ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA SCADA PARA LA PRODUCCIÓN DE HARINA EN LA EMPRESA MOLINO ELECTRO MODERNO S.A.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y CONTROL

ISAAC PÁSTOR ACOSTA
Email: ipastor@yahoo.com

DIRECTOR: MSc. PATRICIO CHICO H.
Email: pchico@fie.edu.ec

Quito, Junio 2008

DECLARACIÓN

Yo Isaac Pástor Acosta, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

_____

Isaac Pástor Acosta

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Isaac Pástor Acosta, bajo mi supervisión**.**

_____

MSc. Patricio Chico H.
DIRECTOR DEL PROYECTO

AGRADECIMIENTOS

Agradezco a todo el personal de la empresa Molino Electro Moderno S.A. por la ayuda y sugerencias que me brindaron durante el desarrollo de este proyecto.

De manera especial agradezco al Ing. Rodrigo López por permitirme realizar la tesis en la empresa que dirige y al Sr. Agustín Cueva por toda su ayuda y colaboración.

Agradezco además al Ing. Patricio Chico por su contribución en la dirección de esta tesis.

CONTENIDO

INDICE DE FIGURAS..... viii
INDICE DE TABLAS..... x
RESUMEN..... xii
PRESENTACIÓN..... xiii
CAPÍTULO 1
1 INTRODUCCIÓN
1.1 PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO..... 1
1.1.1 DIAGRAMA DE BLOQUES DEL PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO..... 2
1.1.2 DESCRIPCIÓN DEL PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO..... 3
1.1.3 DIAGRAMAS ESQUEMÁTICOS DEL PROCESO DE PRODUCCIÓN DE HARINA..... 9
1.2 NECESIDADES DE LA EMPRESA MOLINO ELECTRO MODERNO S. A..... 11
1.2.1 SITUACIÓN ACTUAL DE LA PLANTA..... 11
1.2.1.1 Panel OP-1124..... 12
1.2.1.2 Panel OP-1224 13
1.2.1.3 Panel OP-1510..... 14
1.2.1.4 OM9001 Communication Master..... 14
1.2.1.5 Modo de operación de la sección de molienda..... 15
1.2.1.6 Modo de operación de la sección de limpieza..... 17
1.2.2 PROBLEMAS RELACIONADOS AL MONITOREO..... 18
1.2.3 CARACTERÍSTICAS DEL SISTEMA SCADA..... 19
1.2.4 REQUERIMIENTOS DE LA EMPRESA..... 20
1.3 SUSTENTACIÓN TEÓRICA..... 21
1.3.1 SISTEMAS SCADA..... 21
1.3.1.1 Prestaciones del software de un sistema SCADA..... 22
1.3.1.2 Requisitos de un sistema SCADA..... 22
1.3.1.3 Componentes del sistema..... 23
1.3.1.3.1 Unidad de Terminal Remota (RTU) 23
1.3.1.3.2 Estación Maestra..... 24
1.3.1.3.3 Infraestructura y Métodos de Comunicación..... 24
1.3.1.4 Funciones principales de un sistema SCADA..... 25
1.3.1.4.1 Supervisión remota de instalaciones y equipos..... 25

1.3.1.4.2 Control remoto de instalaciones y equipos..... 26
1.3.1.4.3 Procesamiento de datos..... 26
1.3.1.4.4 Visualización gráfica dinámica..... 26
1.3.1.4.5 Generación de reportes..... 26
1.3.1.4.6 Representación se señales de alarma..... 26
1.3.1.4.7 Almacenamiento de información histórica..... 27
1.3.1.4.8 Programación de eventos..... 27
1.3.2 PROTOCOLO OPC..... 27
1.3.2.1 Introducción..... 27
1.3.2.2 Objetivo de la creación del protocolo..... 27
1.3.2.3 Antecedentes para la creación de OPC..... 28
1.3.2.4 Características..... 29
1.3.2.4.1 Conectividad abierta..... 29
1.3.2.4.2 Especificaciones estándares..... 30
1.3.2.5 Especificaciones OPC actuales y en desarrollo..... 32
1.3.2.5.1 OPC Data Access(Acceso de datos)..... 32
1.3.2.5.2 OPC Alarms & Events (Alarmas y eventos)..... 32
1.3.2.5.3 OPC Batch..... 32
1.3.2.5.4 OPC Data Exchange (intercambio de datos)..... 32
1.3.2.5.5 OPC Historical Data Access (acceso de datos históricos)..... 33
1.3.2.5.6 OPC Security (seguridad)..... 33
1.3.2.5.7 OPC XML-DA..... 33
1.3.2.5.8 OPC Complex Data (Datos complejos)..... 33
1.3.2.5.9 OPC Commands (comandos)..... 34
1.3.2.5.10 OPC Unified Architecture (arquitectura unificada)..... 34

CAPÍTULO 2

2 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

2.1 INTRODUCCIÓN..... 35
2.2 ESTUDIO DE LA PLANTA DE PRODUCCIÓN DE MOLINO ELECTRO MODERNO S.A..... 36
2.2.1 CONEXIÓN DE LOS PLC CON LOS COMPUTADORES..... 36
2.2.2 ACTUADORES DE LA ETAPA DE LIMPIEZA..... 37
2.2.3 SENSORES DE LA ETAPA DE LIMPIEZA..... 38
2.2.4 SENSORES Y ACTUADORES DE LA ETAPA DE MOLIENDA..... 40
2.2.5 SENSORES DE LA ETAPA DE MOLIENDA..... 42
2.3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA..... 44

2.3.1 DISEÑO DE LA BASE DE DATOS..... 44
2.3.1.1 Pasos para diseñar una base de datos..... 44
2.3.1.1.1 Reunión de información..... 44
2.3.1.1.2 Desarrollo de un diseño lógico..... 45
2.3.1.1.2.1 Determinar entidades 45
2.3.1.1.2.2 Convertir las entidades en tablas..... 45
2.3.1.1.2.3 Determinación de las relaciones y de la cardinalidad..... 45
2.3.1.1.2.4 Dibujo de diagramas de relaciones..... 46
2.3.1.1.2.5 Validación del diseño conceptual..... 47
2.3.1.1.3 Conversión a un modelo físico..... 47
2.3.1.1.3.1 Establecer claves primarias..... 47
2.3.1.1.3.2 Establecer las claves foráneas..... 48
2.3.1.1.3.3 Establecer tipos de datos..... 49
2.3.1.2 Diseño de la base de datos..... 49
2.3.1.2.1 Reunión de información..... 49
2.3.1.2.2 Desarrollo de un diseño lógico..... 49
2.3.1.2.2.1 Determinar entidades..... 49
2.3.1.2.2.2 Convertir las entidades en tablas..... 51
2.3.1.2.2.3 Determinación de las relaciones y de la cardinalidad..... 52
2.3.1.2.2.4 Dibujo de diagramas de relaciones..... 53
2.3.1.2.3 Conversión a un modelo físico..... 54
2.3.1.2.3.1 Establecer claves primarias..... 54
2.3.1.2.3.2 Establecer las claves foráneas..... 54
2.3.1.2.3.3 Establecer tipos de datos..... 54
2.3.1.3 Creación de la base de datos..... 56
2.3.2 COMUNICACIÓN ENTRE VISUAL BASIC Y POSTGRESQL..... 57
2.3.2.1 Implementación de la comunicación entre la aplicación y la base de datos..... 58
2.3.3 MODOS DE OPERACIÓN DEL SOFTWARE DEL SISTEMA SCADA..... 63
2.3.3.1 Modo de supervisión..... 63
2.3.3.2 Modo de control..... 63
2.3.4 DEFINICIÓN DEL NÚMERO DE VENTANAS QUE SE UTILIZARÁN Y LA FUNCIÓN DE CADA UNA..... 64
2.3.4.1 Ventana de administración del programa..... 65
2.3.4.2 Ventana para el ingreso de clave..... 65
2.3.4.3 Ventana para agregar usuarios..... 65
2.3.4.4 Ventana para eliminar usuarios..... 65

2.3.4.5 Ventana para cambio de contraseña..... 66
2.3.4.6 Ventanas del proceso..... 66
2.3.4.7 Ventana para ingresar el motivo que ocasionó una alarma..... 66
2.3.4.8 Ventanas para mostrar registros de la base de datos..... 67
2.3.5 CONTROL DE ACCESO Y ADMINISTRACIÓN DE USUARIOS..... 67
2.3.5.1 Implementación del control de acceso y administración de usuarios..... 68
2.3.6 SEGURIDADES PARA EVITAR EL APAGADO Y ENCENDIDO NO DESEADO DE MÁQUINAS..... 69
2.3.6.1 Implementación de seguridades para evitar el apagado y encendido no deseado de máquinas..... 60
2.3.7 COMUNICACIÓN ENTRE LOS COMPUTADORES Y LOS PLCS..... 70
2.3.7.1 Cálculo del tráfico en la red..... 71
2.3.7.1.1 Cálculo del tráfico de red para la sección molienda..... 72
2.3.7.1.2 Cálculo del tráfico de red para la sección limpieza..... 73
2.3.7.2 Cableado necesario para la comunicación..... 73
2.3.7.3 Configuración del servidor OPC..... 74
2.3.7.3.1 Configuración del canal de comunicación..... 75
2.3.7.3.2 Agregar un dispositivo al canal de comunicación..... 78
2.3.7.3.3 Agregar etiquetas..... 83
2.3.7.4 Configuración del cliente OPC..... 84
2.3.7.4.1 Código para conectarse con servidor OPC..... 85
2.3.7.4.2 Código para agregar el grupo..... 85
2.3.7.4.3 Código para agregar elementos..... 85
2.3.7.5 Método utilizado para leer y escribir información..... 86
2.3.7.5.1 Lectura automática de datos que han cambiado..... 87
2.3.7.5.2 Escritura de datos..... 88
2.3.7.5.3 Lectura manual de datos..... 89
2.3.8 DESARROLLO DE LAS VENTANAS DE INTERFASE DEL OPERADOR... 91
2.3.8.1 Desarrollo de las imágenes..... 91
2.3.8.1.1 Iconos de la barra de tareas..... 91
2.3.8.1.2 Imágenes bmp..... 91
2.3.8.1.3 Imágenes gif con fondo transparente..... 92
2.3.8.2 Implementación de animaciones..... 92
2.3.8.2.1 Realización de GIFs animados..... 92
2.3.8.2.2 Realización de animación mediante programación en Visual Basic..... 94
2.3.8.3 Implementación de la barra de herramientas y menú..... 94

2.3.8.3.1 Barra de herramientas..... 95
2.3.8.3.2 Funciones de los iconos de la barra de herramientas..... 95
2.3.8.3.3 Menú..... 99
2.3.8.3.4 Funciones del menú..... 100
2.3.8.4 Botones para prender y apagar motores e indicar su estado..... 102
2.3.8.4.1 Implementación de botones para prender y apagar motores e indicar su estado. 103
2.3.8.5 Encendido y apagado de motores..... 104
2.3.8.5.1 Implementación de encendido y apagado de motores con arranque directo..... 105
2.3.8.5.2 Implementación del encendido y apagado de motores con arranque estrella – triángulo..... 105
2.3.8.5.3 Implementación de arranque y parada de motor con arrancador suave..... 107
2.3.8.5.4 Implementación de encendido y apagado de motores que se debe realizar la operación de manera simultanea..... 107
2.3.8.5.5 Implementación de encendido y apagado de motores que se debe esperar un intervalo de tiempo entre el apagado y encendido..... 108
2.3.8.6 Apertura y cierre de las compuertas..... 109
2.3.8.6.1 Implementación del control de las compuertas..... 110
2.3.8.7 Implementación de sensores..... 112
2.3.8.8 Implementación de alarmas..... 113
2.3.9 DESCRIPCIÓN DE LAS VENTANAS DEL PROGRAMA Y SUS FUNCIONES..... 117
2.3.9.1 Ventana de administración del programa..... 118
2.3.9.2 Ventana para el ingreso del nombre de usuario y clave..... 120
2.3.9.3 Ventana para agregar usuarios..... 123
2.3.9.4 Ventana para eliminar usuarios..... 124
2.3.9.5 Ventana para cambio de contraseña..... 126
2.3.9.6 Ventanas de operación..... 128
2.3.9.6.1 Ventanas de la etapa de limpieza..... 129
2.3.9.6.2 Ventanas de la etapa de molienda..... 131
2.3.9.7 Ventana para ingresar el motivo que ocasionó una alarma..... 132
2.3.9.8 Ventana para mostrar el registro de los eventos ocurridos..... 134
2.3.9.9 Ventana para mostrar el registro de las alarmas ocurridas..... 137
2.3.10 IMPLEMENTACIÓN DE OTRAS FUNCIONES..... 138
2.3.10.1 Código utilizado para ingresar a modo de control 139
2.3.10.2 Implementación del registro de eventos y alarmas en la base de datos..... 140

2.3.10.2.1 Registro de eventos..... 141
2.3.10.2.2 Registro de alarmas..... 142
2.3.10.3 Exportación de información a Microsoft Excel..... 143
2.3.10.4 Implementación de cuadro de información..... 145
2.3.10.5 Implementación del arranque y parada de la planta desde el computador.... 147
2.3.10.6 Implementación de detector de comunicaciones..... 150
2.3.10.7 Implementación de rutinas para cerrar la aplicación..... 151
2.3.10.8 Elaboración de mensajes en el panel de operador para indicar funcionamiento de SCADA..... 152

CAPÍTULO 3

3 PRUEBAS Y RESULTADOS

3.1 INTRODUCCIÓN..... 154
3.2 PRUEBAS DE COMUNICACIÓN..... 154
3.2.1 COMUNICACIÓN ENTRE LA BASE DE DATOS Y EL SOFTWARE DEL SISTEMA SCADA..... 154
3.2.2 COMUNICACIÓN ENTRE OPC SERVER Y PLC..... 155
3.2.3 COMUNICACIÓN ENTRE VISUAL BASIC Y OPC SERVER..... 156
3.2.4 DETECTOR DE COMUNICACIÓN ENTRE VISUAL BASIC Y SERVIDOR OPC..... 157
3.3 PRUEBAS DE OPERACIÓN..... 157
3.3.1 ENCENDIDO MANUAL DE MOTORES..... 158
3.3.2 APAGADO MANUAL DE MOTORES..... 158
3.3.3 ENCENDIDO AUTOMÁTICO DE LA ETAPA DE MOLIENDA..... 159
3.3.4 ENCENDIDO AUTOMÁTICO DE LAS ETAPAS DE PRE-LIMPIEZA, PRIMERA LIMPIEZA Y SEGUNDA LIMPIEZA..... 159
3.3.5 PARADAS DE LA ETAPA DE MOLIENDA..... 160
3.3.6 PARADA DE LAS ETAPAS DE PRE-LIMPIEZA, PRIMERA LIMPIEZA Y SEGUNDA LIMPIEZA..... 161
3.3.7 SEGURIDADES PARA EVITAR OPERACIONES NO DESEADAS..... 161
3.4 PRUEBAS DE SENSORES Y ALARMAS..... 162
3.5 PRUEBAS DE LAS FUNCIONES DEL SOFTWARE..... 163
3.5.1 MENÚ..... 163
3.5.2 BARRA DE HERRAMIENTAS..... 164
3.5.3 INGRESO DE USUARIO..... 165
3.5.4 CREACIÓN DE UN USUARIO..... 165
3.5.5 ELIMINACIÓN DE USUARIOS..... 166

3.5.6 CAMBIO DE CLAVE..... 166
3.5.7 SALIR DEL PROGRAMA..... 167
3.5.8 INDICADOR DE INFORMACIÓN..... 168
3.5.9 ANIMACIONES..... 168
3.6 PRUEBAS DE LECTURA DE REGISTROS DE LA BASE DE DATOS..... 169
3.7 PRUEBA DE EXPORTACIÓN DE INFORMACIÓN A MICROSOFT EXCEL.. 170
3.8 PRUEBA GENERAL DEL SISTEMA SCADA..... 172
CAPÍTULO 4
4 CONCLUSIONES Y RECOMENDACIONES
4.1 CONCLUSIONES..... 177
4.2 RECOMENDACIONES..... 179
REFERENCIAS BIBLIOGRÁFICAS..... 180
ANEXOS
ANEXO A: COMANDOS UTILIZADOS PARA LA CREACIÓN DE LA BASE DE DATOS
ANEXO B: ESPECIFICACIONES DEL CPU DEL PLC DIRECT LOGIC KOYO D4-450
ANEXO C: ESPECIFICACIONES DEL MÓDULO DE ENTRADAS DIRECT LOGIC KOYO D4-16ND2F
ANEXO D: ESPECIFICACIONES DEL MÓDULO DE SALIDAS DIRECT LOGIC KOYO D4-16TA
ANEXO E: ESPECIFICACIONES DEL MÓDULO DE SALIDAS DIRECT LOGIC KOYO D4-16TD1
ANEXO F: ESPECIFICACIONES DE LA BASE DE EXPANSIÓN DIRECT LOGIC KOYO D4-08B-1
ANEXO G: ESPECIFICACIONES DE LOS PANELES OPTIMATE
ANEXO H: ESPECIFICACIONES DEL CABLE DE PROGRAMACIÓN DE LOS PLCS DIRECT LOGIC KOYO D4-450
ANEXO I: ESPECIFICACIONES DEL SERVIDOR OPC KEPDIRECT
ANEXO J: DIAGRAMA DE CONEXIÓN DE LOS PANELES OPTIMATE CON LOS PLCS DIRECT LOGIC KOYO D4-450
ANEXO K: DIAGRAMAS DE CONEXIÓN DE LA SECCIÓN LIMPIEZA
ANEXO L: DIAGRAMAS DE CONEXIÓN DE LA SECCIÓN MOLIENDA
ANEXO M: DIAGRAMA P&ID DE LA SECCIÓN LIMIEZA
ANEXO N: DIAGRAMA P&ID DE LA SECCIÓN MOLIENDA

INDICE DE FIGURAS

CAPÍTULO 1

1.1. Diagrama de bloques del proceso de producción de harina de trigo **2**
1.2. Constitución del grano de trigo **3**
1.3. Limpiadora de trigo **4**
1.4. Selector de gravedad **4**
1.5. Dosificador de agua utilizado para incrementar la humedad del trigo **5**
1.6. Compartimentos de almacenamiento de trigo húmedo **5**
1.7. Bancos de molienda **6**
1.8. Purificador **7**
1.9. Dosificador de químicos **7**
1.10. Empacadora de harina **8**
1.11. Diagrama esquemático de la etapa de prelimpieza del proceso de producción de harina **9**
1.12. Diagrama esquemático de las etapas de primera y segunda limpieza del proceso de producción de harina **10**
1.13. Diagrama esquemático de la etapa de molienda del proceso de producción de harina **11**
1.14. Esquema de conexión de PLCs, paneles y elementos **12**
1.15. Panel OP-1124 ..... **13**
1.16. Panel OP-1224 ..... **13**
1.17. Panel OP-1510 ..... **14**
1.18. OM9001 Communication Master **15**
1.19. Esquema de los selectores y pulsadores del tablero de control de la sección molienda **16**
1.20. Esquema de los selectores y pulsadores del tablero de control de la sección limpieza **18**
1.21. Esquema sistema SCADA **21**

CAPÍTULO 2

2.1. Conexión entre los computadores y los PLCs **36**
2.2. Diagrama de conexión de sensores mediante matriz **38**
2.3. Esquema de conexión PLC limpieza **40**
2.4. Esquema de conexión PLC molienda **43**
2.5. Símbolos de cardinalidad **46**
2.6. Relación simple entre dos tablas **46**
2.7. Diagrama de relaciones de la base de datos **53**
2.8. Ventana Herramientas Administrativas de Windows **58**
2.9. Pestaña System DNS de la ventana ODBC Data Source Administrador **59**
2.10. Create New Data Source **59**
2.11. PostgreSQL ANSI ODBC Driver (psqlODBC) Setup **60**
2.12. Ventana mostrando en mensaje que indica que la conexión con la base de datos fue exitosa **61**
2.13. Conexión de los PLCs con los computadores **71**
2.14. Esquema del cable utilizado para comunicar los PLCs con los computadores **74**
2.15. Ventana principal del software KEPDirect for PLCs **75**
2.16. Ventana New Channel - Identification del software KEPDirect for PLCs **75**
2.17. Ventana New Channel – Device Driver del software KEPDirect for PLCs **76**
2.18. Ventana New Channel - Communications del software KEPDirect for PLCs **77**
2.19. Ventana New Channel – Write Optimizations del software KEPDirect for PLCs **77**
2.20. Ventana New Channel – Summary del software KEPDirect for PLCs **78**
2.21. Ventana principal de KEPDirect for PLCs con el enlace para agregar dispositivos **78**
2.22. Ventana New Device –Name del software KEPDirect for PLCs **79**
2.23. Ventana New Device – Model del software KEPDirect for PLCs **79**

2.24. Ventana New Device – ID del software KEPDirect for PLCs **80**
2.25. Ventana New Device – Timing del software KEPDirect for PLCs **80**
2.26. Ventana New Device – Demotion del software KEPDirect for PLCs **81**
2.27. Ventana New Device – Database Creation del software KEPDirect for PLCs **82**
2.28. Ventana New Device – Tag Import Settings Creation del software KEPDirect for PLCs **82**
2.29. Ventana New Device – Sumary del software KEPDirect for PLCs **83**
2.30. Ventana Tag Properties del software KEPDirect for PLCs **84**
2.31. Imágenes que conforman la animación de un tornillo sin fin que gira de izquierda a derecha **93**
2.32. Imágenes que conforman la animación de un tornillo sin fin que gira de derecha a izquierda **94**
2.33. Barra de herramientas del software del sistema SCADA para la sección molienda.. **95**
2.34. Barra de herramientas del software del sistema SCADA para la sección limpieza… **95**
2.35. Pantalla de molienda con motores visibles **97**
2.36. Pantalla de molienda sin motores visibles **98**
2.37. Menú del software del sistema SCADA **99**
2.38. Menú Archivo **100**
2.39. Menú velocidad animación **101**
2.40. Menú color fondo **101**
2.41. Menú ventana del software del sistema SCADA de la sección molienda **102**
2.42. Menú ventana del software del sistema SCADA de la sección limpieza **102**
2.43. Menú modo **102**
2.44. Mensaje indicando que no se debe esperar a que transcurra un intervalo de tiempo antes de encender el motor **109**
2.45. Pantalla de molienda indicando activación de alarma: PT410 ..... **114**
2.46. Mensaje indicando que hay alarmas activadas **115**
2.47. Sección del código escalera en el que se ve la acción del relé de control C41 ..... **117**
2.48. Ventana administración del programa con el botón ingresar **119**
2.49. Ventana administración del programa con todos los botones **119**
2.50. Mensaje indicando que se ha ingresado al sistema con acceso de administrador… **121**
2.51. Mensaje indicando que la clave ingresada es incorrecta **122**
2.52. Ventana ingreso de clave **122**
2.53. Ventana agregar usuarios **124**
2.54. Mensaje preguntado si el operador está seguro de eliminar un usuario **125**
2.55. Mensaje indicando que no se puede eliminar al usuario por ser el único registrado. **125**
2.56. Ventana eliminar usuarios **126**
2.57. Mensaje indicando que las claves ingresadas no son iguales **127**
2.58. Mensaje indicando que se realizó el cambio de clave **127**
2.59. Ventana cambio de contraseña **128**
2.60. Ventana para etapa de recepción, prelimpieza, almacenaje y trasilaje **129**
2.61. Ventana para etapa de primera y segunda limpieza **130**
2.62. Ventana para etapa de molienda **131**
2.63. Ventana para la etapa de ensacado **132**
2.64. Mensaje mostrando que el motivo ingresado no es del tamaño requerido **133**
2.65. Ventana ingreso de motivo **133**
2.66. Botón buscar **135**
2.67. Botón utilizado para exportar datos a Excel **136**
2.68. Ventana eventos **136**
2.69. Ventana alarmas **138**
2.70. Modificación realizada a diagrama escalera para deshabilitar selectores del tablero de control **140**
2.71. Cuadro utilizado para mostrar información **145**
2.72. Selectores y pulsadores de la pantalla de la sección molienda **148**
2.73. Selectores y pulsadores de la pantalla de la sección limpieza **148**

2.74. Diagrama escalera con relés de control en paralelo con señales de selectores del tablero de control..... ***149***
2.75. Indicación de comunicación correcta..... ***151***
2.76. Indicación de error en la comunicación..... ***151***
2.77. Mensaje preguntando al operador si desea cerrar la aplicación..... ***152***

CAPÍTULO 3

3.1. Programa OPC Quick Client mostrando la calidad y valor de algunas variables..... ***156***
3.2. Cuadro de información mostrando datos del motor M410..... ***169***
3.3. Registros de eventos mostrados en la aplicación del sistema SCADA..... ***171***
3.4. Registros de eventos exportados a Microsoft Excel..... ***172***

INDICE DE TABLAS

2.1. Salidas conectadas al PLC de la sección limpieza ***37***
2.2. Entradas conectadas al PLC de la sección limpieza a través de la matriz ***39***
2.3. Entradas conectadas directamente al PLC de la sección limpieza ***39***
2.4. Salidas conectadas al PLC de la sección molienda ***41***
2.5. Entradas conectadas al PLC de la sección molienda a través de la matriz ***42***
2.6. Entradas conectadas directamente al PLC de la sección molienda ***42***
2.7. Determinación de entidades ***50***
2.8. Conversión de entidades en tablas ***51***
2.9. Claves primarias de las tablas ***54***
2.10. Tipos de datos y restricciones de las columnas de la base de datos ***55***
2.11. Cálculo del número de tramas para la sección molienda ***72***
2.12. Cálculo del número de tramas para la sección limpieza ***73***

RESUMEN

Con el objetivo de facilitar el trabajo de los operadores y obtener registros de los eventos y las alarmas ocurridas, la empresa Molino Electro Moderno S.A. requirió la implementación de un sistema SCADA. La empresa solicitó cubrir sus requerimientos del momento y futuras ampliaciones utilizando la menor inversión posible.

