch, y driver ICeR (vía radio también realiza el telecontrol de los bombeos adyacentes).

23. Telecontrol de la red de abastecimiento y saneamiento de la Costa del Sol, Málaga

Telecontrol vía radio de un total de 6 estaciones remotas que conforman los bombeos y depósitos principales de la red de abastecimiento, telecontrol vía radio de 11 estaciones de reutilización de aguas residuales, y control de una estación depuradora (se han instalado tres repetidores para una coberturas de 200 Km). El control y supervisión se realiza mediante autómatas de Omron y Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

24. Telecontrol de Faros en el Ferrol, La Coruña

Telecontrol vía radio de 4 faros. El control y supervisión se realiza mediante Spiral 3000, un ordenador, software SCADA InTouch, y driver ICeR.

25. Telecontrol de Faros en Shangai, China

Red de comunicaciones híbrida radio / teléfono para controlar inicialmente 4 faros de distintos distritos de China. El control y supervisión se realiza mediante SP-5000, un ordenador en cada una de las tres centrales enlazadas vía teléfono, software SCADA InTouch, y driver ICeR.

26. Telecontrol de los bombeos de aguas residuales de L'Horta Nord, Valencia

Telecontrol y supervisión de 8 bombeos. El control y supervisión se realiza mediante autómatas de Alfa-Laval, un ordenador, y software SCADA SattControl.

27. Telecontrol de la red de abastecimiento y saneamiento del Bajo Almanzora, Almería

Telecontrol vía radio de 40 depósitos de la red de abastecimiento, automatización y telecontrol de 11 EDARs, telecontrol de 19 estaciones de bombeo de aguas residuales. El control y supervisión se realiza mediante autómatas de Siemens y Spiral 3000, ordenadores en red, software SCADA InTouch, y driver ICeR.

28. Telecontrol de la red de abastecimiento de agua de la ciudad de Córdoba

Telecontrol global vía radio de un total de 19 estaciones remotas que conforman los principales bombeos y depósitos que abastecen a la ciudad de Córdoba, y supervisión de la planta de tratamiento de agua potable. El control y supervisión se realiza mediante autómatas de Siemens (115U/95U), ordenadores en red, software SCADA InTouch, y driver ICeR.

29. Telecontrol de la red de bombeos de aguas residuales de Quart de Benager, Valencia

Automatización de la red escalonada de bombeos de aguas residuales mediante 7 estaciones remotas, enlazadas en modo dual radio/cable con la central. El control y supervisión se realiza mediante autómatas Allen Bradley, un ordenador, software SCADA InTouch, y driver ICeR.

30. Telecontrol de la red de bombeos de aguas residuales de Benidorm, Alicante

Automatización de bombeos de aguas residuales en Benidorm mediante 9 estaciones remotas. El control y supervisión se realiza mediante autómatas de Alfa-Laval, un ordenador, software SCADA InTouch, y driver ICeR.

31. Control y Supervisión de la línea de recorte en la Factoría Celupal en Algeciras, Cádiz

El sistema de control y supervisión para este proceso se realiza mediante autómatas de Siemens (115U) en red (Sinec-L1), un ordenador, y software SCADA InTouch.

32. Automatización de la ETAP de la Palma del Condado, Huelva

Control automático de la planta de tratamiento de agua potable. El control y supervisión se realiza mediante autómatas de Omron, un ordenador, y software SCADA InTouch.

33. Automatización y Supervisión de la EDAR de León

El control y supervisión de la EDAR se realiza mediante autómatas Allen Bradley, un ordenador, y software SCADA InTouch.

34. Automatización y Supervisión de la EDAR de Melilla

El control y supervisión de la EDAR se realiza mediante autómatas de Siemens, un ordenador en cada una de las centrales, y software SCADA InTouch.

35. Telecontrol de la red de abastecimiento de agua de La Bureba, Briviesca, Burgos

Telecontrol vía radio de 18 estaciones remotas que controlan los bombeos, depósitos, válvulas, instrumentación de campo, etc. que conforman la red de abastecimiento a la comarca de La Bureba. El control y supervisión se realiza mediante autómatas de Omron, un ordenador en cada central, software SCADA InTouch, y driver ICeR.

36. Telecontrol de la red de riego de Parques de Gandía, Valencia

Control y supervisión remoto del riego de 7 parques en Gandía. El control y supervisión se realiza mediante SP-500, un ordenador, software SCADA InTouch, y driver ICeR.