En este trabajo de tesis se diseñó, desarrolló e implementó un sistema SCADA en la planta de producción de Molino Electro Moderno S.A. Para la implementación del sistema se crearon dos aplicaciones en Visual Basic, una para la etapa de limpieza y la otra para la etapa de molienda. Cada una de las aplicaciones corre en un computador independiente. La comunicación entre las aplicaciones y los PLCs se realiza utilizando el protocolo OPC mediante el servidor KEPDirect. Para guardar los registros de alarmas y eventos se creó una base de datos utilizando el software gratuito PostgreSQL.

Luego de la implementación del sistema se ha observado que su uso ha resultado beneficioso para la empresa. Las operaciones de encendido y apagado son más fáciles y rápidas de realizar y en las pantallas de los computadores se pueden determinar rápidamente las condiciones de funcionamiento de la maquinaria de la planta. Además, los registros de los eventos se pueden utilizar para programar el mantenimiento preventivo de la maquinaria y los registros de las alarmas para prevenir la reiteración de fallas.

PRESENTACIÓN

El presente documento consta de cuatro capítulos en los que se describe el proceso seguido para el desarrollo de proyecto. Al final del documento se encuentran los anexos donde se muestra información adicional.

El primer capítulo es una introducción al documento. En él se encuentran: una descripción del proceso de producción de harina de trigo, una descripción de los requerimientos de la empresa y un análisis de la situación de la planta antes de la implementación del sistema. Además se encuentra en este capítulo el fundamento teórico del proyecto, en el cual hay información referente a los sistemas SCADA y al protocolo OPC.

En el capítulo dos se describen el diseño y la implementación del sistema. Se muestra el proceso seguido en la creación de la base de datos y en la creación de las aplicaciones. También se muestra el desarrollo de la comunicación entre la base de datos y las aplicaciones y el desarrollo de la comunicación entre las aplicaciones y los PLCs.

En el tercer capítulo se describen las pruebas realizadas y los resultados obtenidos. Se describen las pruebas que se realizaron de manera independiente a cada función de las aplicaciones y las pruebas generales del sistema SCADA.

En el capítulo cuatro se encuentran las conclusiones obtenidas luego de realizar el proyecto y algunas recomendaciones para mejorar el sistema.

En los anexos se encuentran las especificaciones de los PLCs y paneles utilizados en la planta, los diagramas de conexión y otra información relevante.

1 INTRODUCCIÓN

1.1 PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO

En esta sección se explica el proceso de producción de harina de trigo. La descripción detallada del proceso se complementa con un diagrama de bloques. De este modo se pretende que el lector comprenda el proceso de transformación de la materia prima (trigo) en el producto final (harina).

1.1.1 DIAGRAMA DE BLOQUES DEL PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO

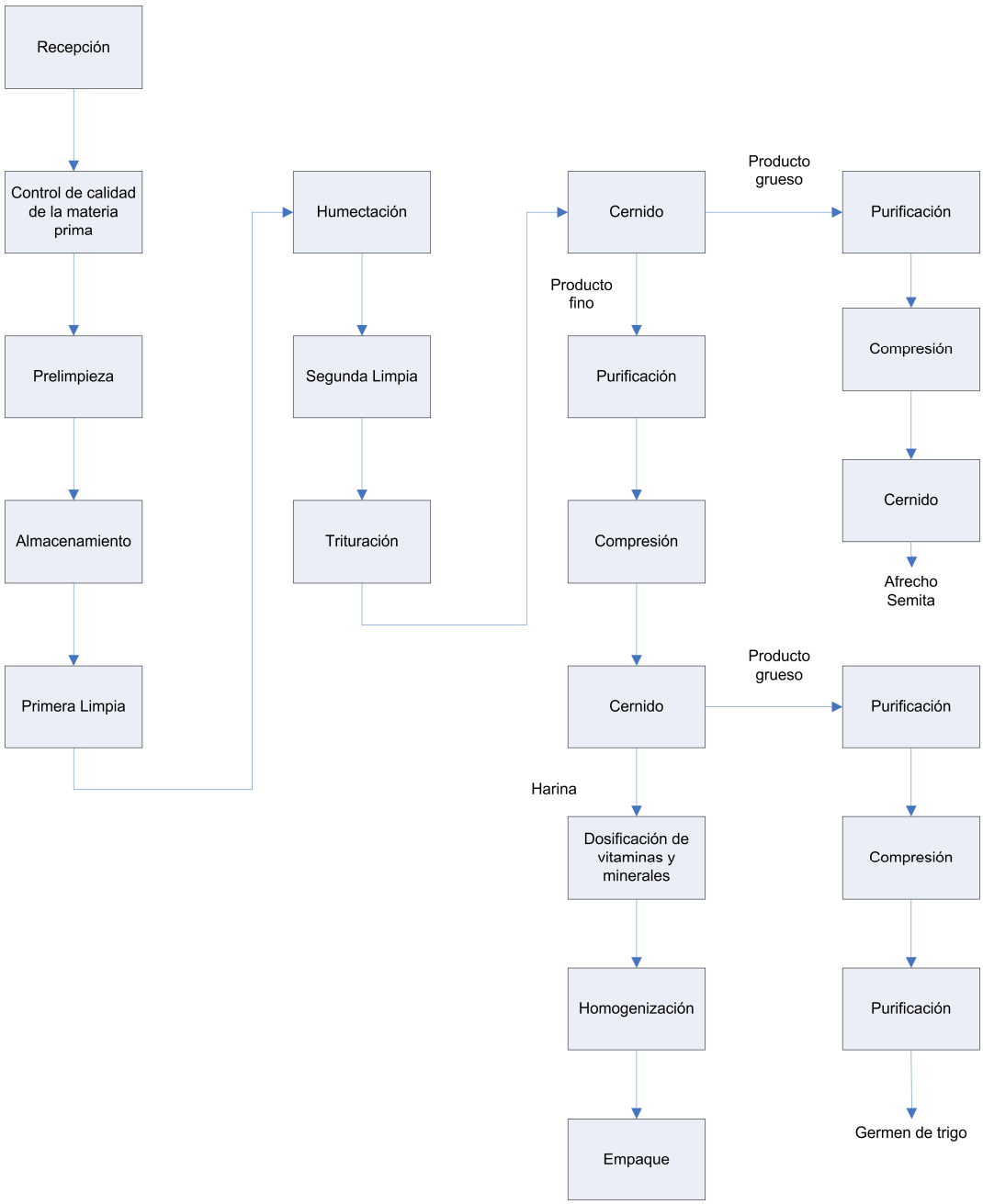


Figura 1.1. Diagrama de bloques del proceso de producción de harina de trigo.

1.1.2 DESCRIPCIÓN DEL PROCESO DE PRODUCCIÓN DE HARINA DE TRIGO

El proceso de molienda del trigo está compuesto por dos etapas: limpieza y molienda. La etapa de limpieza se subdivide en: prelimpieza, primera limpieza y segunda limpieza. El objetivo de este proceso es separar el endospermo del resto del grano de trigo. Esta parte del grano se convierte en harina que será utilizada en panadería, pastelería, pastas, etc. El endospermo constituye aproximadamente el 83 por ciento del grano[1]. Los componentes restantes del grano forman afrecho, semita y germen de trigo[2].

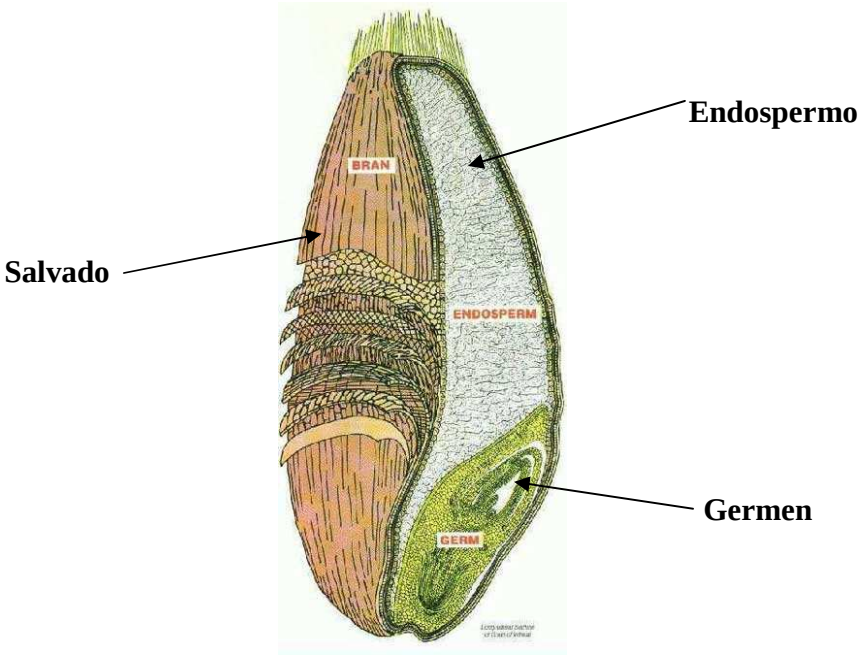


Figura 1.2. Constitución del grano de trigo.[1]

La materia prima contiene elementos extraños que deben ser separados del trigo antes de ser molido. En la pre-limpieza se utiliza un imán para remover cualquier elemento metálico mezclado con el trigo. Luego, en la primera limpieza, el cereal

pasa por un conjunto de tamices donde se eliminan las sustancias de tamaño superior o inferior al del trigo, y mediante aspiración se elimina el polvo contenido.

Figura 1.3. Limpiadora de trigo.[1]

A continuación el trigo pasa por un selector de gravedad utilizado para separar materiales basándose en peso o densidad. Esta máquina, además, posee un conjunto de cribas o tamices. De esta manera se eliminan rocas, palos, morocho y materiales muy livianos.

Figura 1.4. Selector de gravedad.[3]

Cuando el trigo se encuentra limpio se debe acondicionar para la molienda. Esto se consigue añadiendo agua para incrementar el porcentaje de humedad hasta alcanzar un valor entre 16 y 17 por ciento[2]. Luego el trigo es almacenado de 12 a 24 horas[2], dependiendo de la variedad y dureza del grano, en compartimientos adecuados antes de ser molido. Este procedimiento permite obtener una mejor separación de grano. Además, el grano se vuelve más suave, por lo que es necesario emplear menos fuerza en el proceso de molienda.

Figura 1.5. Dosificador de agua utilizado para incrementar la humedad del trigo.[1]

Figura 1.6. Compartimentos de almacenamiento de trigo húmedo.[1]

Después que el grano húmedo ha reposado, se realiza la segunda limpieza. Mediante una despuntadora, se elimina tierra y arena que se encuentra adherida al grano. Esta máquina también posee un imán para eliminar cualquier metal que se encuentre mezclado con el cereal.

A continuación, el trigo es trasladado hacia los bancos de molienda. En esta etapa se remueve el endospermo del trigo y se muele hasta alcanzar el tamaño de grano deseado. Este proceso se realiza mediante trituraciones y compresiones. Las trituraciones se realizan en bancos de molienda con rodillos estriados con el objetivo de separar el endospermo del grano. Los bancos de molienda utilizados para la compresión poseen rodillos lisos. Éstos se encargan de triturar el grano hasta alcanzar el tamaño deseado.

El proceso de molienda se puede resumir en moler y cernir. Este proceso se repite varias veces hasta que se ha removido por completo el endospermo del resto del grano. Luego de la primera trituración (primera molienda del cereal), todavía existe mucho endospermo adherido al salvado (capa externa del grano), por lo que será molido y cernido varias veces.

Figura 1.7. Bancos de molienda.[4]

El trigo es molido y luego cernido, los elementos gruesos son enviados nuevamente hacia los bancos de molienda, mientras los productos finos son trasladados hacia los purificadores y dosificadores de químicos.

Los purificadores utilizan cortinas de tela y flujo de aire para separar el salvado de las partículas de harina que son más finas. En este momento del proceso ya se ha obtenido harina pura. Antes de ser empacada, se utilizan dosificadores para agregar minerales y vitaminas a la harina para mejorar sus características.

Figura 1.8. Purificador.[4]

Figura 1.9. Dosificador de químicos.[4]

Es importante mencionar que de cada 60 libras de trigo, se obtienen aproximadamente 45 libras de harina y 15 libras de subproductos.[4] Estos subproductos son afrecho y semita, los cuales se utilizan como alimento de ganado.

Figura 1.10. Empacadora de harina.[4]

1.1.3 DIAGRAMAS ESQUEMÁTICOS DEL PROCESO DE PRODUCCIÓN DE HARINA

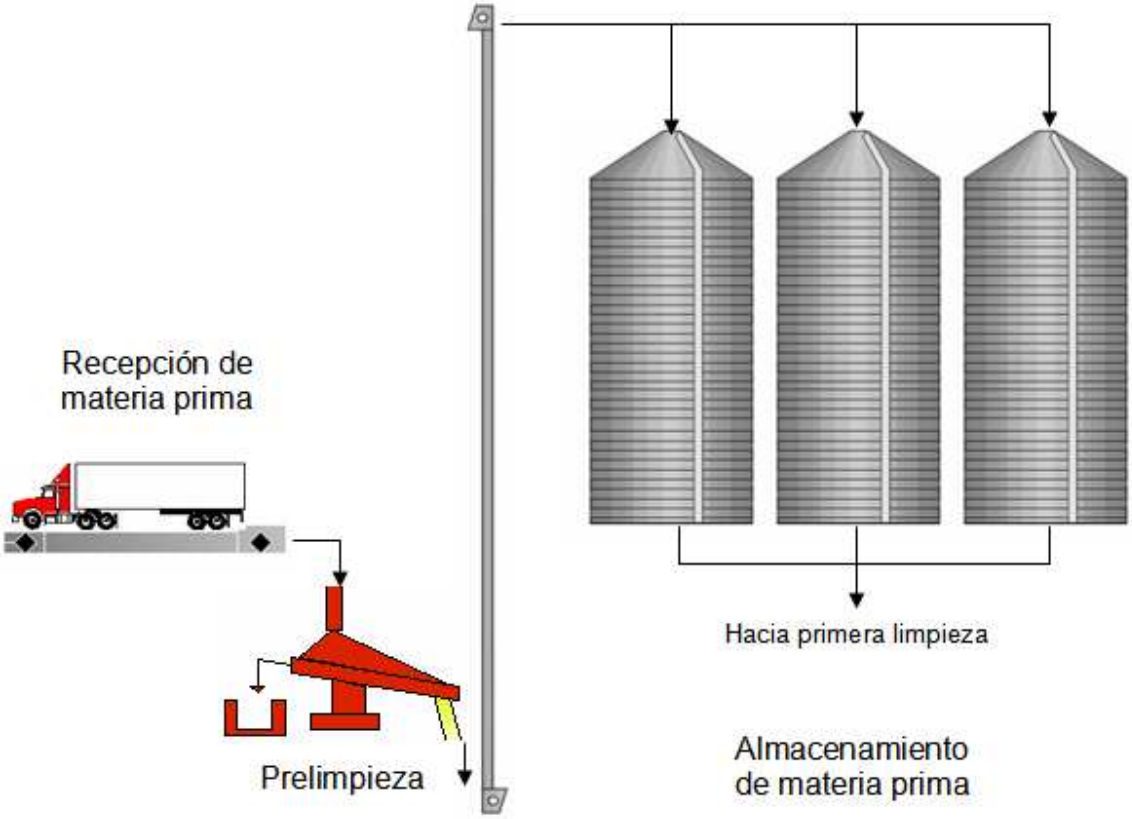


Figura 1.11. Diagrama esquemático de la etapa de prelimpieza del proceso de producción de harina

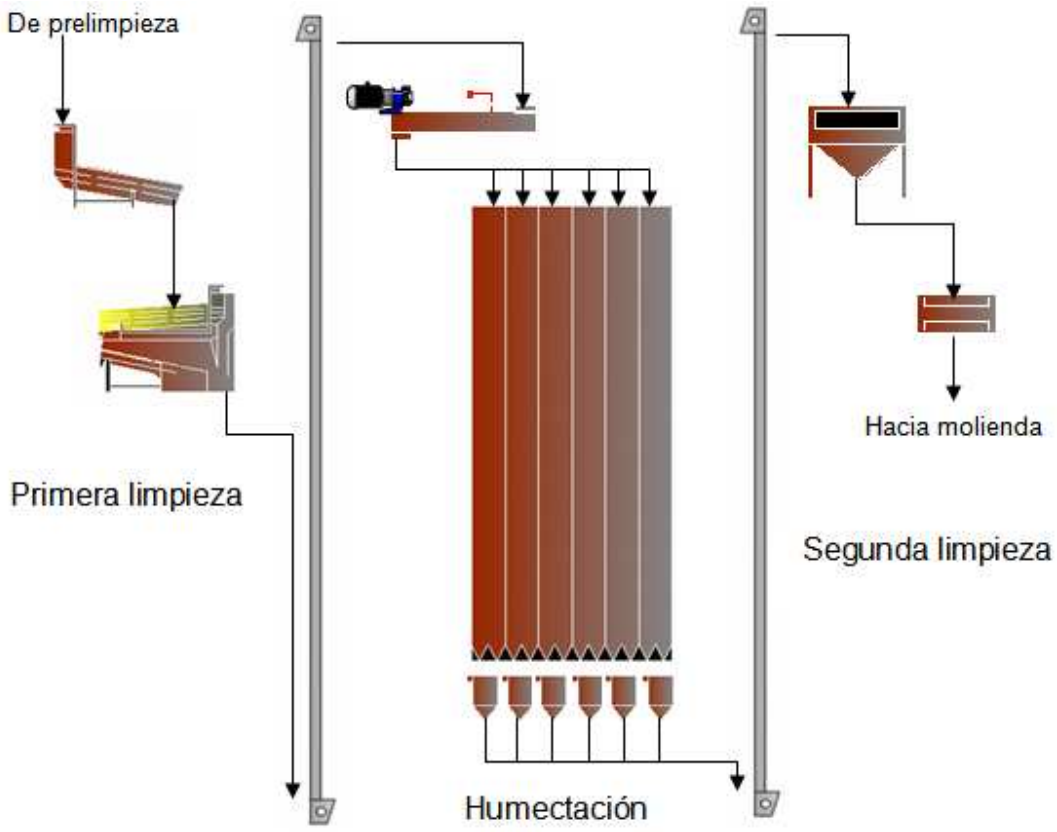


Figura 1.12. Diagrama esquemático de las etapas de primera y segunda limpieza del proceso de producción de harina

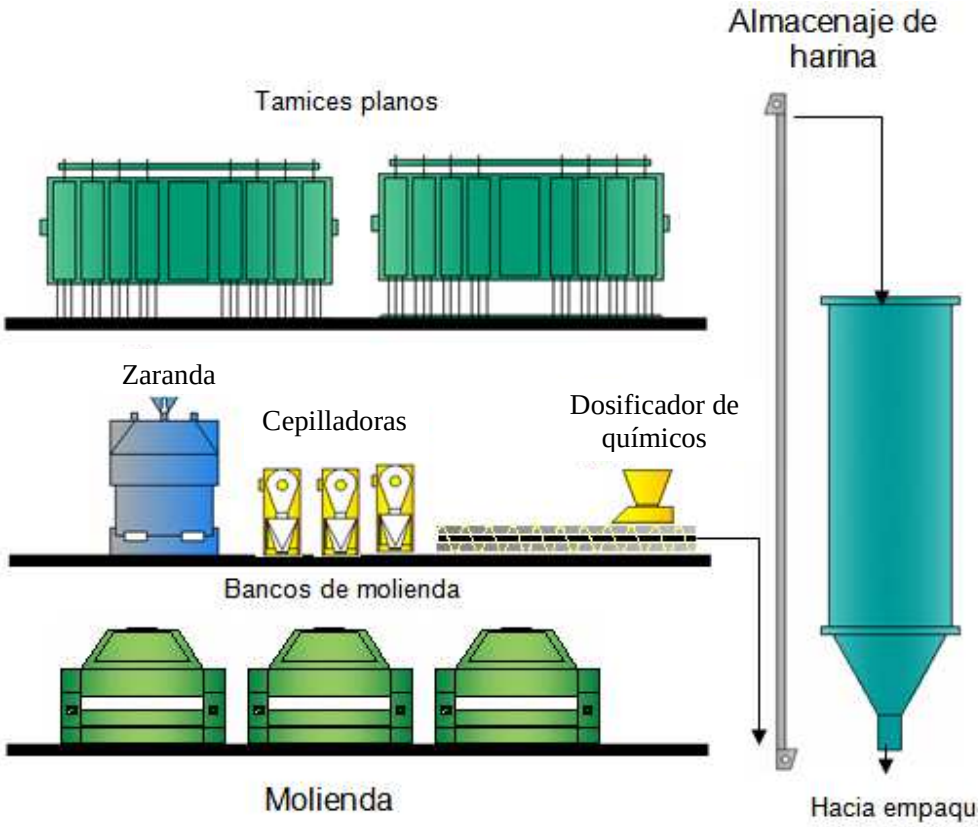


Figura 1.13. Diagrama esquemático de la etapa de molienda del proceso de producción de harina

1.2 NECESIDADES DE LA EMPRESA MOLINO ELECTRO MODERNO S. A.

1.2.1 SITUACIÓN ACTUAL DE LA PLANTA

La maquinaria de la empresa es controlada por dos PLCs *Koyo* modelo *D4 - 450*. El primero controla las etapas de limpieza y el otro la etapa de molienda. Para la visualización de información y el ingreso de datos se utilizan 3 paneles conectados a cada PLC. Los paneles y el PLC se comunican a través del modulo de comunicación *OM9001 Communication Master.* Los paneles utilizados son: *OP-1124*, *OP-1224* y

OP-1510. A continuación se muestra un esquema de la conexión de los PLCs con los paneles y con los elementos de entrada – salida.

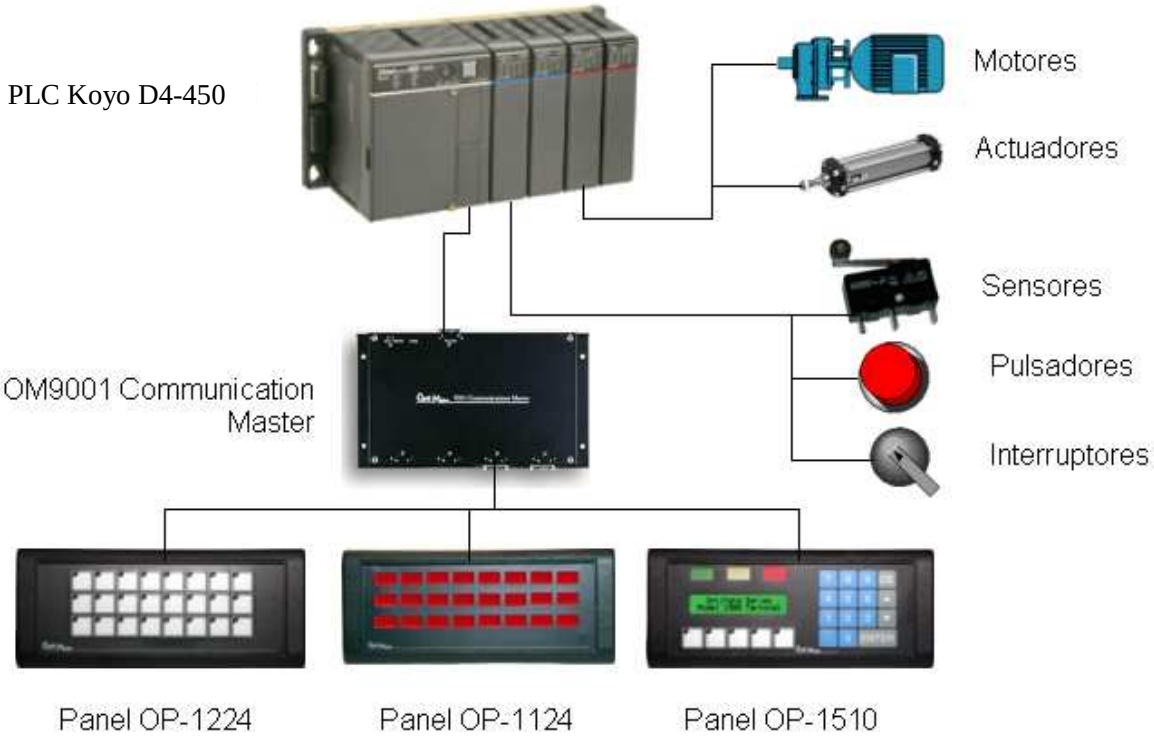


Figura 1.14. Esquema de conexión de PLCs, paneles y elementos.

1.2.1.1 Panel OP-1124

Este panel posee un total de veinticuatro indicadores luminosos distribuidos en tres filas y ocho columnas. Las luces se encienden y apagan según la señal enviada por el PLC. Se puede utilizar para indicar el estado de cualquier señal digital. En la fábrica Molino Electro Moderno S.A., este panel se utiliza para indicar que se activó un protector térmico y para indicar a que grupo de máquinas pertenece el motor correspondiente. Debido a los pocos indicadores que posee el panel no se puede mostrar el motor del cual se activo el protector, la señal luminosa solo indica el grupo

de máquinas al cual pertenece, por ejemplo, se prende una señal indicando que el protector pertenece a los bancos de molienda pero no muestra a cual de los motores.

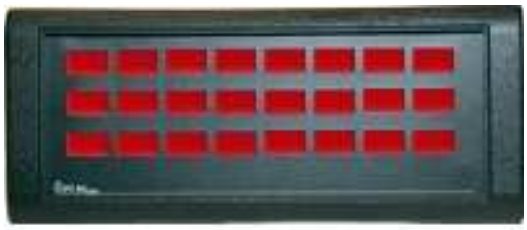


Figura 1.15. Panel OP-1124.[5]

1.2.1.2 Panel OP-1224

El panel OP-1224 posee tres filas de ocho pulsadores, cada uno con su correspondiente indicador luminoso. En el tablero de control de la etapa de molienda cada pulsador representa un motor, por lo que se puede determinar su estado y prenderlos y apagarlos rápidamente. El número de pulsadores no es suficiente para manejar todos los motores de la sección. Por este motivo, mediante este panel solo se pueden monitorear y controlar una parte de los motores empleados en la etapa de molienda. En la etapa de limpieza, este panel se utiliza para activar y desactivar seis electro válvulas que controlan la salida de trigo húmedo.

Figura 1.16. Panel OP-1224.[6]

1.2.1.3 Panel OP-1510

El panel OP-1510 posee varias funciones de ingreso de comandos y visualización de información. Entre ellas se encuentran: una pantalla LCD, un panel numérico, cinco pulsadores con indicador luminoso, 3 indicadores luminosos de diferentes colores, una tecla de cancelación (CE), una de aceptación (ENTER), una fecha hacia arriba y una hacia abajo. Este panel se utiliza para interactuar con el operador. En el se muestran mensajes para indicar la operación que está realizando el PLC, por ejemplo, el código del motor que está prendiendo o apagando, o el código del protector térmico que se activo. También se utiliza para arrancar y parar los motores de manera manual. Los pulsadores con indicación luminosa se emplean para notificar al PLC que se solucionó el problema que provocó una alarma.

Figura 1.17. Panel OP-1510.[7]

1.2.1.4 OM9001 Communication Master

El modulo OM9001 commnunication Master provee una interfase inteligente entre un PLC y módulos OptiMate. Se comunica en modo serial con el PLC transfiriendo datos en ambas direcciones. Este módulo permite la conexión de hasta 31 dispositivos OptiMate de las series 1000 o 600.[8]

El OM9001 contiene dos puertos seriales para comunicarse con los módulos a una velocidad de 19200 baudios. Los puertos seriales son RS422 y pueden conectarse a alguno de los 31 módulos. La distancia máxima entre el módulo y OP9001 es de 4000 píes (aproximadamente 1220 m).[8]

Figura 1.18. OM9001 Communication Master.[8]

1.2.1.5 Modo de operación de la sección de molienda

En la sección de molienda, además de los paneles descritos en la parte superior, hay tres selectores y cuatro pulsadores en el tablero de control. Los selectores permiten realizar las siguientes operaciones: selección de modo manual o automático, selección del tipo de acción: arranque o parada, y selección entre parada general y arranque general. Dos de los pulsadores cumplen la función de parada de emergencia total y parada de emergencia parcial. Otro pulsador se utiliza para borrar las señales de alarma del PLC y el último se encarga de activar un contactor que energiza el PLC y los paneles conectados a éste.

Cuando se selecciona parada general se interrumpe la alimentación del PLC apagando la maquinaria encargada de la molienda. Este selector funciona con una llave con el objetivo de que se retire cuando se van a realizar trabajos de mantenimiento para evitar el encendido imprevisto de máquinas. El selector de modo permite realizar el arranque secuencial de toda la maquinaria de la sección cuando está en modo automático. Cuando se encuentra en modo manual permite encender o apagar motores individualmente.

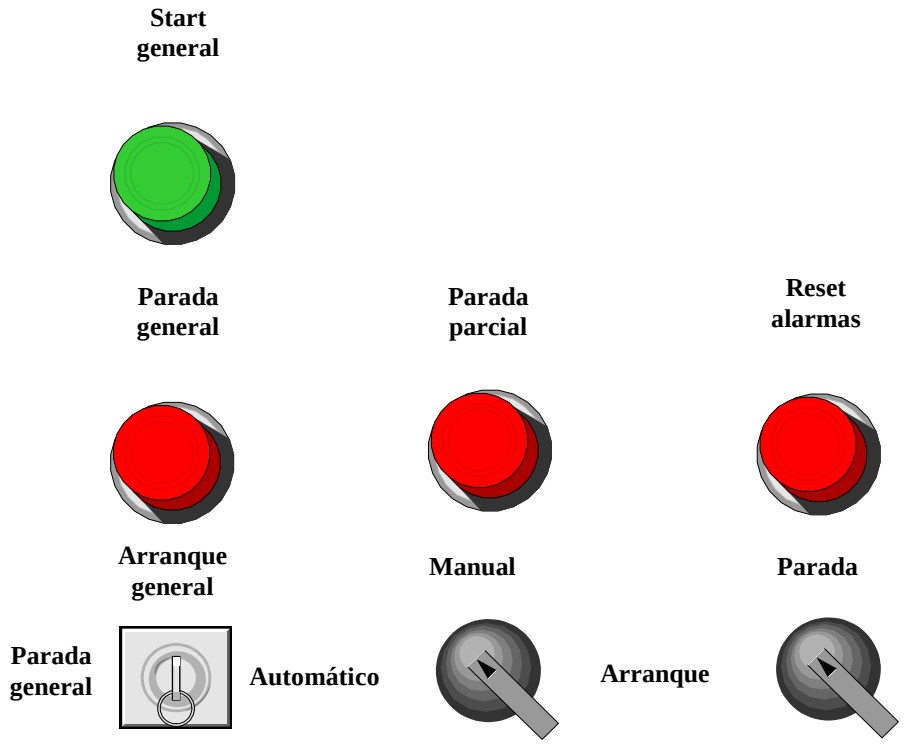


Figura 1.19. Esquema de los selectores y pulsadores del tablero de control de la sección molienda.

Mediante estos paneles y elementos se puede: encender y apagar motores, determinar si se ha disparado un protector térmico, determinar que motores están trabajando, e informar al PLC que se solucionó el problema que ocasionó una alarma.

El procedimiento que se debe seguir para encender un motor es: mediante las perillas seleccionar modo manual y operación arranque; ingresar mediante el panel OP-1510 el código del motor y pulsar la tecla de aceptación en el mismo módulo. Para apagar una máquina se sigue el mismo procedimiento pero se debe seleccionar la opción parada utilizando la perilla de selección de operación.

En el panel OP-1224, mediante los leds incorporados, se indica si los motores se encuentran prendidos o apagados. Utilizando este panel también se pueden prender y apagar fácilmente algunos de los motores mediante los pulsadores incorporados. Este método resulta más conveniente que el descrito anteriormente, pero al no poseer suficientes pulsadores e indicadores no se pueden monitorear ni controlar todos los motores de la sección.

Cuando se detecta una alarma, ya sea un protector térmico, un detector de giro, u otro, el PLC realiza una parada parcial o general dependiendo de la máquina involucrada, e informa al operador mediante una señal luminosa en el panel OP-1124 y un mensaje de texto en le panel OP-1510. Luego de solucionar el problema que provocó la alarma, el operador debe presionar una tecla para informar al PLC que puede continuar con la operación normal.

1.2.1.6 Modo de operación de la sección de limpieza

En el tablero de control de esta sección, además de los paneles conectados al PLC, hay tres selectores. Los cuales se utilizan para el encendido y apagado de la maquinaria encargada de realizar los procesos de limpieza de la materia prima. Los selectores activan y desactivan las etapas de prelimpieza, primera limpieza y segunda limpieza. Cuando se activa una de las etapas, el PLC se encarga de encender todas las compuertas y motores necesarios para ejecutar dicha operación. En este tablero también hay tres pulsadores mediante los cuales se realiza la parada de cada etapa.

Existen además dos pulsadores que activan y desactivan un contactor encargado de proveer energía al PLC y a los paneles. Al pulsar el botón *General Start,* se energizan dichos elementos. Mientras que al activar el pulsador *General Stop* se elimina el suministro de energía.

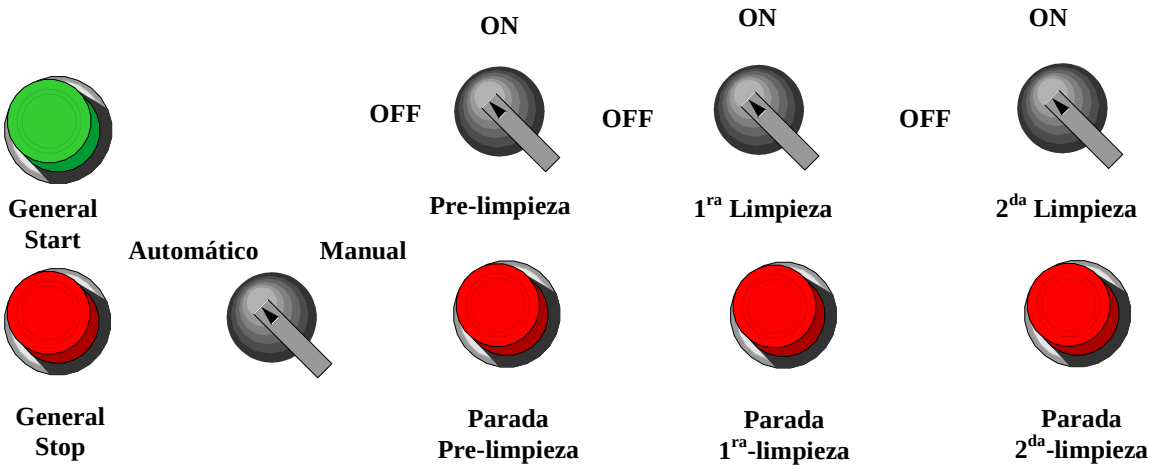


Figura 1.20. Esquema de los selectores y pulsadores del tablero de control de la sección limpieza

Las operaciones de encendido y apagado de motores, y el monitoreo de alarmas, motores y protectores térmicos, se realizan de la misma manera que en la sección de molienda. Estos procedimientos están descritos en la sección *modo de operación de la sección molienda.*

1.2.2 PROBLEMAS RELACIONADOS AL MONITOREO

Supervisar el comportamiento de la planta mediante estos módulos presenta varios inconvenientes. Por ejemplo: no es posible llevar datos históricos de funcionamiento, prender y apagar motores es un procedimiento demorado y complicado, no se puede determinar rápidamente que motores están funcionando, y no se puede conocer el estado actual de los sensores.

El método de monitoreo, utilizado antes de la instalación del sistema SCADA, no permite llevar un registro de alarmas, arranques, paradas, errores de operación, etc. Estos datos históricos son muy importantes para programar el mantenimiento preventivo de la maquinaria. Además, esta información se puede utilizar para

determinar que tan seguido ocurre una misma alarma con el objetivo de realizar modificaciones en la planta para disminuir su ocurrencia.

El procedimiento para prender y apagar motores, descrito anteriormente, es demorado de realizar. Esto representa un grave problema en caso de que fuese necesario realizar estas operaciones rápidamente. Si una máquina presenta algún problema, y se debe apagar rápido, es necesario realizar una parada de emergencia, a pesar que el proceso pueda continuar sin ella.

Otro problema relacionado al monitoreo es que no se puede determinar rápidamente cuales son los motores que se encuentran funcionando en un determinado momento ni el estado de los sensores. Los módulos instalados solo permiten visualizar que grupo de motores está funcionado, mas no permiten determinar el estado individual de cada motor. Además, solo se puede determinar el estado de algunos de los sensores.

1.2.3 CARACTERÍSTICAS DEL SISTEMA SCADA

Para facilitar la labor del operador durante las etapas de arranque, funcionamiento y parada de la planta, se va a implementar un sistema de supervisión, control y adquisición de datos (SCADA, por sus siglas en inglés), en la planta de producción de la empresa Molino Electro Moderno S. A.

Las características que debe poseer el sistema para solventar las necesidades de la empresa son las siguientes:

- Pantallas de interfase de operador para el manejo de los silos.
- Pantallas de interfase de operador para acondicionamiento de materia prima
- Pantallas de interfase de operador para molienda que permitan la supervisión y control de la producción de harina, afrecho, semita y subproductos.

- Pantallas de interfase de operador que permita realizar el arranque y paro del molino de manera manual y automática.
- Almacenamiento de datos con información referente a la operación de la maquinaria, tal como, horas de funcionamiento, registro de paradas y arranques y los motivos que los ocasionaron.

La empresa necesita un sistema de bajo costo y fácil mantenimiento que permita agregar actuadores y sensores a medida que la planta crezca. Además, se debe utilizar un protocolo que permita una comunicación rápida y segura, entre los PLCs y el computador.

1.2.4 REQUERIMIENTOS DE LA EMPRESA

Molino Electro Moderno S.A. requiere que el sistema se implemente utilizando el software Visual Basic 6.0 para la programación y PostgreSQL 8.2 para la creación y mantenimiento de la base de datos. El protocolo de comunicación mediante el cual se obtendrá la información desde los PLCs es el protocolo OPC.

El software Visual Basic 6.0. se escogió porque permite realizar las pantallas de interfase requeridas y permite comunicarse fácilmente con la base de datos y con los PLCs mediante un driver de comunicación. Además, la empresa posee la licencia de este programa, por lo que no es necesaria su adquisición, diminuyendo los costos de desarrollo.

La base de datos se desarrollará utilizando el programa de libre distribución PostgreSQL 8.2. Éste programa tiene características muy parecidas a otras bases de datos, y tiene la ventaja de ser gratuito.

La empresa también solicita que se desarrollen dos programas para el sistema SCADA. Uno se encargará de la sección de molienda y el otro de la sección de

limpieza. Los programas funcionarán en computadores diferentes y en cada uno habrá una base de datos donde se almacenará la información

1.3 SUSTENTACIÓN TEÓRICA

1.3.1 SISTEMAS SCADA

SCADA es un acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos, en español). Un sistema SCADA está formado por hardware de señal de entrada y salida, controladores, interfaz hombre – máquina, redes, comunicaciones, base de datos y software.[9]

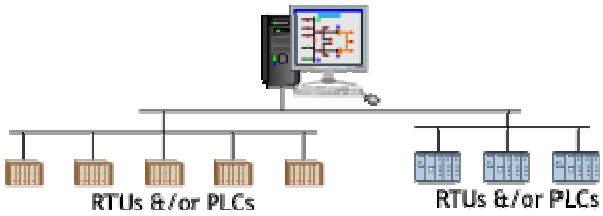


Figura 1.21. Esquema sistema SCADA.[16]

En un sistema SCADA existe una estación central con la capacidad de monitorear y controla un proceso. Pero, en realidad, la mayor parte del control es realizada automáticamente por una Unidad Terminal Remota (RTU, por sus siglas en inglés) o por un Controlador Lógico Programable (PLC, por sus siglas en inglés). Las funciones de control del servidor están casi siempre restringidas a reajuste básicos. Por ejemplo un PLC puede controlar el flujo de agua fría a través de un proceso, pero un sistema SCADA puede permitirle a un operador cambiar el punto de consigna (set point) de control para el flujo, y permitirá grabar y mostrar cualquier condición de alarma como la pérdida de flujo o temperatura elevada. La realimentación del lazo de control es cerrada a través del RTU o el PLC; el sistema SCADA monitorea el desempeño general de dicho lazo.[9]

Para el sistema SCADA se debe diseñar una aplicación de software para funcionar sobre ordenadores en el control de producción, proporcionando comunicación con los dispositivos de campo y controlando el proceso de forma automática desde la pantalla del ordenador. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios, tanto del mismo nivel como de otros supervisores dentro de la empresa: control de calidad, supervisión, mantenimiento, etc.[10]

1.3.1.1 Prestaciones del software de un sistema SCADA [10]

Un paquete SCADA debe estar en disposición de ofrecer las siguientes prestaciones:

- Posibilidad de crear paneles de alarma, que exigen la presencia del operador para reconocer una parada o situación de alarma, con registro de incidencias.
- Generación de históricos de señal de planta, que pueden ser volcados para su proceso sobre una hoja de cálculo.
- Ejecución de programas, que modifican la ley de control, o incluso anular o modificar las tareas asociadas al autómata, bajo ciertas condiciones.
- Posibilidad de programación numérica, que permite realizar cálculos aritméticos de elevada resolución sobre la CPU del ordenador.

Con ellas, se pueden desarrollar aplicaciones para ordenadores (tipo PC, por ejemplo), con captura de datos, análisis de señales, presentaciones en pantalla, envío de resultados a disco e impresora, etc.

1.3.1.2 Requisitos de un sistema SCADA[10]

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

- Deben ser sistemas de arquitectura abierta, capaces de crecer o adaptarse según las necesidades cambiantes de la empresa.
- Deben comunicarse con total facilidad y de forma transparente al usuario con el equipo de planta y con el resto de la empresa (redes locales y de gestión).
- Deben ser programas sencillos de instalar, sin excesivas exigencias de hardware, y fáciles de utilizar, con interfaces amigables con el usuario.

1.3.1.3 Componentes del sistema[9]

Los tres componentes de un sistema SCADA son:

1. Unidades de Terminal Remota
2. Estación Maestra y Computador con HMI.
3. Infraestructura de Comunicación

1.3.1.3.1 Unidad de Terminal Remota (RTU)

El RTU se conecta físicamente a los equipos y sensores y lee información de su estado. Un RTU, por ejemplo, posee la información de si una válvula esta abierta o cerrada, si un motor está funcionando o no, etc. También lee las señales de los sensores como medidas de: presión, flujo, voltaje, corriente, etc. Además, estos aparatos pueden enviar señales de control hacia los actuadores.

La RTU puede leer el estado de los datos digitales o medidas de datos analógicos y envía comandos digitales de salida o puntos de ajuste analógicos.

Las responsabilidades de operación entre RTUs, PLCs y DCS (Distributed Control System, o sistema de control distribuido en español) se han sustituido paulatinamente. Muchos fabricantes ofrecen RTUs con características similares a los PLCs y viceversa.[11] Gracias a las altas prestaciones de comunicación que poseen

los PLCs actuales, éstos pueden sustituir a RTUs en la implementación de sistemas SCADA.