37. Telecontrol del Faro de Cap Bengut, Argelia

Telecontrol de un faro vía línea telefónica en Argelia. El control y supervisión se realiza mediante SP-500, un ordenador, software SCADA InTouch, y driver ICeR.

38. Automatización y telecontrol de los regadíos de Jumilla, Murcia

Control y Supervisión con el Sistema Hidrobus (® ICR) de la red de riego de la zona regable de Jumilla, con 7 Sectores, y un total de 1850 estaciones remotas. Además de medidas de caudal y presión en puntos clave de la red (35 puntos de control), y automatización, telecontrol e integración en el sistema de 8 sondeos, 5 balsas y 1 partidor. En el control y supervisión de todo estos sistemas se utiliza, entre otros equipos, el sistema SP-500 como concentrador de comunicaciones y el driver ICeR.

Apéndice B:

Glosario de términos

ActiveX	Un tipo de objeto COM, con código asociado que realiza una determinada función de forma totalmente integrada dentro de la aplicación que se está desarrollando
AGBAR	Sociedad General de Aguas de Barcelona
AGs	Algoritmos Genéticos
AI	Autómata Industrial
API	<i>Application Programmable Interface</i> . Interfaz de programación de aplicaciones
ASIC	<i>Application Specific Integrated Circuit</i> . Circuito integrado de propósito específico
C.H.G.	Confederación Hidrográfica del Guadalquivir
CAN	<i>Control Area Network</i> . Bus de campo desarrollado por la firma alemana Bosch en colaboración con Intel para ser empleado en el sector automovilístico, y que gracias a sus prestaciones se ha extendido también a los entornos industriales
CE	Computación Evolutiva
CIM	<i>Computer Integrated Manufacturing</i> . Fabricación integrada por computador
CIME	<i>Computer Integrated Manufacturing & Engineering</i>
CN	Controladores numéricos
COM/DCOM	<i>Component Object Model / Distributed Component Object Model</i> , tecnología de Microsoft para desarrollar aplicaciones distribuidas
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i> . Acceso múltiple por detección de portadora y con detección de colisiones.
DDE	<i>Dynamic Data Exchange</i> . Intercambio dinámico de datos
DLL	<i>Dynamic Data Library</i> . Librería dinámica de datos
DNA	<i>Distributed interNet Application</i> . Arquitectura para el desarrollo de aplicaciones altamente escalables, fácilmente portables, multi-capas, intrínsecamente distribuidas tanto en datos, procesos y plataformas de red, y capaz de integrar los entornos de desarrollo Cliente/Servidor y los basados en Internet/Intranet
EDAR	Estación Depuradora de Aguas Residuales
EMASAGRA	Empresa Municipal de Agua y Saneamiento de Granada

ERP	<i>Enterprise Resource Planning</i> . Planificación de Recursos Empresariales o paquetes de software comerciales para finanzas, distribución, manufactura, recursos humanos, etc., que permiten a las empresas trabajar de una forma más eficiente y productiva (SAP, Baan, Oracle, etc.)
ETAP	Estación de Tratamiento de Agua Potable
FMS	<i>Fieldbus Message Specification</i> .
GIS	Geographical Information Systems. Sistema de Información Geográfica
GPIB	<i>General Purpose Instrument Bus</i> . Interfaz paralela estándar IEEE 488 que se usa para conectar sensores e instrumentos a un computador
GRAF CET	GRÁfico Funcional de Control de Etapas y Transiciones, también denominado SFC (<i>Sequential Function Chart</i>)
GSM	<i>Global System for Mobile communications</i> . Sistema global para comunicaciones móviles
ICeR	Interfaz de Comunicaciones con estaciones Remotas
ICR	Ingeniería y Control Remoto S.A.
IEC	<i>International Electrotechnical Commission</i> .
LAN	<i>Local Area Network</i> . Red de área local
LED	<i>Light Emitter Diode</i> . Diodo emisor de luz
LLC	<i>Logical Link Control</i> . Control de enlace lógico. Subcapa de la capa de enlace de datos del RM-OSI, a propuesta del comité IEEE 802
MAC	<i>Medium Access Control</i> . Control de acceso al medio. Subcapa de la capa de enlace de datos del RM-OSI, a propuesta del comité IEEE 802
MAP	<i>Manufacturing Automation Protocols</i> , MAP-IEEE 802.4
MES	<i>Manufacturing Execution System</i> . Es un software que proporciona una gestión global de los recursos y actividades de fabricación de una empresa.
MMI	<i>Man Machine Interface</i> . Interfaz hombre máquina
MO	Módulo de Operaciones del driver de comunicaciones ICeR
MRP	<i>Manufacturing Resource Planning</i> . Es un software que mantiene información de los distintos niveles de fabricación de una empresa.
NA	Contacto Normalmente Abierto
NC	Contacto Normalmente Cerrado
NetDDE	Sistema que permite enlazar por DDE aplicaciones que se ejecutan en computadores conectados en red o vía modems
NPD	Neuro-Programación Dinámica
OCX	<i>OLE Custom control</i> . Módulo de programa independiente accesible por otros programas en entorno Windows
ODBC	<i>Open DataBase Connectivity</i> . Conectividad con bases de datos abiertas
OLE	<i>Object Linking and Embedding</i> . Interfaz para integrar información desde un programa a otra aplicación
OPC	<i>OLE for Process Control</i> . OLE para control de procesos
OSI	<i>Open Systems Interconnection</i> . Interconexión de sistemas abiertos
PC	<i>Personal Computer</i> . Computador personal