1.3.1.3.2 Estación Maestra

El termino "Estación Maestra" se refiere a los servidores y el software responsable para comunicarse con el equipo del campo (RTUs, PLCs, etc). Entre estos se encuentra el software HMI corriendo en las estaciones de trabajo en el cuarto de control, o en cualquier otro lado. En un sistema SCADA pequeño, la estación maestra puede estar en un solo computador. En procesos de mayor tamaño, en los sistemas SCADA la estación maestra puede incluir muchos servidores y varias aplicaciones de software.

El sistema SCADA usualmente presenta la información al personal operativo de manera gráfica, en la forma de un diagrama de representación. Esto significa que el operador puede ver un esquema que representa la planta que está siendo controlada. Los diagramas de representación puede consistir en gráficos de líneas y símbolos esquemáticos para representar los elementos del proceso, o pueden consistir en fotografías digitales de los equipos sobre los cuales se animan las secuencias.

El paquete HMI para el sistema SCADA típicamente incluye un programa de dibujo con el cual los operadores o el personal de mantenimiento del sistema usan para cambiar la manera que estos puntos son representados en la interfase.

1.3.1.3.3 Infraestructura y Métodos de Comunicación[12]

Los sistemas SCADA necesitan comunicarse vía red, puertos GPIB (General-Purpose Instrumentation Bus), telefónica o satélite. Es necesario contar con computadoras remotas que realicen el envío de datos hacia una computadora central, esta a su vez será parte de un centro de control y gestión de información.

Para realizar el intercambio de datos entre los dispositivos de campo y la estación central de control y gestión, se requiere un medio de comunicación. Existen diversos medios que pueden ser: cableados (cable coaxial, fibra óptica, cable telefónico) o no cableados (microondas, ondas de radio, comunicación satelital).

Cada fabricante de equipos para sistemas SCADA emplea diferentes protocolos de comunicación y no existe un estándar para la estructura de los mensajes. Sin embargo, existen estándares internacionales que regulan el diseño de las interfaces de comunicación entre los equipos del sistema SCADA y equipos de transmisión de datos.

Un protocolo de comunicación es un conjunto de reglas y procedimientos que permite a las unidades remotas y central, el intercambio de información. Los sistemas SCADA hacen uso de los protocolos de las redes industriales.

La comunicación entre los dispositivos generalmente se realiza utilizando dos medios físicos: cable tendido, en la forma de fibra óptica o cable eléctrico, o radio. En cualquiera de los casos se requiere un MODEM, el cual modula y demodula la señal. Algunos sistemas grandes usan una combinación de radio y líneas telefónicas para su comunicación.

1.3.1.4 Funciones principales de un sistema SCADA[12]

1.3.1.4.1 Supervisión remota de instalaciones y equipos

Permite al operador conocer el estado de desempeño de las instalaciones y de los equipos alojados en la planta, lo que permite dirigir las tareas de mantenimiento y llevar estadísticas de fallas.

1.3.1.4.2 Control remoto de instalaciones y equipos

Mediante el sistema se pueden activar o desactivar los equipos remotamente (por ejemplo abrir válvulas, activar interruptores, prender motores, etc.), de manera automática y también manual. Además es posible ajustar parámetros, valores de referencia, algoritmos de control, etc.

1.3.1.4.3 Procesamiento de datos

El conjunto de datos adquiridos conforma la información que alimenta el sistema. Esta información es procesada, analizada, y comparada con datos anteriores, y con datos de otros puntos de referencia, dando como resultado una información confiable y veraz.

1.3.1.4.4 Visualización gráfica dinámica

El sistema es capaz de brindar imágenes en movimiento que representen el comportamiento del proceso, dándole al operador la impresión de estar presente dentro de una planta real. Estos gráficos también pueden corresponder a curvas de las señales analizadas en el tiempo.

1.3.1.4.5 Generación de reportes

El sistema permite generar informes con datos estadísticos del proceso en un tiempo determinado por el operador.

1.3.1.4.6 Representación se señales de alarma

A través de las señales de alarma se logra alertar al operador frente a una falla o la presencia de una condición perjudicial o fuera de lo aceptable. Estas señales pueden ser tanto visuales como sonoras.

1.3.1.4.7 Almacenamiento de información histórica

Se cuenta con la opción de almacenar los datos adquiridos, esta información puede analizarse posteriormente, el tiempo de almacenamiento dependerá del operador o del autor del programa.

1.3.1.4.8 Programación de eventos

Se refiere a la posibilidad de programar subprogramas que brinden automáticamente reportes, estadísticas, gráfica de curvas, activación de tareas automáticas, etc.

1.3.2 PROTOCOLO OPC

1.3.2.1 Introducción

OLE for Process Control (OPC) es un protocolo de comunicación industrial creado con la finalidad de tener un mecanismo estándar para obtener información de diversas fuentes, ya sean aparatos en la fábrica, o bases de datos en el cuarto de control.

OPC se desarrolló con la finalidad de permitir la comunicación entre dispositivos de control fabricados por diferentes empresas. Con este objetivo se creó la Fundación OPC, la cual está formada por los principales fabricantes de equipo de automatización industrial a nivel mundial.[13]

1.3.2.2 Objetivo de la creación del protocolo[15]

El protocolo OPC se creó con el objetivo de proveer un mecanismo estándar industrial para comunicar e intercambiar datos entre clientes y servidores utilizando la tecnología OLE de Microsoft.

La meta del grupo de trabajo de OPC es que el estándar sea ampliamente aceptado, adoptado e implementado por todos los fabricantes de servidores y clientes, como el mecanismo principal de intercambio de datos.

OPC fue creado con la intención de resolver los problemas del usuario final sobre interoperabilidad entre productos de diferentes vendedores. El usuario final es el que tiene el mayor beneficio de que los dispositivos de diferentes marcas trabajen bien juntos compartiendo datos.

1.3.2.3 Antecedentes para la creación de OPC

La arquitectura de la información dentro de los procesos industriales tiene tres niveles: nivel de campo, nivel de proceso y nivel administrativo. Con la finalidad de que cada nivel cumpla su función de manera eficiente, es necesario acceder a la información al nivel de la planta e integrarla con el sistema administrativo. Esto se logra utilizando una arquitectura de comunicación abierta y efectiva que se concentre en el acceso a los datos, y no en los tipos de datos. Es decir, se requiere un método común para que las aplicaciones adquieran datos desde cualquier fuente, como un dispositivo o una base de datos.

Hay muchas aplicaciones desarrolladas que acceden a los datos mediante drivers desarrollados independientemente. Esto conlleva los siguientes inconvenientes:

- Todos los desarrolladores de aplicaciones deben escribir su propio driver destinado para el hardware de cada proveedor, lo que provoca un esfuerzo duplicado.[15]
- No todas las características del hardware son soportadas por los todos los desarrolladores de drivers.[15]

- Cambios en las características del hardware pueden impedir el correcto funcionamiento del driver.[15]
- Generalmente dos paquetes no pueden acceder al mismo dispositivo simultáneamente ya que cada uno emplea un driver diferente.[15]

Los desarrolladores de hardware han intentando resolver estos problemas desarrollando los drivers, pero se ven obstaculizados por las diferencias de los protocolos de los clientes. Actualmente no han podido desarrollar un driver eficiente que pueda ser utilizado por todos los clientes. OPC conecta a los proveedores de hardware con los desarrolladores de software, proveyendo un mecanismo que permite obtener datos desde una fuente y comunicar la información a cualquier aplicación. De esta manera, un vendedor puede desarrollar servidores de alta optimización y reutilizables para comunicarse con la fuente de datos de manera eficiente. Si se provee al servidor con una interfase OPC se permite que cualquier cliente tenga acceso al dispositivo.

1.3.2.4 Características

La característica principal de éste protocolo es la conectividad abierta mediante especificaciones estándares abiertas.[13]

1.3.2.4.1 Conectividad abierta[13]

OPC posee conectividad abierta en automatización industrial y sistemas empresariales. La interoperabilidad se asegura a través de la creación y mantenimiento de especificaciones estándares abiertas. En la actualidad existen siete[13] especificaciones estándares terminados o en desarrollo.

La fundación OPC, basándose en estándares fundamentales y en la tecnología computacional, adapta o crea especificaciones para satisfacer las necesidades de la

industria. La fundación continuará creando nuevos estándares si es necesario y se adaptarán los existentes para poder utilizar nueva tecnología.

1.3.2.4.2 Especificaciones estándares[13]

El primer estándar (hoy llamado *Data Access Specification*) es el resultado de la colaboración de varios fabricantes de dispositivos de automatización trabajando en cooperación con Microsoft. El estándar, basado en las tecnologías de Microsoft OLE (Object Linking and Embedding) COM (Component Object Model) y DCOM (Distributed Component Object Model), define un conjunto estándar de objetos, interfases y métodos para ser usados en el control de procesos y en las aplicaciones de automatización industrial para facilitar la interoperatividad.

Para explicar la necesidad de la creación del estándar *Data Access Specification* se puede utilizar la siguiente analogía sobre los drivers de impresoras en los sistemas operativos DOS y Windows. En DOS, el desarrollador de cada aplicación tenía que escribir además el driver de impresora para cada impresora. En el mundo de la automatización industrial, una empresa que desarrolle un software de interfase hombre máquina (HMI), tiene que desarrollar además un driver para cada dispositivo industrial, incluyendo todas las marcas de PLCs.

Windows solucionó el problema de los drivers de impresoras incorporando el soporte de las impresoras dentro del sistema operativo. De esta manera un solo driver de impresora es utilizado por todas las aplicaciones, y estos drivers son desarrollados por los fabricantes de impresoras y no por los desarrolladores de aplicaciones. Windows también proporciono la solución para los drivers a nivel industrial. Agregar la especificación OPC a la tecnología OLE de Microsoft en Windows permitió la estandarización. Ahora, los fabricantes de dispositivos industriales pueden escribir los servidores de acceso de datos OPC y el software, como HMIs, pueden convertirse en clientes OPC.

Esto brinda al proveedor de software el beneficio de reducir sus gastos en desarrollo de herramientas de conectividad y enfocarse en las características principales del software. La ventaja para los usuarios es la flexibilidad. Ellos pueden seleccionar al proveedor de software basados en las características que ofrecen y no en el hecho de que posean el driver para la el modelo específico de impresora. De esta manera el usuario ya no tienen que correr con el costo que involucra la creación o actualización de la interfase debido a cambios en el sistema operativo o cambios del dispositivo.

En el campo industrial, a los usuarios se les asegura una mejor calidad de conectividad, ya que la especificación de acceso de datos OPC codificó los mecanismos de conexión y realizó pruebas de cumplimiento. Los productos de interfase OPC son construidos una vez y utilizados muchas veces, por lo que son sometidos a continuo control de calidad y mejoramiento.

Otros beneficios para los usuarios son: el tiempo de desarrollo del proyecto es menor y su costo es inferior gracias a la utilización de componente de software estandarizado; la fiabilidad técnica está garantizada ya que los estándares OPC se basan en estándares de la industria computacional.

Las especificaciones originales estandarizaron la adquisición de datos del proceso. Rápidamente se comprendió que la comunicación de otros tipos de datos también se beneficiaría de la estandarización, por lo que se introdujo estándares para alarmas y eventos (Alarms & Events), datos historicos (Historical Data) y Batch Data (información transferida a través de dispositivos portátiles), entre otros.

1.3.2.5 Especificaciones OPC actuales y en desarrollo[13]

1.3.2.5.1 OPC Data Access(Acceso de datos)

Este es el estándar original. Es utilizado para mover datos en tiempo real desde PLCs, DCSs, y otros dispositivos de control hacia HMIs y otros clientes de visualización.

1.3.2.5.2 OPC Alarms & Events (Alarmas y eventos)

Provee notificación de eventos y alarmas cuando son solicitados, al contrario del continuo flujo de datos de Data Access. Estos incluyen alarmas de proceso, acciones del operador, mensajes de información y mensajes de seguimiento y auditoria.

1.3.2.5.3 OPC Batch

Esta especificación se basa en la filosofía OPC sobre las necesidades especiales de procesos en los cuales se realiza la transferencia de datos a través de algún dispositivo portátil.. Provee interfase para el intercambio de capacidad de los equipos y condiciones actuales de operación.

1.3.2.5.4 OPC Data Exchange (intercambio de datos)

Esta especificación transforma los sistemas cliente - servidor en sistemas servidor – servidor con comunicación sobre redes Ethernet fieldbus. Esto provee interoperabilidad entre diferentes vendedores. Además, permite la configuración remota y servicios de diagnostico, monitoreo y administración.

1.3.2.5.5 OPC Historical Data Access (acceso de datos históricos)

OPC historical data access provee acceso a información almacenada. Se puede recuperar en una manera uniforme archivos históricos provenientes de simples sistemas de registro seriales hasta de complejos sistemas SCADA.

1.3.2.5.6 OPC Security (seguridad)

Todos los servidores OPC proveen información de alto valor para la compañía, que si no es actualizada apropiadamente, puede provocar consecuencias significativas al proceso de la planta. OPC Security especifica como controlar el acceso de clientes a estos servidores con la finalidad de proteger esta información y prevenir modificaciones no autorizadas a los parámetros del proceso.

1.3.2.5.7 OPC XML-DA

Provee reglas y formatos, consistentes y flexibles, para mostrar los datos de la planta utilizando XML (siglas de eXtensible Markup Language), complementando el trabajo realizado por Microsoft y otras compañías sobre SOAP (siglas de Simple Object Access Protocol) y servicios Web

1.3.2.5.8 OPC Complex Data (Datos complejos)

Es una especificación complementaria de Data Access y XML-DA, que permite a los servidores exponer y describir tipos de datos más complicados, como estructuras binarias y documentos XML.

1.3.2.5.9 OPC Commands (comandos)

Esta especificación está siendo desarrollada por un grupo de trabajo para crear un nuevo conjunto de interfases que permitan a los servidores y clientes OPC identificar, enviar y monitorear comandos de control que se ejecutan en un dispositivo.

1.3.2.5.10 OPC Unified Architecture (arquitectura unificada)

Es un nuevo conjunto de especificaciones que no se basan en la tecnología COM de Microsoft que proveerá estándares basados en la capacidad cross-platform, por lo que permitirá desarrollar aplicaciones que trabajen con sistemas operativos distintos de Microsoft Windows.

2 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

2.1 INTRODUCCIÓN

En el presente capítulo se describen el diseño y la implementación del sistema SCADA en Molino Electro Moderno S.A. Se inicia con un estudio de la planta de producción en la que va a funcionar el sistema. Se analiza el número motores, electro válvulas y demás actuadores que se controlan y el número de señales de sensores utilizadas.

Luego se describe como se realizan las comunicaciones, el manejo de la base de datos y el desarrollo de las aplicaciones utilizando Visual Basic 6.0. También se muestran las funciones del software, el método empleado para realizar las operaciones y las seguridades adoptadas para evitar encendidos o apagados accidentales.

En este capítulo se describe como se realiza el control de acceso y la administración de los usuarios. Se muestra como se crearon las ventanas utilizadas en las aplicaciones. Se describe la realización de las imágenes, la creación de las animaciones, la implementación de la barra de tareas y del menú, entre otras funciones del software.

En este capítulo también se encuentra información sobre: el detector de comunicación entre el PLC y la aplicación, el método utilizado para registrar los eventos y las alarmas, como se exporta la información a Microsoft Excel y sobre el cuadro de información utilizado para mostrar datos de la maquinaria de la empresa.

2.2 ESTUDIO DE LA PLANTA DE PRODUCCIÓN DE MOLINO ELECTRO MODERNO S.A.

Como se analizó en el capitulo uno, el proceso de producción de harina de trigo se puede dividir en dos etapas: limpieza de la materia prima y molienda de la misma. A su vez, la limpieza se puede subdividir en: prelimpieza, primera limpia y segunda limpia.

La maquinaria del molino es controlada mediante dos PLCs *Koyo D4-450*. Uno controla la etapa de limpieza y el otro la etapa de molienda. Estos PLCs monitorean el estado de sensores y activan y desactivan actuadores. Todas las entradas y salidas del PLC son digitales.

2.2.1 CONEXIÓN DE LOS PLC CON LOS COMPUTADORES

La empresa Molino Electro Moderno S.A. requiere que se desarrollen dos aplicaciones. La una se encargará del monitoreo y control de la etapa de limpieza y la otra cumplirá ésta función en la etapa de molienda. Cada aplicación correrá en un computador diferente que estará conectado con su respectivo PLC.

Figura 2.1. Conexión entre los computadores y los PLCs

En las siguientes secciones de este documento se enumeran los sensores y actuadores de cada etapa del proceso de molienda de trigo.

2.2.2 ACTUADORES DE LA ETAPA DE LIMPIEZA

En Tabla 2.1 se puede observar que el PLC de la sección limpieza controla 53 salidas. Estas salidas activan contactores y relés encargados de accionar 30 motores y 17 electro válvulas. Cuatro de los motores arrancan mediante el método estrella triángulo. El arranque del resto de los motores se lo realiza de forma directa.

Descripción del elemento	Cantidad	Número de salidas del PLC por cada elemento	Número total de salidas del PLC por tipo de elemento
Motor de rosca de trasporte de trigo (arranque directo)	8	1	8
Motor de rosca de trasporte de trigo (arranque estrella – triángulo)	1	3	3
Motor de elevador de trigo (arranque directo)	6	1	6
Motor de ventilador (arranque estrella – triángulo)	2	3	6
Motor de ventilador (arranque directo)	1	1	1
Motor de esclusa (arranque directo)	4	1	4
Motor de despuntadora (arranque estrella – triángulo)	1	3	3
Motor canal de aspiración (arranque directo)	1	1	1
Motores zaranda 1 (arranque directo)	2	1	1*
Motores zaranda 2 (arranque directo)	1	1	1
Motor bomba (arranque directo)	1	1	1
Motor de vibrador (arranque directo)	2	1	1*
Electro válvulas	17	1	17
NÚMERO TOTAL DE SALIDAS DEL PLC			53

* La zaranda 1 y el vibrador funcionan con dos motores que se activan desde una sola salida del PLC.

Tabla 2.1. Salidas conectadas al PLC de la sección limpieza

2.2.3 SENSORES DE LA ETAPA DE LIMPIEZA

El PLC lee el estado de los sensores y de los protectores térmicos para realizar el control de la maquinaria. La lectura de la mayoría de las señales se realiza a través de una matriz en la que las columnas se encuentran conectadas a un módulo de salidas del PLC y las filas a un módulo de entradas. Realizando un barrido de las columnas y leyendo el valor de las filas se determina el estado de las señales. A continuación se muestra el diagrama de conexión de la matriz.

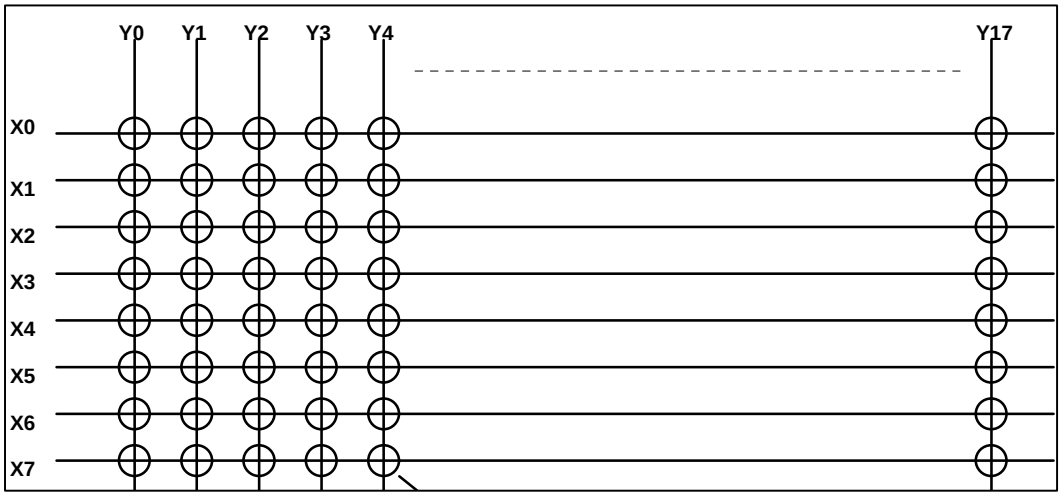


Figura 2.2. Diagrama de conexión de sensores mediante matriz

En la Figura de arriba (Figura 2.2) cada círculo representa a un sensor o a un protector térmico. Utilizando este método de conexión se pueden leer señales de hasta 128 elementos.

En la tabla que se muestra a continuación se encuentra un listado de los elementos conectados al PLC por medio de la matriz.

Descripción del elemento	Cantidad
Protector térmico	30
Detector de giro	1
Sensor de nivel máximo	3
Número total de elementos	34

Tabla 2.2. Entradas conectadas al PLC de la sección limpieza a través de la matriz.

Además de los enumerados en la tabla superior, en la etapa de limpieza se emplean otros sensores que se conectan directamente a un modulo de entradas del PLC. El listado de estos elementos se encuentra en la siguiente tabla.

Descripción del elemento	Cantidad
Sonda capacitiva	6
Selectores	4
Sensor compuerta abierta	7
Sensor compuerta cerrada	7
Detector de giro	3
Sensor de nivel máximo	1
Número total de elementos	28

Tabla 2.3. Entradas conectadas directamente al PLC de la sección limpieza

A continuación se muestra un esquema de la conexión del PLC de la sección limpieza:

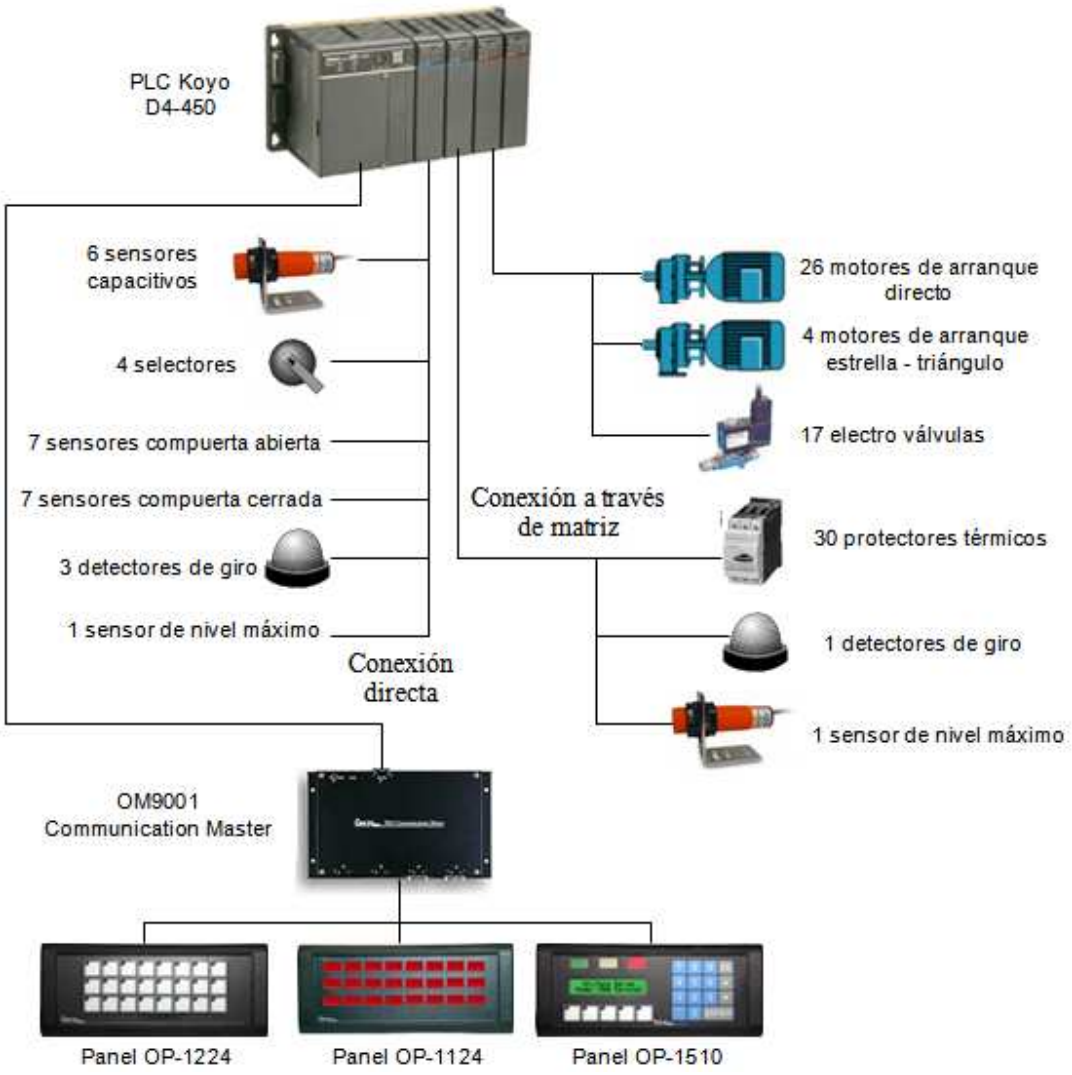


Figura 2.3. Esquema de conexión PLC limpieza

2.2.4 SENSORES Y ACTUADORES DE LA ETAPA DE MOLIENDA

A continuación se muestra una tabla con un listado de las salidas controladas por PLC de la etapa de molienda.

Descripción del elemento	Cantidad	Número de salidas del PLC por cada elemento	Número total de salidas del PLC por tipo de elemento
Motor esclusa (arranque directo)	11	1	11
Motor bomba (arranque directo)	1	1	1
Motor ventilador (arranque estrella – triángulo)	1	3	3
Motor disgregador (arranque directo)	4	1	4
Motor turbo tamiz (arranque directo)	1	1	1
Motor elevador de harina (arranque estrella – triángulo)	1	3	3
Motor cepilladora (arranque directo)	3	1	3
Motor banco de molienda (arranque directo)	3	1	3
Motor banco de molienda (arranque estrella – triángulo)	13	3	39
Motor rosca de trasporte de harina (arranque directo)	1	1	1
Motor sasor (arranque directo)	2	1	2
Motor cernidera plana (arranque directo)	2	1	2
Motor transflowtron (arranque directo)	1	1	1
Electro válvula	1	1	1
Vibrador	1	1	1
Motor dosificador (arranque directo)	1	1	1
Contactor alimentación arrancador suave	1	1	1
Foco señalización	2	1	2
Motor filtro de deposito de harina (arranque directo)	1	1	1
Motor desatador (arranque directo)	6	1	6
Contactor marcha arrancador suave	1	1	1
Motor bomba (arranque estrella – triángulo)	2	3	6
Válvula MAUB	1	1	1
Martilo	1	1	1
NÚMERO TOTAL DE SALIDAS DEL PLC			96

Tabla 2.4. Salidas conectadas al PLC de la sección molienda

El PLC de la sección limpieza controla 96 salidas. Estas salidas activan contactores y relés encargados de accionar 54 motores, una electro válvula, un vibrador, dos

señales para el control de un arrancador suave, dos focos de señalización, una válvula MAUB y un martillo. Diecisiete de los motores arrancan mediante el método estrella triángulo. El arranque del resto de los motores se lo realiza de forma directa con excepción de un motor de elevada potencia que se enciende utilizando un arrancador suave.

2.2.5 SENSORES DE LA ETAPA DE MOLIENDA

Para controlar la maquinaria el PLC lee el estado de los sensores y de los protectores térmicos. Al igual que en la etapa de limpieza la mayoría de las señales se leen a través de una matriz. En la tablas que se muestran a continuación se enlistan los elementos conectados al PLC por medio de la matriz y los elementos que se conectan directamente a un modulo de entradas.

Descripción del elemento	Cantidad
Protector térmico	54
Detector de giro	4
Micro switch	1
Detector de presencia de trigo	2
Sensor capacitivo	2
Alarmas en bancos	4
Número total de elementos	67

Tabla 2.5. Entradas conectadas al PLC de la sección molienda a través de la matriz.

Descripción del elemento	Cantidad
Selectores	3
Sensor válvula abierta	1
Sensor válvula cerrada	1
Señal alarma arrancador suave	1
Sensor de nivel máximo	2
Número total de elementos	8

Tabla 2.6. Entradas conectadas directamente al PLC de la sección molienda

A continuación se muestra un esquema de la conexión del PLC de la sección molienda:

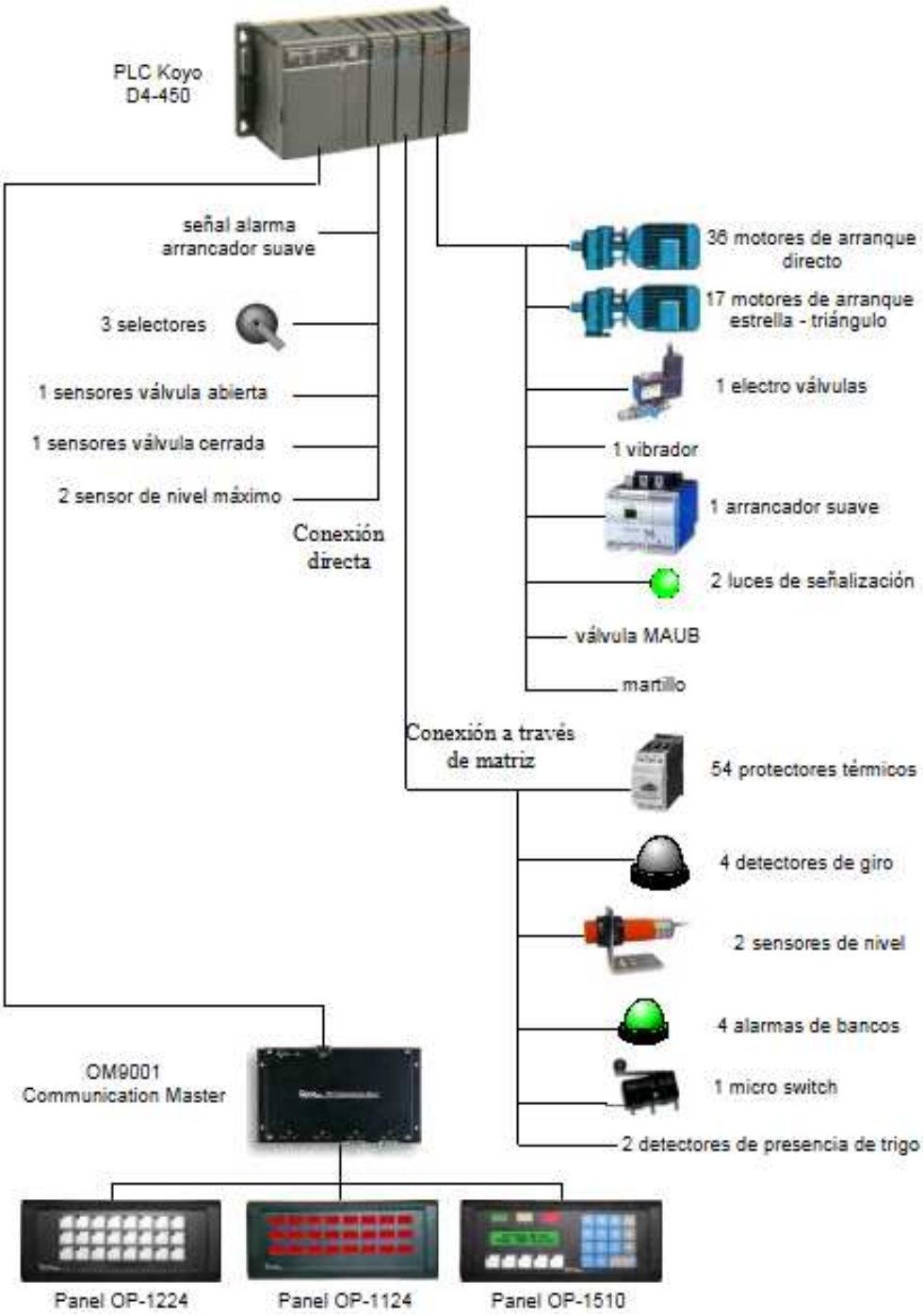


Figura 2.4. Esquema de conexión PLC molienda

2.3 DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En esta sección se realiza el diseño y se describe la implementación del sistema SCADA. Se describe como se implementaron las comunicaciones y las funciones del software. Se describen la lectura y la escritura de información en la base de datos, las seguridades implementadas para evitar operaciones no deseadas y el método empleado para encender y apagar la maquinaria. También se describe la implementación de animaciones e indicadores gráficos del estado de la maquinaria.

2.3.1 DISEÑO DE LA BASE DE DATOS

Los registros de alarmas y eventos ocurridos se almacenan en una base de datos. También se almacena en la base de datos información como: los códigos y las descripciones de la maquinaria, la información de los usuarios, y el tipo de acceso de cada usuario. Para poder utilizar la base de datos primero se debe realizar su diseño y luego implementarla utilizando algunos códigos SQL.

En la presente sección se resumen los pasos sugeridos en el libro “Beginning with PostgreSQL: From Novice to Professional”[14] para diseñar una base de datos. Luego se siguen estos pasos para realizar el diseño que se utilizará en el sistema SCADA.

2.3.1.1 Pasos para diseñar una base de datos

El diseño de una base de datos se puede dividir en los siguientes pasos:

2.3.1.1.1 Reunión de información

La primera etapa en el diseño de una base de datos consiste en reunir información sobre la función que ésta debe cumplir. Es decir, conocer para qué se está realizando el diseño. Se debe tener un objetivo claro antes iniciar la elaboración de la base de datos.

2.3.1.1.2 Desarrollo de un diseño lógico

La primera etapa real de diseño consiste en desarrollar un diseño lógico. Esto requiere de varias etapas. El diseño lógico se concentra en la creación de una estructura lógica, mas no en detalles específicos de implementación.

2.3.1.1.2.1 Determinar entidades

Una vez que se ha reunido información sobre los objetivos iniciales y los requerimientos de la empresa, se está en posición de identificar las entidades principales. Es decir, lo objetos claves que deben aparecer en la base de datos. Por ejemplo, usuarios, elementos de la fábrica, etc. Luego de identificar los componentes principales, se deben identificar los atributos de éstos. Por ejemplo, para la entidad usuarios, los atributos podría ser: nombre, apellido y clave. En este momento del diseño es importante revisar la información reunida inicialmente y constatar que no se esté dejando de lado nada importante.

2.3.1.1.2.2 Convertir las entidades en tablas

El siguiente paso es más técnico, y consiste en convertir las listas de los componentes y atributos a un formato que se parezca más a una base de datos. Lo primero que se debe hacer es seleccionar nombres significativos para las tablas. Se recomienda nombrarlas en modo singular y utilizar palabras de menos de 250 caracteres. A continuación, se deben convertir los atributos en palabras con mayor significado.

2.3.1.1.2.3 Determinación de las relaciones y de la cardinalidad

Este es un paso importante dentro del diseño de la base de datos. Consiste en eliminar aquellos atributos que ocurren varias veces en la misma entidad y decidir como se deben relacionar las entidades entre sí. Este proceso se suele nombrar cardinalidad.

2.3.1.1.2.4 Dibujo de diagramas de relaciones

En las bases de datos, una representación gráfica de la estructura de los datos puede ser de mucha ayuda para entender el diseño. En esta etapa de diseño, la estructura elaborada es conocida como modelo conceptual. Todavía no se han definido los detalles de la implementación, pero se ha desarrollado la estructura lógica de los datos. En un modelo conceptual de datos, las tablas se pueden representar como cuadros, y las relaciones entre las tablas se muestran utilizando líneas. Estas líneas poseen símbolos en su extremo que indican el tipo de relación o cardinalidad. La relación entre las tablas siempre es bi-direccional. Por lo tanto, siempre debe haber un símbolo en ambos extremos de las líneas. Los símbolos que se utilizarán en este documento se muestran a continuación:

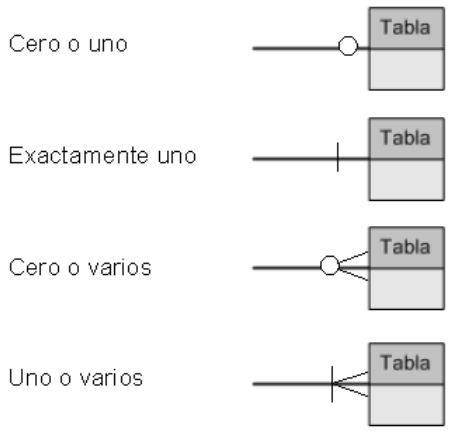


Figura 2.5. Símbolos de cardinalidad

El siguiente ejemplo ilustra como se realiza la lectura de estos gráficos:

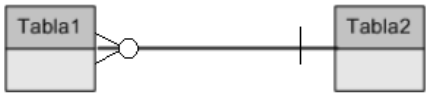


Figura 2.6. Relación simple entre dos tablas

El diagrama indica que las tablas poseen las siguientes relaciones:

- Por cada fila de la tabla 1, debe haber exactamente una fila en la tabla 2.
- Por cada fila de la tabla 2, puede haber cero, una o varias filas en la tabla 1.

2.3.1.1.2.5 Validación del diseño conceptual

En este momento, es muy importante detenerse y analizar la validez del diseño conceptual. Un error que se deje pasar será mucho más difícil corregir en etapas posteriores. También es importante, de ser posible, dialogar con los futuros usuarios del sistema, mostrarles el diagrama, explicar lo que significa, y verificar que el diseño esté acorde con sus expectativas del sistema.

2.3.1.1.3 Conversión a un modelo físico

Una vez que se tiene un modelo lógico de los datos, el cual se ha verificado que es correcto, se pude empezar a realizar la representación física del diseño. Esta etapa también se compone de varias etapas.

2.3.1.1.3.1 Establecer claves primarias

Usualmente, el primer paso consiste en decidir cual será la clave primaria de cada tabla. Se debe trabajar con una tabla a la vez, considerándolas individualmente, y decidir cual de los atributos en cada fila convierte a esa fila en única. También se puede dar el caso que no exista un atributo con estas características. Se debe generar una lista con claves candidatas y luego seleccionar una que será la clave primaria. Si no se pueden encontrar claves candidatas, se debe recurrir a una clave primaria lógica, que es un atributo creado específicamente para ser una clave primaria.

El proceso consiste en buscar una sola columna que será única, y luego buscar combinaciones de columnas que sean únicas. Además, se debe tener en cuenta que

los valores de las columnas candidatas no deben ser nunca nulos. No tiene sentido tener una clave primaria cuyo valor podría ser desconocido.

Cuando se buscan columnas para usarlas como clave primaria, se debe considerar que entre más corta sea la longitud del campo, la búsqueda de valores específicos será más eficiente.

Por lo general, es más común buscar información en las tablas utilizando la clave primaria que cualquier otra columna, por lo que es importante que esta búsqueda sea eficiente. Un elemento es más rápido de encontrar cuando se busca dentro de una lista de elementos de pocos caracteres. Por lo tanto, es importante seleccionar como clave primaria una columna cuyos datos sean de corta longitud. Lo ideal son valores de tipo *integer,* auque también se pueden utilizar variables de tipo *string* que sean de pocos caracteres.

2.3.1.1.3.2 Establecer las claves foráneas

Después de establecer las claves primarias se debe trabajar en el mecanismo usado para relacionar las tablas. El modelo conceptual muestra el modo en que las tablas se relacionan entre sí y el identificador único de cada fila de la tabla. Para establecer las claves foráneas, por lo general, solo se debe asegurar que la columna de una tabla, definida como clave primaria, también aparezca en todas las otras tablas que están directamente relacionadas con aquella tabla.

En este momento se puede modificar el diagrama conceptual incluyendo como se relacionan físicamente las tablas. Además, en este momento, en el diagrama, las claves primarias aparecen subrayadas para ser identificadas.

2.3.1.1.3.3 Establecer tipos de datos

Cuando ya se tienen las tablas, columnas y relaciones, se debe trabajar con cada tabla, una por una, agregando los tipos de datos a cada columna. En esta etapa, también se necesita identificar las columnas que deberán aceptar valores nulos (null), y se deben declarar el resto de las columnas como *not null,* es decir, que no aceptan valores de este tipo.

2.3.1.2 Diseño de la base de datos

Siguiendo los pasos descritos en la sección anterior, a continuación se realiza el diseño de la base de datos.

2.3.1.2.1 Reunión de información

Para cumplir con los objetivos planteados para la implementación del sistema SCADA, la base de datos debe almacenar la siguiente información: los registros de los eventos y de las alarmas ocurridas, la información de los usuarios, la información de los tipos de accesos y la información de la maquinaria.

2.3.1.2.2 Desarrollo de un diseño lógico

A continuación se describe el desarrollo del diseño lógico de la base de datos utilizada en el sistema SCADA de Molino Electro Moderno S.A.

2.3.1.2.2.1 Determinar entidades

En la tabla mostrada a continuación se encuentran las entidades y los atributos que debe tener la base datos. También se muestra una descripción de los atributos.

Entidad	Atributos	Descripción del atributo
Usuarios del sistema	Nombre	Nombre del usuario
	Apellido	Apellido del usuario
	Nombre de usuario	Identificación utilizada para ingresar al sistema
	Clave	Contraseña del usuario
	Tipo de acceso	Define las restricciones de la persona que ingresa al sistema
Accesos	Identificación del tipo de acceso	Código que identifica el tipo de acceso
	Descripción del tipo de acceso	
Elemento PLC	Identificación de la entrada/salida	Código para diferenciar un sensor de otro
	Código de la entrada/salida	Ej. PS1002
	Tipo de sensor/salida	Ej. presóstato
	Descripción	Ej. trasporte harina
	Análogo o digital	
	Señalización	Relacionado a la señal de falla del sensor
	Número en Visual Basic	Código empleado en Visual Basic para identificar el elemento
	Sección	Molienda o limpieza
Eventos	Identificación del evento	
	Elemento involucrado	Identificación del elemento
	Hora de inicio	Ej. hora a la que se activa el motor
	Causa de inicio	Encendido manual o automatico
	Hora de finalización	Ej. hora a la que se apaga el motor
	Causa de finalización	Parada de emergencia o apagado manual
	Intervalo	Tiempo durante el cual el motor permaneció encendido
Alarmas	Identificación de la alarma	
	Sensor involucrado	Identificación del elemento
	Hora	Hora en la que sucedió la alarma
	Causa	Que provocó la alarma
	Usuario	Operador encargado al momento de la alarma
	Consecuencia	Si la alarma provocó parada parcial o total

Tabla 2.7. Determinación de entidades

2.3.1.2.2.2 Convertir las entidades en tablas

En la siguiente tabla se muestran las entidades convertidas en tablas. En la tabla se muestran los nombres que se utilizarán para identificar a las tablas y a los atributos de la base de datos.

Tabla	**Atributos**
Usuarios	Nombre Apellido Usuario_id Clave Nombre_acceso
Accesos	Nombre_acceso Descripción
Elementos_plc	Elemento_id Elemento_co Tipo Descripcion Ana_o_digi Señalización Numero_vb Sección
Eventos	Evento_id Elemento_id Hora_inicio Hora_fin Intevalo Causa_inicio Causa_fin
Alarmas	Alarma_id Elemento_id Hora Causa Usuario_id Consecuencia

Tabla 2.8. Conversión de entidades en tablas

2.3.1.2.2.3 Determinación de las relaciones y de la cardinalidad

A continuación se describen las relaciones que deben existir entre las tablas de la base de datos.

Relación entre las tablas eventos y elementos_plc

Por cada evento debe haber un elemento del PLC, y por cada elemento puede haber ninguna o varios eventos.

Relación entre las tablas usuarios y acceso_id

Por cada usuario existe un acceso, y por cada acceso existe cero o varios usuarios.

Relación entre las tablas alarmas y elementos_plc

Por cada alarma debe haber un elemento del PLC, y por cada elemento puede haber ninguna o varios alarmas.

Relación entre las tablas alarmas y usuarios

Por cada alarma debe haber un usuario, y por cada usuario puede haber ninguna o varios alarmas.

2.3.1.2.2.4 Dibujo de diagramas de relaciones

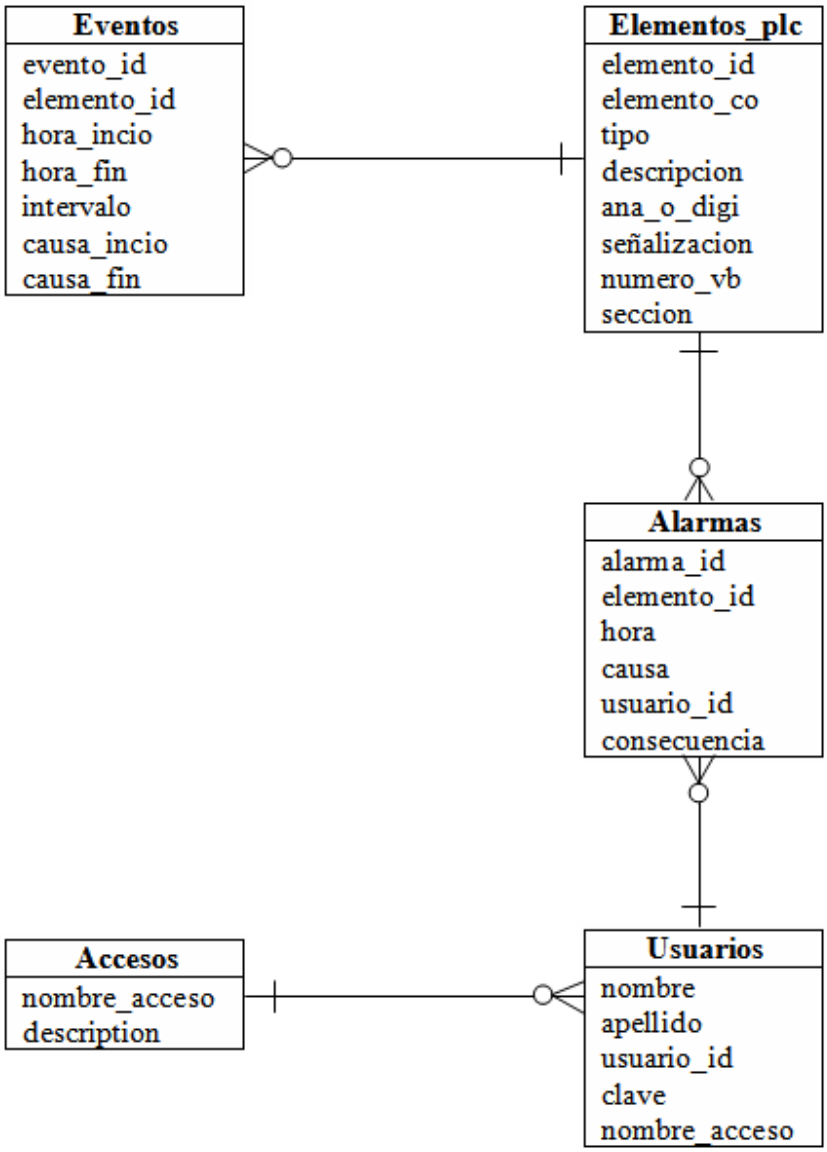


Figura 2.7. Diagrama de relaciones de la base de datos

2.3.1.2.3 Conversión a un modelo físico

A continuación se muestra la conversión del modelo lógico a un modelo físico

2.3.1.2.3.1 Establecer claves primarias

Las claves primarias para cada tabla son las siguientes:

Tabla	Clave primaria
Usuarios	**Usuario_id**
Accesos	**Nombre_acceso**
Elemento_plc	**Elemento_id**
Eventos	**Evento_id**
Alarmas	**Alarma_id**

Tabla 2.9. Claves primarias de las tablas

2.3.1.2.3.2 Establecer las claves foráneas

Las tablas *Eventos* y *Elementos_plc* se relacionan mediante la clave *Elemento_id*; la clave *Nombre_acceso* relaciona las tablas *Usuarios* y *Accesos*. *Elemento_id* también relaciona las tablas *Alarmas* y *Elementos_plc*. Para relacionar las tablas *Alarmas* y *Usuarios* se emplea la clave *Usuario_id*.