PID	Controlador proporcional integral derivativo
PLC	<i>Programmable Logic Controller</i> . Controlador Lógico Programable. También es conocido como “Autómata Programable”
PLD	<i>Programmable Logic Device</i> . Dispositivo lógico programable
POO	Programación Orientada a Objetos
PVM	<i>Parallel Virtual Machine</i> . Software que permite trabajar en paralelo con un conjunto de computadores (Máquina Virtual Paralela)
RA	Red de Alta en la distribución de agua de la ciudad de Granada
RAD	Regulación Automática de Depósitos
RB	Red de Baja en la distribución de agua de la ciudad de Granada
RDBMS	<i>Relational Database Management System</i> . Sistema Gestor de Bases de Datos Relacional
RNAs	Redes Neuronales Artificiales
RS	Red de Saneamiento en el abastecimiento de agua de la ciudad de Granada
RTC	Red Telefónica Conmutada
SCADA	<i>Supervision Control And Data Acquisition</i> . Control de Supervisión y Adquisición de Datos. Se suele utilizar como denominación genérica de los programas para la supervisión y el control de los procesos industriales
SCD	Sistema de Control Distribuido
SMS	<i>Short Message Service</i> . Servicio de mensajes cortos en GSM
SPDT	<i>Single Pole Double Throw</i> . Polo único doble contacto
SPIRAL	Arquitectura de red que permite el telecontrol vía radio de diferentes estaciones desde una estación central.
SQL	<i>Structured Query Language</i> . Lenguaje estructurado de consultas
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
TOP	<i>Technical and Office Protocols</i> , TOP-IEEE 802.3
UDP	<i>User Datagram Protocol</i> . Alternativa al protocolo TCP en la capa de transporte (4) del modelo OSI
VBA	<i>Visual Basic for Applications</i> . Visual Basic para aplicaciones. Subconjunto de Visual Basic que provee un lenguaje de macros común para aplicaciones de Microsoft
VME	<i>Versa Module Europe</i> . Bus flexible y abierto que hace uso del estándar <i>Eurocard</i> .
VRP	Válvula Reductora de Presión
WAN	<i>Wide Area Network</i> . Red de área extensa
XMODEM	Protocolo orientado a caracteres para transferencia de archivos

Referencias Bibliográficas

- [Ali-98] Aliaga, C.; Griñó, R.; Barracó, M.: *"Optimización de las Estrategias de Control de Redes de Transporte Hidráulico"*. Ingeniería del Agua, Vol.5, No.2, pp.76-82, Junio 1998.
- [All-88] Alla, P.; Jarrige, P.-A.: *"Optimal control of the West Parisian Area Water Supply Network"*. Computer Applications in Water Supply, Vol.2, pp.376-391, Research Studies Press Ltd, Letchworth, 1988.
- [And-95] Anderson, T.; Culler, D.; Patterson, D.: *"A case for NOW (Networks of Workstations)"*, IEEE Micro, 15 (1), 1995.
- [Ayz-98] Ayza, J.: *"Evolución del software de control. La integración vertical como objetivo"*. Automática e Instrumentación, pp. 101-116, Abril 1998.
- [Ayz-99] Ayza, J.: *"Software de adquisición, supervisión y control. Una evolución permanente"*. Automática e Instrumentación, pp. 117-131, Septiembre 1999.
- [Bac-97] Bäck, T.; Hammel, U.; Schwefel, H.-P.: *"Evolutionary Computation: comments on the history and current state"*. IEEE Trans. on Evolutionary Computation, Vol.1, No.1, pp.3-17, Abril 1997.
- [Bag-97] Baginski, A.; Covarrubias, G.: *"Open Control - the standard for PC-based automation technology"*. IEEE International Workshop on Factory Communication Systems, Edited by: J.M. Fuertes, G. Juanole, pp. 329-333, Barcelona, Octubre 1997.
- [Bal-97] Balcells, J.; Romeral J.L.: *"Autómatas programable"*. Marcombo. Barcelona, 1997.
- [Bel-62] Bellmann, R.; Dreyfus, S.: *"Applied Dynamic Programming"*. Princeton University Press, Princeton, N.J., 1962.
- [Ber-95a] Bernal, J.L. et al.: *"Diseño óptimo de redes de distribución de energía eléctrica utilizando algoritmos genéticos"*. IBERDROLA, Secretaría de I+D, 1995.
- [Ber-95b] Bertsekas, D.P.: *"Dynamic Programming and Optimal Control"*. Athena Scientific, Belmont, Massachusetts, 1995.
- [Ber-96] Bertsekas, D.P.; Tsitsiklis, J.N.: *"Neuro-Dynamic Programming"*. Athena Scientific, Belmont, Massachusetts, 1996.