2.3.1.2.3.3 Establecer tipos de datos

En las siguientes tablas se muestran los tipos de datos y las restricciones seleccionados para cada columna:

Nombre de la columna	Tipo de datos	Restricción (Constraint)
Usuarios		
Nombre	CHARACTER VARYING(16)	NOT NULL
Apellido	CHARACTER VARYING(16)	NOT NULL
Usuario_id	CHARACTER VARYING(32)	PRIMARY KEY
Clave	CHARACTER VARYING(32)	NOT NULL
Nombre_acceso	INTEGER	REFERENCE a accesos.nombre_acceso
Accesos		
Nombre_acceso	CHARACTER VARYING(20)	PRIMARY KEY
Descripción	CHARACTER VARYING(100)	
Elementos_plc		
Elemento_id	INTEGER	PRIMARY KEY
Elemento_co	CHARACTER VARYING(10)	UNIQUE
Tipo	CHARACTER VARYING(32)	
Descripcion	CHARACTER VARYING(32)	
Ana_o_digi	CHARACTER VARYING(10)	
Señalización	CHARACTER VARYING(32)	
Numero_vb	INTEGER	UNIQUE
Sección	CHARACTER VARYING(18)	
Eventos		
Evento_id	INTEGER	PRIMARY KEY
Elemento_id	INTEGER	REFERENCE a salidas_plc.elemento_id
Hora_inicio	TIMESTAMP	NOT NULL
Hora_fin	TIMESTAMP	
Intevalo	INTERVAL	
Causa_incio	CHARACTER VARYING(20)	
Causa_fin	CHARACTER VARYING(20)	
Alarmas		
Alarma_id	Serial	
Elemento_id	Integer	
Hora	Timestamp	NOT NULL
Causa	CHARACTER VARYING(400)	
Usuario_id	CHARACTER VARYING(32)	
Consecuencia	CHARACTER VARYING(25)	

Tabla 2.10. Tipos de datos y restricciones de las columnas de la base de datos.

2.3.1.3 Creación de la base de datos

Para crear las tablas de la base de datos se utiliza el comando SQL *create.* Este comando va acompañado del nombre de la tabla, de los nombres de sus columnas y de los tipos de datos y de las restricciones de cada columna. Otro argumento de este comando es la clave primaria y las claves foráneas de cada tabla. A continuación se muestra el comando utilizado para la creación de la tabla alarmas:

```
create table Alarmas
(
        Alarma_id           SERIAL                          ,
        Elemento_id         INTEGER                         ,
        Hora                TIMESTAMP       not null,
        Causa               CHARACTER VARYING(400)          ,
        usuario_id          CHARACTER VARYING(32)           ,
        consecuencia        CHARACTER VARYING(25)           ,
        CONSTRAINT alarma_id_pk PRIMARY KEY(alarma_id),
        CONSTRAINT alarmas_alarma_id_fk FOREIGN KEY (elemento_id) REFERENCES elementos_plc (elemento_id),
        CONSTRAINT alarmas_usuario_id_fk FOREIGN KEY(usuario_id) REFERENCES usuarios(usuario_id)
);
```

Para crear las otras tablas se utilizan comandos similares al que se encuentra sobre este texto. En estos comandos se incluyen: los nombres de las tablas, los nombres de las columnas, los tipos de datos, las restricciones de cada columna, las claves primarias y las claves foráneas. Al ejecutar estos comandos desde el Command Promt se crean las tablas descritas en esta sección. El resto de los comandos se pueden observar en el Anexo A.

Los valores iniciales de las tablas usuarios, accesos y elementos_plc se llenaron manualmente desde el Comand Promp utilizando comandos SQL. El resto de las tablas se llenan automáticamente cuando corren las aplicaciones elaboradas en Visual Basic. Cuando se crea un nuevo usuario la información de éste se escribe en la base de datos desde las aplicaciones.

En la siguiente sección se explica como se comunican las aplicaciones con la base de datos elaborada en PostgreSQL para ingresar, y leer información. A continuación se muestra uno de los comandos SQL empleados para llenar la información de la tabla usuarios:

```
insert into usuarios(nombre, apellido, usuario_id, clave, nombre_acceso) values('Isaac','Pastor','isapacos','548300','administrador');
```

Este comando ingresa los datos: Isaac, Pastor, isapacos, 548300 y administrador en las columnas: nombre, apellido, usuario_id, clave y nombre_acceso, respectivamente.

Luego de crear las tablas que conforman las base de datos se pueden ingresar, borrar y leer datos desde las aplicaciones creadas en Visual Basic utilizando comandos SQL. En secciones posteriores de este documento se muestran ejemplos de como se realizan estas operaciones.

2.3.2 COMUNICACIÓN ENTRE VISUAL BASIC Y POSTGRESQL

Uno de los objetivos de la implementación del sistema SCADA es tener un registro de los eventos y de las alarmas que ocurran en la planta de producción. Estos registros se almacenarán en la base de datos de PostgreSQL.

Es necesario ingresar y leer información de la base de datos desde la aplicación desarrollada en Visual Basic. De esta manera, cuando la aplicación detecta la

activación de una alarma o el encendido o apagado de un actuador, se procede a registrarlo. Además se puede leer la información almacenada en PostgreSQL y mostrarla en la aplicación para que el operador pueda obtener datos históricos de funcionamiento.

Para poder escribir o leer información desde la aplicación se debe establecer un método de comunicación. A continuación se describe el método utilizado y la manera de configurarlo.

2.3.2.1 Implementación de la comunicación entre la aplicación y la base de datos

La comunicación entre la base de datos y el software del sistema SCADA se realiza mediante un ODBC (Open Data Base Conectivity o conectividad abierta de bases de datos). Este es un elemento que permite establecer la comunicación entre la aplicación desarrollada en Visual Basic y la base de datos de PostgresSQL. Para configurar al ODBC driver se deben seguir los siguientes pasos:

a. Abrir la ventana de las herramientas administrativas que se encuentra en el panel de control de Windows.

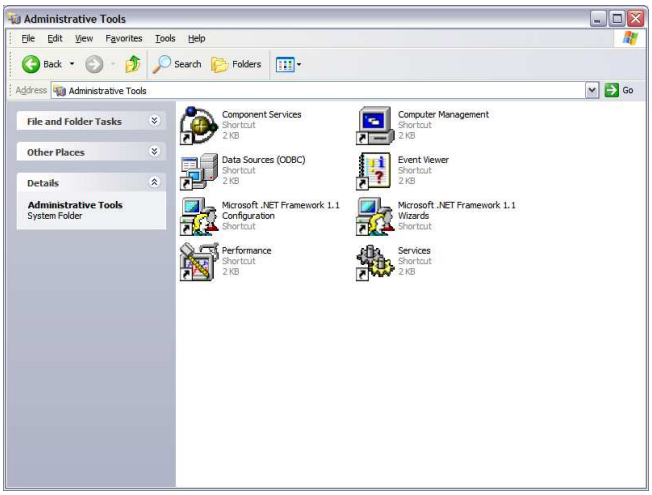


Figura 2.8. Ventana Herramientas Administrativas de Windows

b. Abrir la ventana *Data Sources (ODBC) y h*acer clic sobre el botón *add* que aparece al hacer clic en la pestaña *Sistema DNS.*

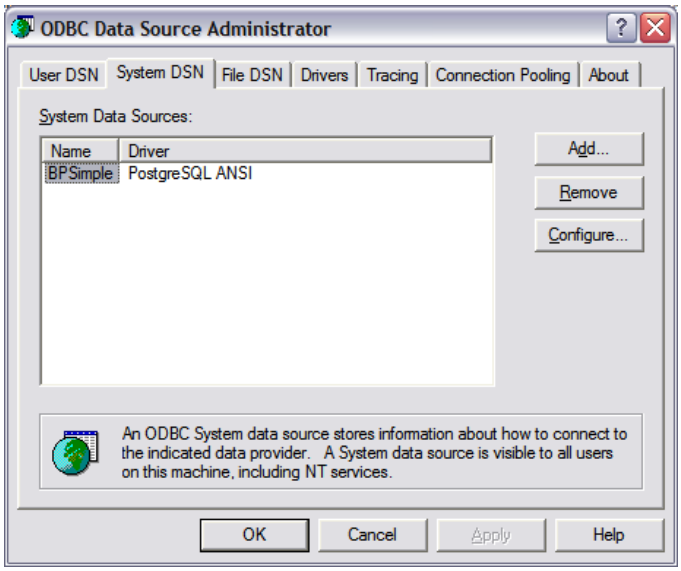


Figura 2.9. Pestaña System DNS de la ventana ODBC Data Source Administrator

c. En la ventana que aparece seleccionar la opción PostgreSQL ANSI

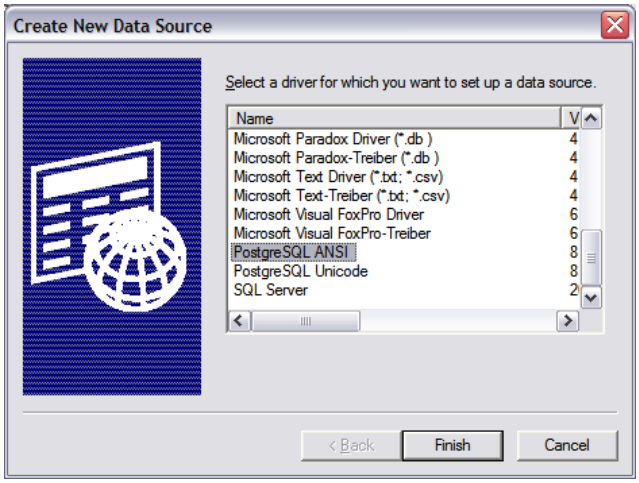


Figura 2.10. Create New Data Source

d. En la siguiente ventana que aparece se debe llenar la información sobre la base de datos con la cual nos queremos comunicar. En el caso del sistema SCADA de Molino Electro Moderno S.A. estos datos son:

Database: molino
Server: localhost
User Name: Isaac

Los datos de SSL Mode y Port se mantienen los preestablecidos, estos son:

SSL Mode: disable
Port: 5432

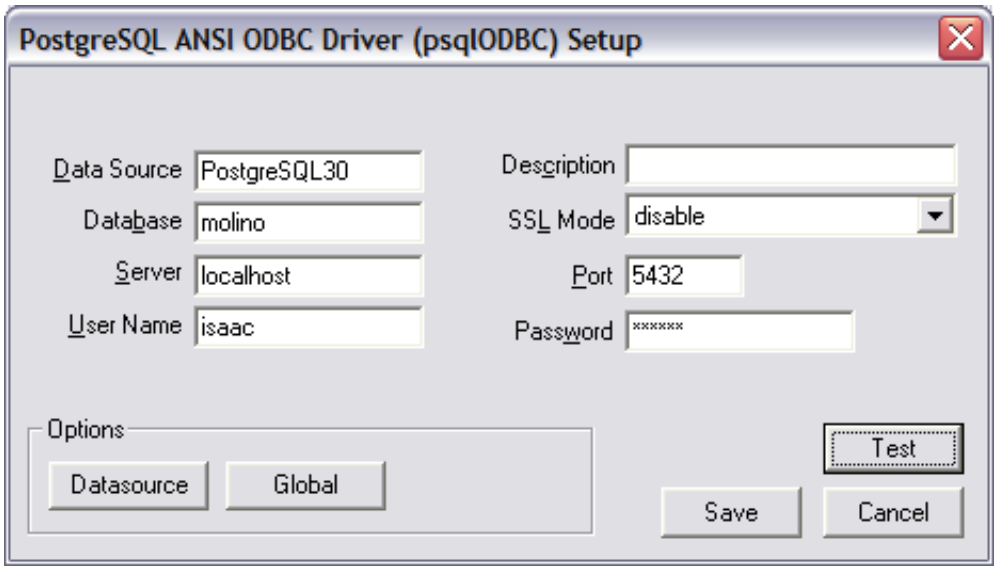


Figura 2.11. PostgreSQL ANSI ODBC Driver (psqlODBC) Setup

e. Para determinar que la configuración fue exitosa se debe realizar una prueba pulsando sobre el botón *Test.* Si se logró establecer la comunicación con la base de datos aparece el siguiente mensaje:

Figura 2.12. Ventana mostrando en mensaje que indica que la conexión con la base de datos fue exitosa.

Para establecer la comunicación entre Visual Basic y la base de datos se deben realizar unas declaraciones y agregar código para realizar las preguntas y leer la información proveniente de PostgreSQL.

Las declaraciones que se deben realizar son las siguientes:

```
Dim SqlQry As String
Dim MyCon As ADODB.Connection
Dim MyRecSet As ADODB.Recordset
```

Estas variables cumplen las siguientes funciones:

- SqlQry: Almacena el comando SQL
- MyCon: Se utiliza para realizar la conexión con la base de datos. En ella se guarda la información requerida y permite abrir y cerrar el canal de comunicación.
- MyRecSet: En esta variable se almacena la información obtenida de la base de datos.

El código utilizado para obtener información de la base de datos es el siguiente:

```
Set MyCon = New ADODB.Connection
```

```
MyCon.ConnectionString = "Provider=MSDASQL.1;Persist Security Info=False;User ID=postgres;Data Source=PostgreSQL30"
Set MyRecSet = New ADODB.Recordset
MyCon.Open

With MyRecSet
    .CursorLocation = adUseClient
    .CursorType = adOpenDynamic
    .LockType = adLockOptimistic
    .ActiveConnection = MyCon
    .Open SqlQry
End With
```

Para leer la información de la variable MyRecSet se utiliza el siguiente código:

```
nombre = MyRecSet.Fields(0)
Descripcion = MyRecSet.Fields(1)
```

En este ejemplo, el valor de la columna cero de la MyRecSet se coloca en la variable nombre y el valor de la columna uno en la variable Descripcion.

Para definir si la base de datos devolvió valores a partir del comando SQL utilizado se emplea el código mostrado a continuación:

```
If MyRecSet.EOF = False Then
```

Cuando se cumple ésta condición, es decir, cuando el valor de MyRecSet.EOF es falso, implica que la base de datos si envió valores.

Una vez que se han leído los datos enviados desde la base de datos, se debe cerrar la variable MyRecSet para poder utilizarla nuevamente. El código que realiza esta operación es el siguiente:

```
MyRecSet.Close
```

2.3.3 MODOS DE OPERACIÓN DEL SOFTWARE DEL SISTEMA SCADA

Es posible que accidentalmente se pulse algún botón del teclado o del ratón del computador en el que corre el software del sistema SCADA. Lo cual podría causar acciones no deseadas. Un ejemplo de esto es el apagado accidental de un motor, lo cual podría provocar una parada total de la producción.

El software del sistema SCADA puede funcionar en dos modos de operación: supervisión y control. Si el usuario selecciona el modo de supervisión se deshabilitan algunas funciones del software. De esta manera se pretende evitar el encendido y el apagado accidental de motores u otros actuadores.

A continuación se describen las características de cada modo de operación:

2.3.3.1 Modo de supervisión

Como su nombre lo indica este modo solamente permite supervisar, mas no controlar. Cuando se ingresa a este modo de operación se puede visualizar el estado de los motores, electro válvulas, sensores y demás equipos, pero no se puede modificar su estado.

2.3.3.2 Modo de control

El modo de control se utiliza para activar y desactivar los elementos de manera individual. Cuando se ingresa a este modo de operación se puede cambiar el estado

de cualquier actuador del proceso. En este modo también se puede realizar el arranque automático de la cada etapa.

2.3.4 DEFINICIÓN DEL NÚMERO DE VENTANAS QUE SE UTILIZARÁN Y LA FUNCIÓN DE CADA UNA

El software del sistema SCADA va a permitir a los operadores realizar varias funciones. Desde la aplicación se puede realizar el encendido y apagado de actuadores, visualizar información de la maquinaria, revisar datos históricos de funcionamiento, realizar paradas de emergencia, agregar y eliminar usuarios, etc.

Para que el operador pueda realizar estas operaciones fácilmente el software debe ser amigable. Esto quiere decir que debe ser fácil de operar, de agradable apariencia y que permita la rápida visualización de la información requerida.

Para cumplir con éstas características es necesario diseñar y elaborar las siguientes ventanas:

- Ventana de administración del programa,
- Ventana para el ingreso de clave,
- Ventana para agregar usuarios,
- Ventana para eliminar usuarios,
- Ventana para cambio de contraseña,
- Ventanas del proceso,
- Ventana para ingresar el motivo que ocasionó una alarma, y
- Ventanas para mostrar registros de la base de datos.

A continuación se describe la función de cada ventana:

2.3.4.1 Ventana de administración del programa

Esta ventana será la primera que se muestre al hacer correr el programa. Mediante esta se podrá iniciar una sesión y luego acceder a las otras ventanas. Iniciar una sesión implica ingresar al sistema utilizando un nombre de usuario y una clave registrada. Si los datos ingresados son correctos el usuario obtiene acceso a las funciones del software. Esta ventana también permitirá salir del programa o cerrar la sesión para que otro usuario ingrese.

2.3.4.2 Ventana para el ingreso de clave

Para poder iniciar una sesión, el operador debe poseer un nombre de usuario y una clave válidos. Ésta ventana permitirá el ingreso de esta información y la comprobación de su validez.

2.3.4.3 Ventana para agregar usuarios

Esta ventana se utilizará para permitir que nuevos usuarios accedan al sistema. En ella se colocarán casilleros para escribir información importante de la persona, como: nombre, apellido, nombre de usuario, clave y tipo de acceso que posee. Esta información se guardará en la base de datos para permitir el ingreso al sistema de dicha persona y conocer quien estuvo operando el software. No todos los usuarios podrán utilizar todas las funciones del sistema. El tipo de acceso indica los recursos que un usuario puede utilizar y cuales le están restringidos. Por ejemplo, algunos usuarios podrán agregar y borrar usuarios mientras otros no.

2.3.4.4 Ventana para eliminar usuarios

Esta ventana se utilizará para eliminar usuarios a los cuales ya no se les quiere permitir el acceso al sistema por diversas razones. En ella se mostrará un listado de los usuarios actuales para seleccionar al que se desea borrar y botones para aceptar o cancelar esta operación.

2.3.4.5 Ventana para cambio de contraseña

Por motivos de seguridad, los usuarios pueden cambiar su contraseña cuando deseen. Para realizar esta operación se deberá ingresar la contraseña actual, la contraseña nueva y confirmar la nueva contraseña. Estas operaciones y la comprobación de la validez de las claves se realizarán en esta ventana.

2.3.4.6 Ventanas del proceso

Con el objetivo de mostrar la situación de los actuadores y sensores involucrados en el proceso se desarrollaran cuatro ventanas. Éstas son:

- Ventana para la etapa de recepción, prelimpieza, almacenaje y trasilaje
- Ventana para la etapa de limpieza
- Ventana para la etapa de molienda
- Ventana para la etapa de ensacado

Estas ventanas permitirán visualizar la situación de los motores, protectores térmicos, sensores y compuertas de cada etapa. También permitirán encender y apagar los motores y otros actuadores. Además, se mostrará una señal gráfica cuando se active un protector térmico u otra alarma. En cada ventana se colocarán gráficos ilustrativos de la maquinaria involucrada en cada sección y algunas animaciones para indicar si los motores están funcionando o no.

2.3.4.7 Ventana para ingresar el motivo que ocasionó una alarma

Cuando ocurra una alarma, se mostrará una indicación gráfica en las pantallas de interfase. Luego de corregir el problema que la ocasionó, el operador debe indicar al PLC que se solucionó el inconveniente pulsado sobre la señal de alarma. En este momento se desplegará esta ventana para que el usuario ingrese el motivo que

provocó la activación del sensor o protector térmico. Ésta información se guardará en la base de datos.

2.3.4.8 Ventanas para mostrar registros de la base de datos

En la base de datos se registrarán los eventos y las alarmas ocurridas. Se considera evento a la acción de prender o apagar un actuador de manera manual o mediante la lógica programada en el PLC. En la base de datos se registra la hora de encendido, la hora de apagado y la diferencia entre estos dos valores (tiempo de funcionamiento). Cuando ocurre una alarma se registra en la base de datos el código del sensor involucrado o del motor cuyo protector térmico se disparó, el usuario encargado de la operación y el motivo que provocó la alarma. Para mostrar esta información se utilizan dos ventanas:

- Ventana para mostrar los eventos ocurridos, y
- Ventana para mostrar las alarmas ocurridas

En estas ventanas se colocarán casilleros para que el usuario ingrese información para facilitar la búsqueda de la información requerida. Se podrá seleccionar el código del elemento y un intervalo de tiempo. Es decir, se podrán observar las alarmas o eventos de un elemento ocurridos entre dos fechas. Si no se ingresa un código se mostrarán os eventos de todos los elementos ocurridos en el intervalo de tiempo.

2.3.5 CONTROL DE ACCESO Y ADMINISTRACIÓN DE USUARIOS

Desde del software del sistema SCADA se puede operar toda la maquinaria de la fábrica y visualizar información sobre ésta. Es muy importante controlar el acceso de los usuarios al sistema para impedir que personas no autorizadas manipulen el programa. Alguien que no conozca la manera de utilizar el software podría encender o apagar una máquina accidentalmente, provocando incluso la parada de la planta. Además, al implementar un control de acceso es posible conocer quien estuvo

operando el sistema SCADA en todo momento. De manera de conocer la persona a cargo en el momento de los arranques, paradas y otras operaciones de la fábrica.

2.3.5.1 Implementación del control de acceso y administración de usuarios

La información requerida para realizar el control de acceso se almacena en dos tablas de la base de datos: accesos y usuarios. En la primera se almacena el tipo o nombre del acceso y la descripción. Mientras que en usuarios que guarda información de la persona como: nombre, apellido, nombre de usuario, clave y nombre de acceso.

Los tipos de accesos que se implementaron son:

- Administrador, y
- Operador.

La diferencia entre los dos tipos de accesos es que un administrador puede crear y eliminar usuarios mientras un operador no. Todas las demás funciones del software pueden ser utilizadas todos los usuarios.

Para ingresar al sistema se deben colocar el nombre de usuario y la contraseña de la persona. Esta información es validada comparándola con la almacenada en la base de datos. Si la información es correcta se busca en la base de datos el nombre de acceso de dicha persona y según esté se le permite o no la operación de las funciones del software.

Los usuarios de mayor jerarquía (administrador), pueden agregar o eliminar usuarios desde el software del sistema. Estas operaciones se realizan desde las pantallas creadas para este propósito.

2.3.6 SEGURIDADES PARA EVITAR EL APAGADO Y ENCENDIDO NO DESEADO DE MÁQUINAS

Al igual que en cualquier planta industrial, apagar o encender un motor o válvula, sin tomar las debidas precauciones puede ocasionar graves perjuicios. Por ejemplo, si se apaga la rosca de transporte de harina, para evitar que el producto se atore, se activa un sensor que ocasiona la parada de la planta. Para que esto no suceda, se implementarán seguridades con el objetivo de impedir cambiar el estado de un actuador de manera accidental.

Si para prender o apagar un motor, fuese suficiente hacer un clic en la pantalla, sería muy probable que ocurrieran acciones no deseadas. Por lo tanto, es indispensable que el operador necesita hacer algo más en para enviar una orden al PLC. La seguridad que se decidió utilizar en este sistema SCADA es que además de hacer clic se deba presionar simultáneamente una botón del teclado.

2.3.6.1 Implementación de seguridades para evitar el apagado y encendido no deseado de máquinas

Para poder prender un motor, además de presionar con el mouse sobre su interruptor, se deberá presionar la tecla C*ontrol* del teclado. Mientras que para apagar, se deberá presionar la tecla *Shift* (mayúsculas) al momento de pulsar sobre el interruptor. Estas seguridades también se aplican para los demás actuadores.

Para encender y apagar los actuadores se deben presionar teclas diferentes. De esta manera se pretende evitar que accidentalmente se envíe señales para prender y apagar al hacer doble clic sobre el interruptor.

2.3.7 COMUNICACIÓN ENTRE LOS COMPUTADORES Y LOS PLCS

Los PLCs instalados en la fábrica son los responsables de controlar la maquinaria. Ellos leen las señales de los sensores, selectores, pulsadores y paneles *opti-mate*, y según su valor realizan acciones de control.

El software del sistema SCADA lee el valor de las salidas del PLC para determinar el estado de los actuadores; y lee el valor de las entradas del PLC para determinar el estado de los sensores y de los botones del panel de control. Además, para encender o apagar un actuador, la aplicación cambia el estado de la salida del PLC.

Para poder controlar y supervisar la planta de producción desde el software del sistema SCADA es indispensable intercambiar datos entre los computadores y los PLCs.

El protocolo seleccionado para realizar la comunicación entre los computadores y los PLCs es OPC. Este protocolo trabaja en el formato cliente – servidor. El servidor es una aplicación que se encarga de leer y escribir los datos en el PLC. Otra aplicación, el cliente, se conecta con el servidor para conocer el valor de estos datos y para modificar su valor. Varios clientes se pueden conectar con un mismo servidor simultáneamente.

En este sistema SCADA, el cliente es la aplicación desarrollada en Visual Basic. El servidor OPC es *KEPDirect.* El fabricante de los PLC proporciona esta aplicación.

En esta sección se describe todo lo relacionado a la comunicación entre el programa de Visual Basic y los PLCs. Al inicio de la sección se encuentra el cálculo del tráfico de la red industrial. Luego se indica como se realizó la configuración del OPC server, la configuración el la aplicación (cliente OPC), el cableado necesario y el método empleado para la lectura y escritura de datos.

2.3.7.1 Cálculo del tráfico en la red

Es importante calcular el tráfico de la red industrial para determinar si el protocolo seleccionado provee un ancho de banda suficiente para la aplicación. En este proyecto existe un canal de comunicación independiente entre cada PLC y los computadores donde están las aplicaciones del SCADA, por lo que se debe realizar el análisis de manera independiente para cada uno. La figura a continuación muestra la conexión de los PLCs con los computadores.

Figura 2.13. Conexión de los PLCs con los computadores

La comunicación se realiza entre cada PLC y la computadora mediante protocolo RS232. La velocidad de comunicación es de 38400 bps y se emplean ocho bits de datos y un bit de parada. Por lo tanto, la trama de comunicación esta formada por ocho bits de datos, un bit de inicio, un bit de paridad y un bit de finalización. Lo que implica un tamaño de la trama de once bits.

El número máximo de tramas de datos de ocho bits que se pueden trasmitir en un segundo es igual a la velocidad de comunicación dividida para el tamaño de la trama, es decir:

$$\#\ de\ \ tramas\ \ _{(5\ bits)} = \frac{velocidad\ \ de\ \ comunicaión}{tamaño\ \ trama}$$

$$\#\ de\ \ tramas\ \ _{(5\ bits)} = \frac{38400\ \ bps}{11\ \ bits}$$

$$\#\ de\ \ tramas\ \ _{(5\ bits)} = 3490\ \ tramas/segundo$$

Por lo tanto, el número de tramas que va a transmitir cada PLC debe ser inferior a 3490. Para el funcionamiento del software del sistema SCADA es necesario conocer y cambiar el estado de las entradas y salidas del PLC, de los relés de control, de las localidades de memoria del PLC y de las etapas del programa del PLC.

2.3.7.1.1 Cálculo del tráfico de red para la sección molienda

En la Tabla se realiza el cálculo del número de tramas requerido en la sección molienda.

Descripción	Tipo	Cantidad	Tramas requeridas por dato	Total de tramas
Entradas digitales PLC	Boolean	8	1	8
Salidas digitales PLC	Boolean	96	1	96
Etapas PLC	Boolean	19	1	19
Relés de control	Boolean	8	1	8
Localidades de memoria	Word	10	1	10
Número total de tramas				141

Tabla 2.11. Cálculo del número de tramas para la sección molienda

2.3.7.1.2 Cálculo del tráfico de red para la sección limpieza

En la Tabla se realiza el cálculo del número de tramas requerido en la sección limpieza.

Descripción	Tipo	Cantidad	Tramas requeridas por dato	Total de tramas
Entradas digitales PLC	Boolean	8	1	8
Salidas digitales PLC	Boolean	64	1	64
Etapas PLC	Boolean	4	1	4
Relés de control	Boolean	11	1	11
Localidades de memoria	Word	13	1	13
Número total de tramas				100

Tabla 2.12. Cálculo del número de tramas para la sección limpieza

El número de tramas de la sección molienda (151) y el número de tramas de la sección limpieza (113) son menores a 3490. Por lo tanto, la capacidad de la red industrial es suficiente para la aplicación.

2.3.7.2 Cableado necesario para la comunicación

Los PLCs utilizados en la fábrica permiten comunicarse a través del protocolo RS-232 y a través del protocolo RS-422. Si se utiliza el segundo de los protocolos es necesario emplear dispositivos que conviertan las señales a señales RS-232 para poder comunicarse con los computadores.

El protocolo RS-232 permite transferir información hasta una distancia máxima de quince metros. El computador de la sección molienda se encuentra aproximadamente a once metros del PLC con el que se comunica. Mientras el PLC de la sección limpieza y el computador en el que corre el software del sistema SCADA de esta sección se encuentran a menos de cuatro metros.

La comunicación del sistema SCADA trabajará con el protocolo RS-232. Las características de este protocolo son adecuadas para la aplicación y no es necesario utilizar elementos adicionales.

El esquema del cable utilizado para la comunicación se muestra a continuación:

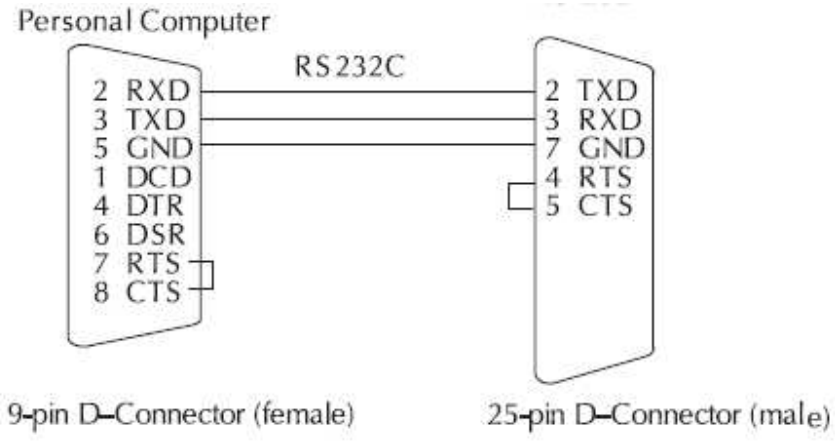


Figura 2.14. Esquema del cable utilizado para comunicar los PLCs con los computadores[17]

2.3.7.3 Configuración del servidor OPC

El servidor OPC permite la transferencia de datos entre los PLC y la aplicación del software SCADA. Para que funcione adecuadamente es necesario configurarlo. Se debe seleccionar un canal de comunicación, agregar un dispositivo y crear etiquetas de las variables que se desean leer. La configuración del OPC server se realizó mediante los siguientes pasos:

2.3.7.3.1 Configuración del canal de comunicación

Abrir el software KEPDirect for PLC y configurar el canal de comunicación de la siguiente manera:

a. Hacer clic sobre el icono *New project* ubicado en la parte superior izquierda de la pantalla. Luego pulsar sobre el texto que dice: "*click to add a channel*"

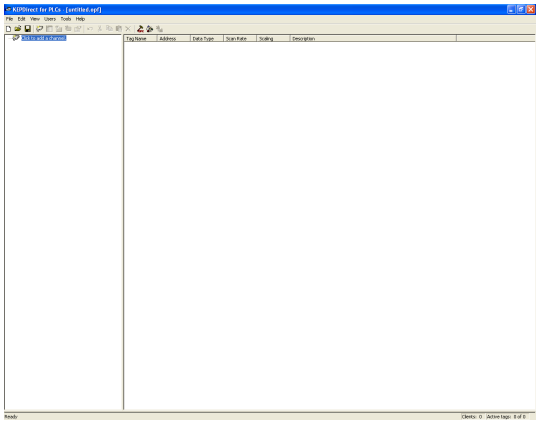


Figura 2.15. Ventana principal del software KEPDirect for PLCs

b. En la ventana que aparece escribir el nombre del canal

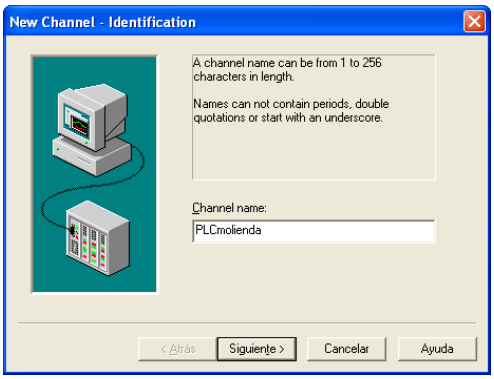


Figura 2.16. Ventana New Channel - Identification del software KEPDirect for PLCs

c. Seleccionar el nombre del driver de dispositivo que se va a utilizar. Los PLCs instalados en la fábrica permiten utilizar Direct – Net o K Sequence. Estos dos drivers pueden comunicarse a la misma velocidad máxima (38400 bps) con los PLCs. En esta aplicación se escogió emplear K-Sequence. Se consultó al fabricante del driver sobre las ventajas de utilizar cada uno de los drivers de dispositivo y recomendaron el empleo de K-Sequence. La única diferencia entre los dos es que K-Sequence permite realizar peticiones de datos de bit en bit mientras Direct - Net no lo permite.

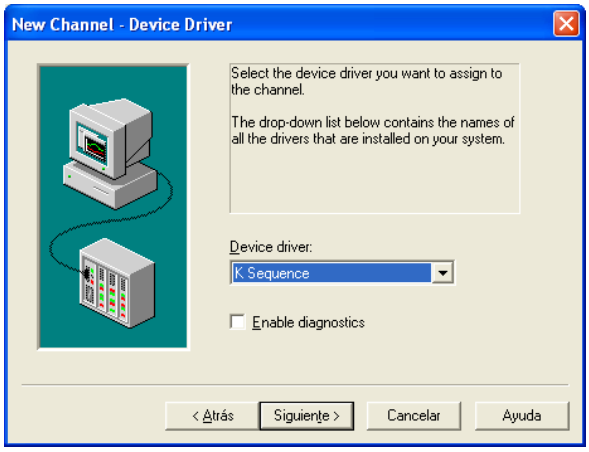


Figura 2.17. Ventana New Channel – Device Driver del software KEPDirect for PLCs

d. Configurar las propiedades de la comunicación. Para el sistema SCADA se empleará:

ID:	COM 1
Baud Rate:	38400
Data bits:	8
Parity:	None
Stop bits:	1
Flow Control:	None

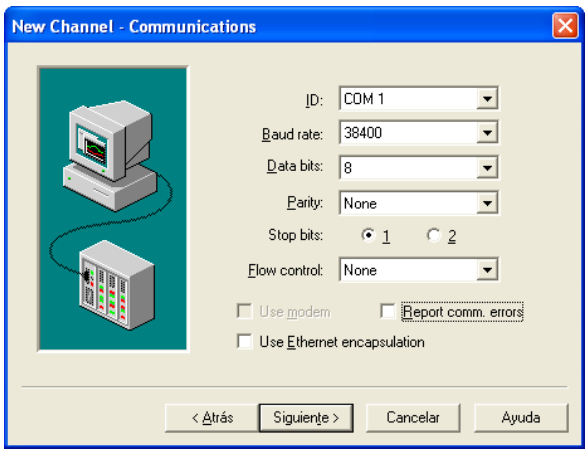


Figura 2.18. Ventana New Channel - Communications del software KEPDirect for PLCs

e. En la siguiente ventana que aparece se debe configurar las optimizaciones de escritura. Para este sistema se utiliza la opción *Write all values for all tags* (escribir todos los valores para todas las etiquetas). De esta manera se garantiza que se envían al PLC todos las señales deseadas. El valor del *Duty cicle* (ciclo de trabajo) utilizado es el recomendado por el fabricante del OPC driver, este valor es de 10 escrituras por cada lectura.

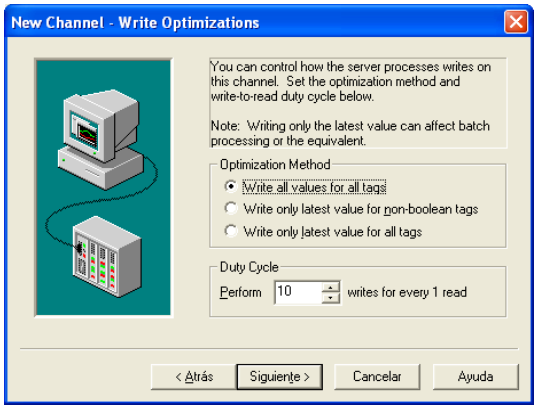


Figura 2.19. Ventana New Channel – Write Optimizations del software KEPDirect for PLCs

f. En la ventana que aparece haciendo clic sobre el botón siguiente se muestra la configuración seleccionada. La imagen a continuación muestra esta ventana. Al hacer clic sobre el botón finalizar se concluye la configuración del canal de comunicación.

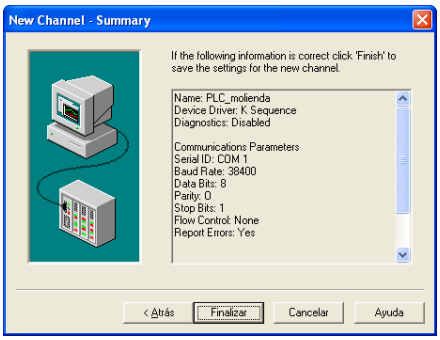


Figura 2.20. Ventana New Channel – Summary del software KEPDirect for PLCs

2.3.7.3.2 Agregar un dispositivo al canal de comunicación

a. Hacer clic sobre el texto “Click to add a device”

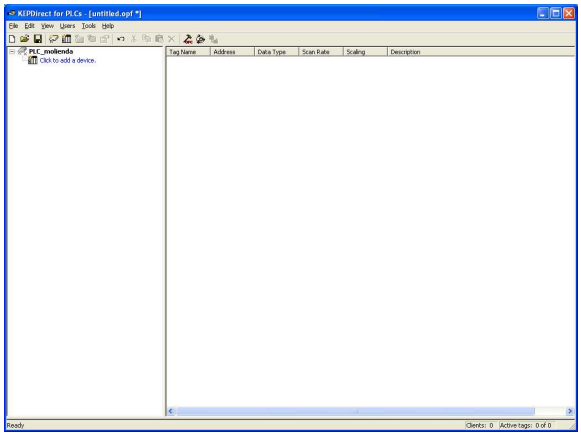


Figura 2.21. Ventana principal de KEPDirect for PLCs con el enlace para agregar dispositivos

b. Escribir el nombre del dispositivo en la ventana *New Device - Name*

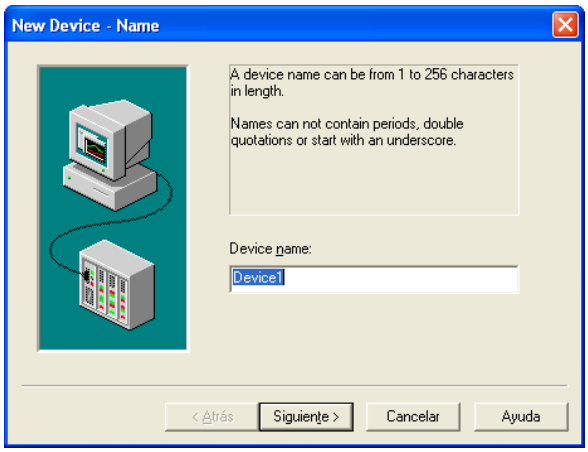


Figura 2.22. Ventana New Device –Name del software KEPDirect for PLCs

c. Seleccionar el modelo del PLC

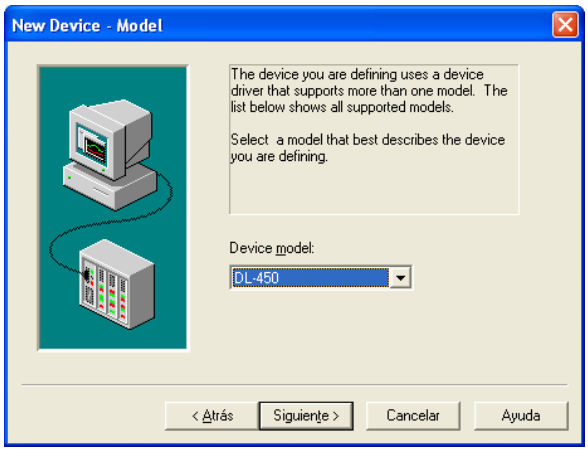


Figura 2.23. Ventana New Device – Model del software KEPDirect for PLCs

d. Asignar un identificar al PLC para poder identificarlos en caso de que se encuentren conectados en una red. La siguiente imagen muestra la pantalla destinada para este fin:

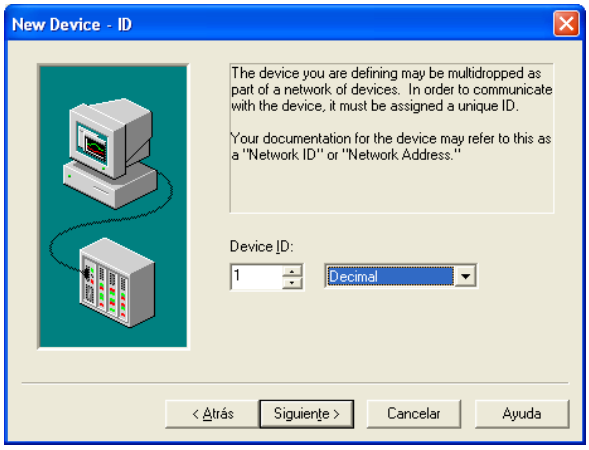


Figura 2.24. Ventana New Device – ID del software KEPDirect for PLCs

e. Configurar tiempos de comunicación. Para esta aplicación se mantuvieron los tiempos recomendados por el fabricante. Estos son de 1000 ms para *Request Timeout* (tiempo de espera se ha agotado) y de 3 *timeouts* sucesivos para indicar la falla en la comunicación.

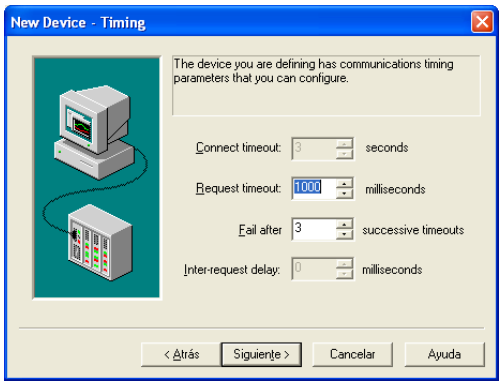


Figura 2.25. Ventana New Device – Timing del software KEPDirect for PLCs

f. En la ventana siguiente se habilitó la opción que permite anular la comunicación con los dispositivos que no responden. La utilidad de esta opción es no es estancar las comunicaciones si el PLC fallido se encuentra conectado a una red. Se habilitó esta opción considerando que Molino Electro Moderno tiene planificado realizar una ampliación en la sección molienda en los próximos meses. Para lo cual se implementará una rede de dos PLCs.

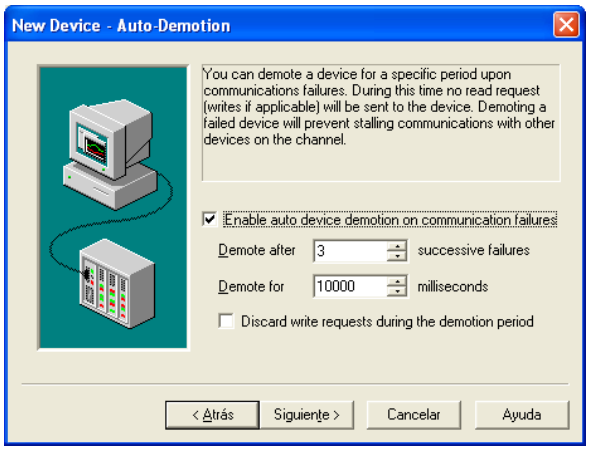


Figura 2.26. Ventana New Device – Demotion del software KEPDirect for PLCs

g. La siguiente ventana que se muestra permite activar la opción generar automáticamente una base de datos de *tags* o etiquetas al momento de conectar el PLC. Las etiquetas se crearon manualmente con el objetivo de incluir información relevante a su función, por lo tanto, no se habilita esta opción. Al crear las etiquetas manualmente, se agrega una breve descripción de cada una. De este modo, cuando se abre la aplicación del servidor OPC se pueden observar las etiquetas agregadas y la función de cada una.