- [Bla-90] Blacher, A.; Dabrowski, R. and Scholz-Reiter, B.: *"Analysis and design of information and communication structures in CIM"*. Computer Integrated Manufacturing Systems, 3, no. 3, pp. 135-140, 1990.
- [Bol-88] Bolling, J.G.; Duffie.: *"Computer Control of Machines and Processes"*. Addison-Wesley. Massachusetts, 1988.
- [Byw-99] Bywater, R.; Gehring, J.; Reinefeld, A.; Rippmann, F.; Weber, A.: *"Metacomputing in practice: a distributed compute server for pharmaceutical industry"*. Future Generation Computer Systems, Elsevier, Vol. 15, No.5-6, pp.769-786, Octubre 1999.
- [Bzu-89] Bzura, J.J.: *"Radio control of water heaters in Rhode Island"*. IEEE Trans. Power Syst., 4(1), pp. 26-29, 1989.
- [Cou-92] Coulbeck, B.; Orr, C.H.: *"On-line control of a city water supply and distribution system"*. In B.Coulbeck and E.P. Evans (eds), Pipeline Systems, Kluwer Academic Publishers, Dordrecht, pp. 261-270, 1992.
- [Cou-95] Coulbeck, B.: *"Optimización del régimen de operación de las estaciones de bombeo para reducir los costes energéticos"*. En Mejora del rendimiento y de la fiabilidad en sistemas de distribución de agua, pp. 271-296, Aguas de Valencia, U.D. Mecánica de fluidos (ISBN: 84-89487-00-6), 1995.
- [Cre-88] Creasey, J.D.: *"Pump Scheduling in Water Supply: More Than a Mathematical Problem"*. Computer Applications in Water Supply, Vol.2, pp. 279-289, Research Studies Press Ltd, Letchworth, 1988.
- [Cri-96] Crites, R.H.; Barto, A.G.: *"Improving elevator performance using reinforcement learning"*. Advances in Neural Information Processing Systems 8, MIT Press, pp. 1017-1023, 1996.
- [Cri-97] Crispin, A.J.: *"Programmable Logic Controllers and their Engineering Applications"*. McGraw-Hill UK, 1997.
- [Cul-99] Culler, D.E.; Singh, J.P.; Gupta, A.: *"Parallel Computer Architecture. A Hardware/Software Approach"*. Morgan Kaufmann Publishers, inc. ISBN:1-55860-343-3, 1999.
- [Dam-00a] Damas, M.; Tabasco, A.; Gomez, F.; Olivares, G.: *"Sistema de telegestión integral de zonas regables extensas: Automatización de regadíos en Jumilla"*. II Symposium Nacional los regadíos españoles, Madrid, Febrero 2000.
- [Dam-00b] Damas, M.; Salmerón, M.; Fernández, F.J.; Ortega, J.; Olivares G.: *"Parallel Control of Water Supply Networks in Clusters of Computers"*. 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI'2000), Orlando(USA), Julio 2000.
- [Dam-00c] Damas, M.; Salmerón, M.; Ortega, J.: *"ANNs and Gas for predictive controlling of water supply networks"*. The IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN2000), Como (Italia), Julio 2000.
- [Dam-00d] Damas, M.; Salmerón, M.; Diaz, A.; Ortega, J.; Olivares, G.; Prieto, A.: *"Genetic Algorithms and Neuro-Dynamic Programming: Application to Water Supply Networks"*. 2000 Congress on Evolutionary Computation (CEC2000), San Diego (USA), Julio 2000.

- [Dam-98] Damas, M.; Valor, R.C.; Gómez, F.; Olivares, G.: *"Algunos aspectos innovadores del Sistema de Telecontrol de la Red de Abastecimiento de Agua de Granada"*. Jornadas Técnicas Aguas Subterráneas y Abastecimiento de Urbano. Canal Isabel II. Abril 1998.
- [Dam-99a] Damas, M.; Gómez, F.; Prieto, A.; Olivares, G.: *"Un Sistema de Control Distribuido para Redes de Abastecimiento de Agua"*. SEID'99, pp.135-144, Febrero 1999.
- [Dam-99b] Damas, M.; Fernández, F.J.; Salmerón, M.; Ortega, J.: *"Procesamiento Paralelo en el Control de Redes de Distribución Hidráulica con Algoritmos Genéticos y Programación Dinámica"*. X Jornadas de Paralelismo, Servicio de Publicaciones de la Universidad de Murcia. Dep. legal MU.1549-1999, pp. 285-290, Septiembre 1999.
- [Dat-94] Data, R.S.N.; Sridharan, K.: *"Parameter Estimation in Water Distribution Systems by Least Squares"*. Journal of Water Resources Planning and Management Div., ASCE, 120(4), pp. 405-421, 1994.
- [Deb-93] Debauche B.: *"The CIME Computing Environment (CCE)"*. 1993
- [Fog-00] Fogel, D.B.: *"What is evolutionary computation?"*. IEEE Spectrum, pp.26-32, Febrero, 2000.
- [Fog-95] Fogel, D.B.: *"Evolutionary computation: Toward a new philosophy of machine intelligence"*. IEEE Press, Piscataway, NJ, 1995.
- [Fol-95] Folgado, M.A.: *"Detección directa de fugas. Metodologías utilizadas y eficacia de las mismas. La experiencia en la ciudad de Valencia"*. En Mejora del rendimiento y de la fiabilidad en sistemas de distribución de agua, pp. 125-135, Aguas de Valencia, U.D. Mecánica de fluidos (ISBN: 84-89487-00-6), 1995.
- [Fue-92] Fuertes, J.M.: *"Comunicaciones Industriales"*. Automática e Instrumentación, No 228, pp. 79-85, Noviembre 1992.
- [Gei-94] Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; Sunderam, V.: *"PVM: Parallel Virtual Machine. A Users's Guide and Tutorial for Networked Parallel Computing"*. The MIT Press, Cambridge, Massachusetts, London, England, 1994.
- [Gen-99] Gentsch, W.: *"Metacomputing: from workstation clusters to internet computing"*. Future Generation Computer Systems (Special issue), Elsevier, Vol. 15, No 5-6. Octubre 1999.
- [Ger-95] Germanopoulos, G.: *"La regulación óptima de válvulas en aras de una minimización de fugas"*. En Mejora del rendimiento y de la fiabilidad en sistemas de distribución de agua, pp. 211-240, Aguas de Valencia, U.D. Mecánica de fluidos (ISBN: 84-89487-00-6), 1995.
- [Gil-99] Gil, E.; Fuertes J.M.: *"Low cost distributed control system with protocol implemented by software"*. 7th IEEE International Conference on Emerging Technologies and factory Automation, Edited by: J.M. Fuertes, pp. 1225-1233, Barcelona, Octubre 1999.
- [Gis-93] Gistau, R.; García M.: *"Previsión de la demanda en los sistemas de abastecimiento de agua"*. En Abastecimientos de Agua Urbanos. Estado actual y tendencias futuras, pp. 41-64, Generalitat Valenciana (ISBN: 84-482-0200-7), 1993.

- [Gol-89] Goldberg, D.E.: *"Genetic Algorithms in Search, Optimization & Machine Learning"*. Addison-Wesley Co., Inc, Reading, MA, 1989.
- [Gom-91] Gómez, F.; Olivares, G.; Prieto, A.; Gázquez, J.A.: *"SPIRAL: a microprocessor-based telemetry and telecontrol system"*. Microcomputer Applications, Vol. 10, Nº 1, pp 22-28, 1991.
- [Gom-94] Gómez F.: *"SPIRAL: Un sistema de telecontrol distribuido"*. Tesis doctoral, Universidad de Granada, 1994.
- [Gor-93] Gordon, S.; Whitley, D.: *"Serial and parallel genetic algorithmns as function optimizers"*. Proc. of the Fifth Int. Conference on Genetic Algorithms, pp.177-183, San Mateo, 1993.
- [Hal-98] Halsall, F.: *"Comunicación de datos, redes de computadores y sistemas abiertos"*. Addison-Wesley, ISBN: 0-201-65307-9, 1998.
- [Haw-99] Hawick, K.A.; et al.: *"DISCWorld: an environment for service-based metacomputing"*. Future Generation Computer Systems, Vol. 15, pp. 623-635. Octubre 1999.
- [Hei-96] Heitkotter, J.; Beasley, D.: *"The Hitch-Hiker's Guide to Evolutionary Computation"*. January 1996.
- [Her-92] Hernández, A.: *"Depuración de aguas residuales"*. Paraninfo, S.A. ISBN: 84-380-0051-7. Segunda edición, Febrero 1992.
- [Hil-95] Hilera, J.R.; Martínez, V.J.: *"Redes Neuronales Artificiales. Fundamentos, modelos y aplicaciones"*. RA-MA. Madrid, 1995.
- [Iec-93] IEC 1131-3: *"Programable controllers - Part 3: Programming languages"*. International Standard, First edition, 1993.
- [Jac-91] Jack, D.W.; Thomas, B.; Devine, G.: *"Pump scheduling and cost optimization"*. Civil Engineering System, Vol.8 Nº4, pp. 197-206, 1991.
- [Jan-91] Janikov, C.Z.; Michalewicz, Z.: *"An experimental comparison of binary and floating pointrepresentation in genetic algorithms"*. In Proc. 4th Int. Conf. Genetic Algorithms, Morgan Kaufman, San Francisco, pp.31-36, 1991.
- [Jar-91] Jarrige, P.-A.; Harding, T.: *"Using optimization for integrated network managemen"*.. Civil Engineering System, Vol.8 Nº4, pp. 241-246, 1991.
- [Jar-93] Jarrige, P.-A.: *"Control óptimo de las redes de distribución. Repaso a las diferentes técnicas y su aplicación práctica"*. En Abastecimientos de Agua Urbanos. Estado actual y tendencias futuras, pp. 401-428, Generalitat Valenciana (ISBN: 84-482-0200-7), 1993.
- [Koh-88] Kohonen, T.: *"An Introduction to Neural Computing"*. Neural Networks, Vol. 1, No.1, pp. 3-16, 1988.
- [Kru-99] Kruglinski, D.J.; Shepherd, G.; Wingo, S.: *"Programación avanzada con Microsoft Visual C++"*. McGraw-Hill, ISBN: 84-481-2265-8, 1999

- [Kuo-96] Kuo, B.J.: *"Sistemas de control automático"*. Séptima edición. Prentice-Hall, ISBN: 968-880-723-0, 1996.
- [Kwo-95] Kwong, W.A.; Passino, K.M.; Laukonen, E.G.; Yurkovich, S.: *"Expert supervision of fuzzy learning systems for fault tolerant aircraft control"*. IEEE Proceeding, vol. 83, no. 3, pp. 466-483, 1995.
- [Lam-94] Lambert, A.: *"Accounting for losses: the bursts and background concept"*. J. Instn Wa. & Envir. Mangt, 8,2, pp. 205-214, 1994.
- [Lig-93] Liggett, J.A.: *"Localización de fugas por seguimiento de la red en régimen estacionario y no estacionario"*. En Abastecimientos de Agua Urbanos. Estado actual y tendencias futuras, pp. 321-343, Generalitat Valenciana (ISBN: 84-482-0200-7), 1993.
- [Mam-74] Mamdani, E.H.: *"Applications of fuzzy algorithms for control of simple dynamic plant"*. Proc. IEE, 121 (12), 1585-1588, 1974.
- [Man-92] Mandado, E.; Marcos, J.; Pérez S.A.: *"Controladores lógicos y autómatas programables"*. Marcombo. Barcelona, 1992.
- [Mar-90] Maren, A.J.; Harston C.T.; Pap R.M.: *"Handbook of Neural Computing Applications"*. Academic Press, 1990.
- [Mar-99a] Martínez, J.L.: *"Evaluation of New Open Control Systems for Light Assembly Applications"*. Martínez Lastra, J. L. MSc. Thesis. Tampere University of Technology, 1999.
- [Mar-99b] Martínez, J.L.; Tuokko, R.: *"Sistemas de control basados en tecnología PC. Estado del arte y tendencias a corto plazo"*. Automática e Instrumentación, pp. 62-72, Septiembre 1999.
- [McC-43] McCulloch, W.S.; Pitts, W.: *"A logical calculus of the ideas inmanent in nervous activity"*. Bulletin of Mathematical Biophysics, Vol. 5, pp. 115-133, 1943.
- [Men-95] Mendel, J.M.: *"Fuzzy Logic Systems for Engineering: A tutorial"*. IEEE Proceedings (Special Issue on Engineering Applications of Fuzzy Logic), pp.345-377, Marzo 1995.
- [Mic-94] Michalewicz, Z.: *"Genetic Algorithms + Data Structures = Evolution Programs"*. Second, Extended Edition. New York, 1994.
- [Mic-96] Microchip Technology Inc.: *"PIC16/17 Microcontroller Data Book"*. 1996.
- [Mir-95] Mira, J.; Delgado, A.E.; Boticario, J.G.; Díez, F.J.: *"Aspectos básicos de la inteligencia artificial"*. UNED, Editorial Sanz y Torres, S.L. ISBN: 84-88667-13-2. 1995.
- [Nea-99] Neary, M.O.; Christiansen, B.O.; Cappello, P.; Schausser, K.E.: *"Javelin: Parallel computing on the internet"*. Future Generation Computer Systems, Vol. 15, pp. 659-674. Octubre 1999.
- [Nec-89] NEC Electronics Inc.: *" μ PD70320/322 (V25), μ PD70330/332 (V35): 16-Bit, Single-Chip CMOS Microcomputers"*. User's Manual, Febrero 1989.

- [Pal-98] Pallás, R.: *"Sensores y acondicionadores de señal"*. Tercera edición, Marcombo, ISBN: 84-267-1171-5. Barcelona 1998.
- [Par-94] Paradigm Systems : *"Paradigm DEBUG/RT-V25, User's Guide"*. 1994.
- [Pom-99] Pomares, H.: *"Nueva metodología para el diseño automático de sistemas difusos"*. Tesis doctoral, Universidad de Granada, 1999.
- [Pre-92] Press, W.; Teukolsky, S.; Vetterling, W.; Flannery, B.: *"Numerical Recipes in C, Second Edition"*. Cambridge University Press, 1992.
- [Pri-96] Prieto, A.: *"Una visión actual sobre la Neurocomputación"*. Conferencia Inaugural del Curso 1996-97 en la ETSI de Ingeniería Informática. Servicio de Publicaciones de la Universidad de Granada, GR/1.060/96, 1996.
- [Rah-80] Rahal, C.M.; Sterling, M.; Coulbeck, B.: *"Parameter tuning for simulation models of water distribution networks"*. Proc. Instn. Civ. Engrs. Part 2, 1980.
- [Ren-96] Renders, J.M.; Flasse, S.P.: *"Hybrid Methods using Genetic Algorithms for Global Optimization"*. IEEE Trans. On Systems, Man, and Cybernetics (Part B), Vol.26, No.2, pp.243-258, Abril 1996.
- [Rum-88] Rumelhart, D.; Hinton, G.; Williams, R.: *"Learning representations by back-propagating errors"*. Nature, 323, pp. 533-536, 1986. Reimpreso en el texto *"Neurocomputing"* (J.Anderson y E.Rosenfeld ed.), MIT Press, 1988.
- [Sak-93] Sakaguchi, S.; Skai, I.; Haga, T.: *"Application of fuzzy logic to shift scheduling method for automatic transmission"*. Proc. 2nd IEEE Conf. On Fuzzy Syst., pp. 52-58, 1993.
- [Sal-99] Salmerón, M; Anguita, M.; Damas, M.; Gil, C.; Ortega, J.: *"Predicción de series temporales en 'clusters' de computadores"*. X Jornadas de Paralelismo, Servicio de Publicaciones de la Universidad de Murcia. Dep. legal MU.1549-1999, pp. 95-100, 1999.
- [Sav-95] Savic, D.A.; Walters, G.A.: *"Integration of a Model for Hydraulic Analysis of Water Distribution Networks with an Evolution Program for Pressure Regulation"*. Microcomputers in Civil Engineering, Vol.10, 10-3, pp. 219-229, 1995.
- [Sch-92] Scholz, B.: *"CIM Interfaces. Concepts, Standars and problems of Interfaces in Computer-Integrated Manufacturing"*. Chapman & Hall. London, 1992.
- [Sie-91] Siemens AG.: *"Simatic S5. Manual autómatas programables S5-115U"*. Siemens® 1991.
- [Sip-90] SIPEX: *"Line Interface Products RS-232/RS-422"*. 1990.
- [Spr-92] Spragins, J.D.; Hammond, J.L.; Pawlikowski, K.: *"Telecommunications: protocols and design"*. Addison-Wesley, ISBN: 0-201-09290-5, 1992.
- [Ter-95] TERN, Inc.: *"TinyDrive™. A C/C++ programable 16-bit controller with 42 I/O lines"*. Technical Manual 1995.
- [Tes-95] Tesauo, G.J.: *"Temporal difference learning and TD-Gammon"*. Communications of the ACM, Vol. 38, pp.58-68, 1995.

- [Tes-96] Tesauro, G.; Galperin, G.R.: *"On-line policy improvement using Monte-Carlo search"*. En *Advances in Neural Information Processing*, 9, pp. 1068-1074, MIT Press, 1996.
- [Tho-97] Thompson, L.: *"Industrial Data Communications"*. Instrument Society of America, ISBN: 1-55617-585-X, 1997.
- [Tri-94] Trillas, E.: *"Fundamentos e Introducción a la Ingeniería <<Fuzzy>>"*. OMRON Electronics S.A., Madrid, 1994.
- [Val-92] Valenzano, A.; Demartini, C.; Ciminiera, L.: *"MAP and TOP Communications"*. Addison Wesley Publishing Company, 1992.
- [Val-99] Valor, R.C.; Damas, M.; Gomez, F.; Olivares, G.: *"Predicción con redes neuronales del volumen de agua diario demandado por una ciudad"*. Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI99), Universidad Pontificia Comillas, ISBN: 84-699-0923-1, pp.123-126, Septiembre 1999.
- [Vel-95] Vela, A.F.; Espert, V.; Fuertes, V.: *"Panorama general de la identificación y reducción de agua no contabilizada en sistemas de distribución de agua"*. En *Mejora del rendimiento y de la fiabilidad en sistemas de distribución de agua*, pp. 81-122, Aguas de Valencia, U.D. Mecánica de fluidos (ISBN: 84-89487-00-6), 1995.
- [Wal-83] Walski, T.M.: *"Technique for Calibrating Network Model"*. *Journal of Water Resources Planning and Management Div.*, ASCE, 109(4), pp. 360-372, 1983.
- [Wan-94] Wang, L.X.: *"Adaptive fuzzy systems and control. Design and stability analysis"*. Prentice Hall, 1994.
- [Wat-89] Watkins, C.J.C.H.: *"Learning from Delayed Rewards"*. PhD Thesis, Cambridge University, 1989.
- [Wat-92] Watkins, C.J.C.H.; Dayan, P.: *"Q-Learning"*. *Machine Learning*, Vol.8, pp. 279-292, 1992.
- [Wid-90] Widrow, B. and Lehr, M.: *"30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation"*. *Proc. IEEE*, vol. 78, no. 9, pp. 1415-1441, 1990.
- [Wil-92] Williams, R.I.: *"Handbook of scada systems"*. 1st Edition, Elsevier Advanced Technology, 1992.
- [Won-99a] Wonderware Corporation: *"Wonderware® FactorySuite InTouch: User's Guide"*. Revision C, Julio 1999.
- [Won-99b] Wonderware Corporation: *"Wonderware® FactorySuite: System Administrator's Guide"*. Revision D, Julio 1999.
- [Yag-92] Yager, R.R.; Zadeh, L.A.: *"An Introduction to Fuzzy Logic Applications in Intelligent Systems"*. Kluwer Academic Publishers, Boston, 1992.
- [Yas-85] Yasunobu, S.; Miyamoto, S.: *"Automatic train operation by predictive fuzzy control"*. *Industrial Application of Fuzzy Control*, M. Sugeno Ed. Amsterdam: North Holland, pp. 1-18, 1985.

- [Yen-95] Yen, J.; Lanagri, R.; Zadeh, L. (Eds.): *"Industrial Applications of Fuzzy Logic and Intelligent Systems"*. IEEE Press, 1995.
- [Zad-65] Zadeh, L.A.: *"Fuzzy Set"*. Information and Control. Vol.8, pp. 338-353, 1965.
- [Zad-96] Zadeh, L.A.: *"The roles of fuzzy logic and soft computing in the conception, design, and deployment of intelligent systems"*. Discurso pronunciado en el acto de su investidura de Doctor Honoris Causa por la Universidad de Granada. Servicio de Publicaciones de la Universidad de Granada, GR/696/96, 1996.
- [Zha-95] Zhang, W.; Dietterich, T.G.: *"High-performance job-shop scheduling with a time-delay TD(λ) network"*. Advances in Neural Information Processing Systems, 8, MIT Press, pp.1024-1030, 1996.