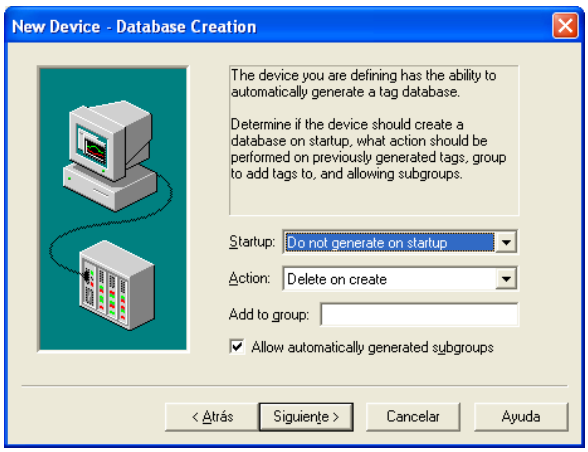


Figura 2.27. Ventana New Device – Database Creation del software KEPDirect for PLCs

h. En la siguiente ventana se puede seleccionar un archivo desde el cual se puede importar una lista de etiquetas. Estos archivos se pueden generar en software recientes de programación de PLCs. El software utilizado en la fábrica no dispone de esta función por lo que no se utiliza ningún archivo.

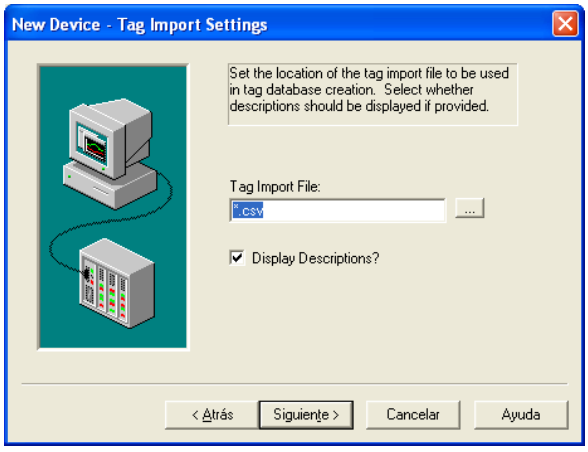


Figura 2.28. Ventana New Device – Tag Import Settings Creation del software KEPDirect for PLCs

i. La siguiente ventana muestra los parámetros seleccionados para el dispositivo. Al hacer clic sobre finalizar se concluye la configuración.

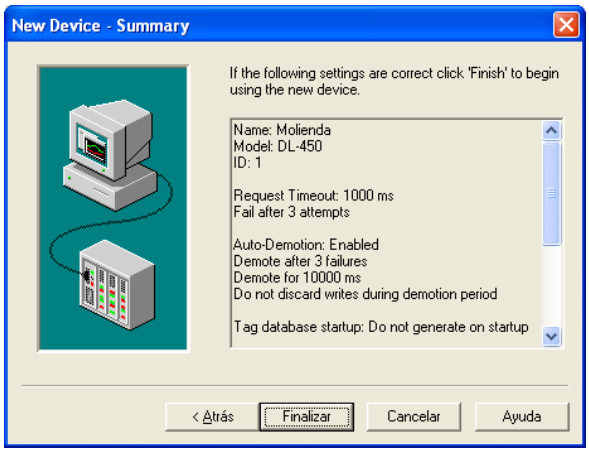


Figura 2.29. Ventana New Device – Sumary del software KEPDirect for PLCs

2.3.7.3.3 Agregar etiquetas

Se debe agregar una etiqueta o tag por cada variable que se desea leer desde el PLC. Al momento de crear las etiquetas se debe asignar un nombre y seleccionar la variable que se desea leer y que tipo de datos se desea recibir, por ejemplo BCD, boolean, word, etc.

Para agreagar las etiquetas se hace clic derecho en la sección derecha de la pantalla del software KEPDirect y se selecciona la opción *New tag.* Luego se llena la información requerida en la ventana que aparece. A continuación se muestra una imagen de dicha ventana.

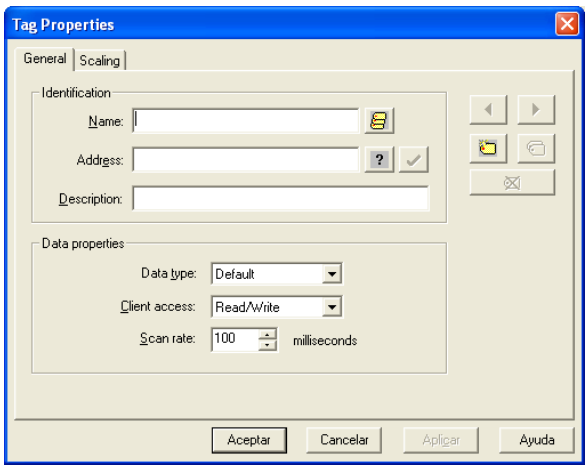


Figura 2.30. Ventana Tag Properties del software KEPDirect for PLCs

2.3.7.4 Configuración del cliente OPC

La aplicación desarrollada en Visual Basic constituye el cliente OPC en este sistema SCADA. En esta sección se describen los pasos necesarios que se deben seguir para poder realizar el intercambio de información entre el cliente OPC y el servidor OPC.

El procedimiento que se debe seguir para poder intercambiar datos entre el software del sistema SCADA con el servidor OPC son: establecer la conexión con el servidor, agregar un grupo, y agregar ítems.

Cuando el cliente OPC se conecta con el servidor le hace conocer que va a iniciar un intercambio de datos. Si ningún cliente se conecta con el servidor OPC, éste no solicita información al PLC.

Una vez establecida la comunicación entre el cliente y el servidor, el cliente debe hacer conocer al servidor con que datos va a trabajar. Esto se hace agregando un grupo e ítems. Cada ítem representa una variable del PLC. Los grupos son métodos para organizar los ítems. Como su nombre lo indica permite su agrupación. Para esta

aplicación resulta más fácil manejar los datos si no se los agrupa. En este sistema SCADA es indispensable conocer el estado de todas las variables, por lo que se trabaja con un solo grupo en el que se encuentra todos ítems utilizados.

A continuación se muestran los códigos utilizados para conectarse al servidor OPC, para agregar el grupo y para agregar ítems.

2.3.7.4.1 Código para conectarse con servidor OPC

El código que se emplea para realizar la conexión con OPC Server es el siguiente:

```
Dim ConnectedServerName As String
Dim ConnectedNodeName As Variant
Set ConnectedOPCServer = New OPCServer
ConnectedServerName = "AutomationDirect.KEPDirectServer"
ConnectedNodeName = ""
ConnectedOPCServer.Connect ConnectedServerName, ConnectedNodeName
```

2.3.7.4.2 Código para agregar el grupo

Para agregar el grupo se utiliza el código siguiente:

```
Set ConnectedServerGroup = ConnectedOPCServer.OPCGroups
ConnectedServerGroup.DefaultGroupIsActive = True
ConnectedServerGroup.DefaultGroupDeadband = 0
Set ConnectedGroup = ConnectedServerGroup.Add("Grupo")
ConnectedGroup.UpdateRate = 10
ConnectedGroup.IsSubscribed = Trae
```

2.3.7.4.3 Código para agregar elementos

El código empleado para agregar ítems es el siguiente:

```
Set OPCItemCollection = ConnectedGroup.OPCItems
OPCItemCollection.DefaultIsActive = True
OPCItemCollection.AddItems itemcount, OPCItemIDs, ClientHandles, ItemServerHandles, ItemServerErrors
```

La última línea del código es la encargada de agregar los ítems. Para agregarlos se debe escribir el número total de ítems que se agregarán (itemcount), el nombre del tag del ítem que se agrega (OPCItemIDs), y un número que se empleará en el software para leer los datos del ítem (ClientHandles).

Los otros dos datos son información que OPC Server devuelve. ItemServerHandles es un código que asigna el server a cada ítem agregado. La variable ItemServerErrors toma el valor de uno cuando no se puedo agregar el ítem. Si la operación fue exitosa esta variable toma el valor de cero.

2.3.7.5 Método utilizado para leer y escribir información

Con el objetivo de optimizar la comunicación, la transferencia de datos se realiza solo cuando existe un cambio de éstos en el PLC y además, se lee al mismo tiempo la mayor cantidad de bits posibles.

Se creó una etiqueta para cada salida y entrada del PLC. De esta manera se conoce el estado de cada uno y cuando cambian el OPC Server entrega el nuevo dato a la aplicación desarrollada en Visual Basic.

Las señales de los protectores térmicos y de las alarmas se ingresan al PLC por medio de una matriz, de manera de poder conectar muchas señales sin emplear demasiadas entradas del PLC. Los valores de estas señales se guardan en memoria, en conjuntos de ocho bits. El programa de Visual Basic lee el valor almacenado en la memoria y no el valor de cada señal individualmente.

La escritura de datos en el PLC se hace individualmente ya que no es necesario escribir muchos datos al mismo tiempo. Por ejemplo para prender un motor de arranque directo basta con colocar en uno lógico la salida del PLC que activa el contactor correspondiente; y para encender los motores con arranque estrella triángulo solo es necesario enviar la orden para activar la etapa del programa del PLC en la cual se desarrolla la rutina de arranque.

Cuando el valor de alguno de los datos cambia, el servidor OPC entrega el nuevo valor a la aplicación desarrollada en Visual Basic. Se escribió una rutina que se ejecuta automáticamente al detectar el cambio de una variable. La descripción de esta rutina se encuentra a continuación.

2.3.7.5.1 Lectura automática de datos que han cambiado

Para conocer el estado de las variables que se están monitoreando se utiliza una rutina que entrega los nuevos valores cada vez que cambian. La declaración de esta rutina es la siguiente:

Sub ConnectedGroup_DataChange(ByVal TransactionID As Long, ByVal NumItems As Long, ClientHandles() As Long, ItemValues() As Variant, Qualities() As Long, TimeStamps() As Date)

Cada vez que cambia alguno de los datos, el servidor OPC envía los nuevos valores al software desarrollado en Visual Basic. Mediante esta rutina se determina cual de las variables cambió de valor y se leen los nuevos datos.

Cuando se ejecuta esta rutina, el servidor OPC entrega algunas variables. Las que se utilizan en la aplicación de este sistema SCADA son:

- NumItems: indica el número de variables que cambiaron.

- ClientHandles: es el número que se asignó a cada ítem el momento de agregarlo. Los sirve para determinar cual fue el que cambio.
- ItemValues: en esta variable se encuentra almacenado el nuevo valor de los ítems.
- Qualities: indica si la calidad de los datos es buena o no.

Cuando se ejecuta esta rutina, primero se debe determinar cual de los ítems es el que cambió. Esta comprobación se la realiza comparando el valor de ClientHandles recibido con el valor que se tiene almacenado en el software. Por ejemplo, en el código mostrado a continuación, se está averiguando si la variable que cambió es la V200. El código que se asignó como ClientHandles el momento de añadir el ítem se encuentra almacenado en V200.numero_opc.

```
If ClientHandles(i) = v200.numero_opc Then
```

Esta condición se cumple cuando el dato que cambió es el V200. Por lo que dentro del lazo se ejecuta un código para determinar el nuevo valor de esta variable y realizar la operación requerida. Esta misma comparación se realiza para todos ítems que se añadieron.

2.3.7.5.2 Escritura de datos

Para poder realizar control desde la computadora es necesario modificar el valor de las salidas del PLC. Para cambiar el valor de una salida o localidad de memoria del PLC se utiliza el siguiente código:

```
Public Sub OPCItemWrite(itemcount, numeros_opc() As Integer, valores() As Integer)
    Dim SyncItemValues(10) As Variant
    Dim SyncItemServerHandles(10) As Long
    Dim SyncItemServerErrors() As Long
    Dim i As Integer
```

```
    For i = 1 To itemcount
        SyncItemServerHandles(i) = ItemServerHandles(numeros_opc(i))
        SyncItemValues(i) = Val(valores(i))
    Next i
    ConnectedGroup.SyncWrite itemcount, SyncItemServerHandles, SyncItemValues,
SyncItemServerErrors

End Sub
```

Para ejecutar esta rutina se la debe llamar incluyendo el número de datos que se desean escribir, el código ClientHandles de cada dato y el nuevo valor. En el código que se muestra a continuación se va a escribir un dato. A la variable C200 se la va a colocar en uno.

```
OPCItemWrite 1, C200.numero_opc, 1
```

El número uno ubicado a la derecha del código *OPCItemWrite* indica que se va a escribir un dato. La siguiente información (C200.numero_opc) indica a que variable se va a modificar. Y el último dato indica el valor que se desea asignar a la variable.

2.3.7.5.3 Lectura manual de datos

En esta aplicación, en algunos casos, no se puede esperar que una variable cambie de valor para conocer su estado. A veces, es importante poder determinar su valor en un momento dado. Por ejemplo, cuando se cambia el modo de operación de control a supervisión se deben leer los valores de los selectores del tablero de control. De esta manera, en la pantalla del sistema SCADA se muestran en la misma selección en la que se encuentran en el tablero.

Para leer el valor de las variables del PLC se utiliza la siguiente subrutina:

```
Public Sub OPCItemSyncRead(itemcount As Integer, numeros_opc() As Integer)

    Dim SyncItemValues() As Variant
    Dim SyncItemServerHandles(10) As Long
    Dim SyncItemServerErrors() As Long
    Dim i As Integer

    For i = 1 To itemcount
    SyncItemServerHandles(i) = ItemServerHandles(numeros_opc(i))
    Next i

    ConnectedGroup.SyncRead OPCDevice, itemcount, SyncItemServerHandles,
SyncItemValues, SyncItemServerErrors

    If SyncItemServerErrors(1) = 0 Then
        For i = 1 To itemcount
        valores_sync_read(i) = SyncItemValues(i)
        Next i
    End If

End Sub
```

Para llamar a esta rutina se debe indicar el número de datos que se desean leer y el código ClientHandles de cada uno. En el ejemplo que se muestra a continuación se va a leer el valor de X10. El código ClientHandles de esta variable está guardado en X10.numero_opc.

```
OPCItemSyncRead 1, X10.numero_opc
```

2.3.8 DESARROLLO DE LAS VENTANAS DE INTERFASE DEL OPERADOR

En esta sección se describe el proceso seguido para realizar las pantallas de interfase del operador. Se muestran el proceso de la creación de las imágenes, el desarrollo de las animaciones, la implementación de la barra de herramientas y menú, la implementación de los interruptores y del resto de los componentes de estas ventanas.

2.3.8.1 Desarrollo de las imágenes

En las pantallas de interfase de operador hay muchas imágenes. Hay representaciones gráficas de la maquinaria, indicaciones de estado, interruptores e íconos utilizados para cambiar de pantalla. Estas imágenes son de varios tipos. En la barra de herramientas se utilizan imágenes con extensión .ico; los interruptores son de tipo bmp; y las imágenes de la maquinaria, los actuadores, los silos y camiones son tipo gif con fondo transparente.

A continuación se describe como se desarrolló cada tipo de imagen:

2.3.8.1.1 Iconos de la barra de tareas

Las imágenes de la barra de herramientas deben ser iconos, es decir dibujos guardados con extensión .ico. Para obtener este tipo de imágenes, se realizaron dibujos que representan las funciones que cumple cada botón, y se guardaron como tipo icono de ocho bits mediante el software de manipulación de imágenes *Gimp* versión 2.2.13.

2.3.8.1.2 Imágenes bmp

Estas imágenes se crearon utilizando el programa *Paint.* Se utilizó este programa por su facilidad de manejo y porque la baja complejidad de las imágenes permitía el desarrollo en un software de pocas funciones.

2.3.8.1.3 Imágenes gif con fondo transparente

Para obtener pantallas con buena apariencia es indispensable utilizar imágenes con fondo transparente. De esta manera se pueden colocar imágenes encima de otras sin dañar la apariencia de la pantalla. Utilizar estas imágenes también permite cambiar el color del fondo de la pantalla fácilmente.

Las imágenes de la maquinaria, de los transportadores y de las compuertas neumáticas se dibujaron en *Autocad 2007.* Luego se hizo transparente el fondo de las imágenes utilizando el software *Gimp* versión 2.2.13.

2.3.8.2 Implementación de animaciones

Con el objetivo de que el operador pueda identificar rápidamente que máquinas se encuentran funcionando en un determinado momento, y para mejorar la apariencia de las pantallas de interfase de operador, se colocaron algunas animaciones. Se decidió animar los transportadores verticales y horizontales. La animación de estos elementos no es muy complicada y llaman la atención debido a su longitud y espacio que ocupan en la pantalla. La animación de otras máquinas es más complicada y al haber muchos objetos en movimiento dentro de la pantalla, se podría distraer el operador y le resultaría más difícil visualizar el cambio de estado de los elementos. En el software se implementó las animaciones de dos maneras: gifs animados y mediante programación propia de *Visual Basic*. A continuación se describen estos dos métodos:

2.3.8.2.1 Realización de GIFs animados

Una animación es una secuencia de imágenes que se muestran en un intervalo de tiempo. Cada cierto tiempo se cambia la imagen actual por una nueva en la cual se muestra la misma figura en una nueva posición. Los GIFs animados son un conjunto de imágenes formado GIF, que poseen un código. Utilizando este código se puede

programar el orden de rotación de las imágenes y el tiempo que permanece visible cada una.

Las animaciones de los tornillos sin fin se realizó utilizando este método. Para lograrlo se dibujaron tornillos con la rosca en diferente posición, se guardaron con formato GIF, y utilizando el software *Microsoft Gif Animador* se agrego el código. Luego de realizar varias pruebas se obtuvieron resultados satisfactorios utilizando cuatro imágenes y permitiendo que cada una esté visible por 0.15 segundos.

A continuación se muestran las imágenes GIF utilizadas para construir la animación:

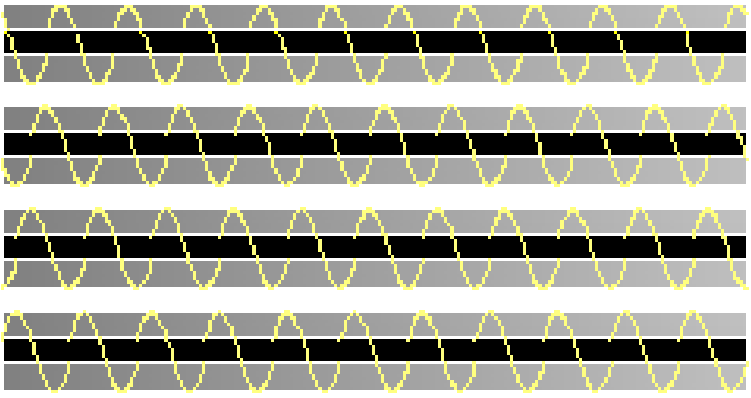


Figura 2.31. Imágenes que conforman la animación de un tornillo sin fin que gira de izquierda a derecha

Cuando las imágenes rotan en el orden mostrado en la figura anterior, se ve como si la rosca moviera el producto de izquierda hacia derecha. Para que se vea que el producto se mueve de derecha a izquierda, el orden de las figuras debe ser invertido, mostrando primero la imagen inferior y al último la superior. A continuación se muestran las imágenes en el orden adecuado para lograr esta animación:

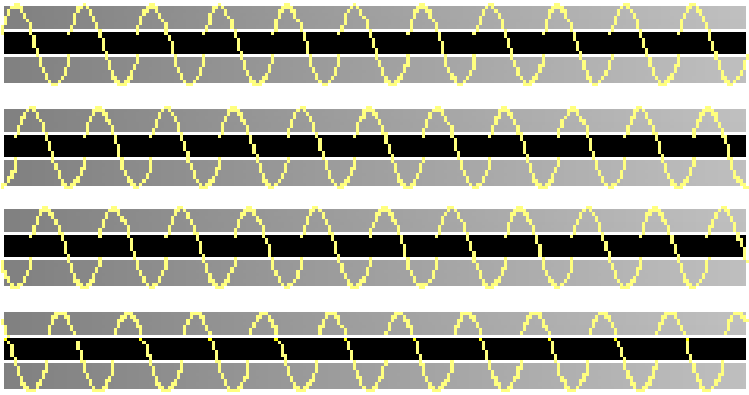


Figura 2.32. Imágenes que conforman la animación de un tornillo sin fin que gira de derecha a izquierda

2.3.8.2.2 Realización de animación mediante programación en Visual Basic

Las animaciones realizadas en Visual Basic se desarrollaron empleando temporizadores. En cada animación se utiliza un temporizador diferente que se activa el momento en que inicia la animación. Para detenerla se para el temporizador. Cuando se cumple el valor preestablecido en cada timer, se ejecuta una rutina en la cual se cambia la posición de la figura. Por ejemplo, para lo elevadores, cada cien milisegundos, la posición de un círculo que simula el producto, sube ocho píxeles.

Para aumentar la velocidad de la animación se aumenta el número de píxeles que la figura se mueve cada vez que se cumple el tiempo preestablecido.

2.3.8.3 Implementación de la barra de herramientas y menú

Con el objetivo de facilitar el manejo del programa se implementó una barra de herramientas y un menú. Éstos se utilizan para cambiar de ventanas y ejecutar las funciones del sistema. Ambos están implementados en el formulario madre (MDIform). En los formularios hijos aparece todo lo que se coloca en el formulario madre, por lo que la barra de tareas y el menú aparecen en todas las ventanas. A continuación se describen los componentes y el funcionamiento de ambos.

2.3.8.3.1 Barra de herramientas

La barra de tareas consiste en un conjunto de imágenes ubicadas en la parte superior de la pantalla, debajo del menú, de manera similar que en cualquier software comercial. En ella se colocaron iconos que simbolizan las funciones que se ejecutan al hacer clic sobre ellos.

Figura 2.33. Barra de herramientas del software del sistema SCADA para la sección molienda

Figura 2.34. Barra de herramientas del software del sistema SCADA para la sección limpieza

2.3.8.3.2 Funciones de los iconos de la barra de herramientas

A continuación se describe la función de cada icono de la barra de herramientas:

Prelimpieza

Muestra la pantalla mediante la cual se supervisa y controla la etapa de prelimpieza de la materia prima.

Limpieza

Muestra la pantalla mediante la cual se supervisa y controla la etapa de limpieza.

Molienda

Muestra la pantalla en la cual se controla y supervisa la etapa de molienda de trigo.

Empaque

Muestra la pantalla utilizada para el control y supervisión de la etapa de empaque.

Mostrar motores

Al hacer clic sobre éste icono, se muestran, si están ocultos, los dibujos de los motores. Cuando los motores están visibles se deshabilita este botón por lo que no puede ser presionado. También se deshabilita el comando equivalente del menú. Debajo de este texto se muestra dos imágenes de la pantalla de molienda, en una se muestran los motores, mientras que en la otra están ocultos.

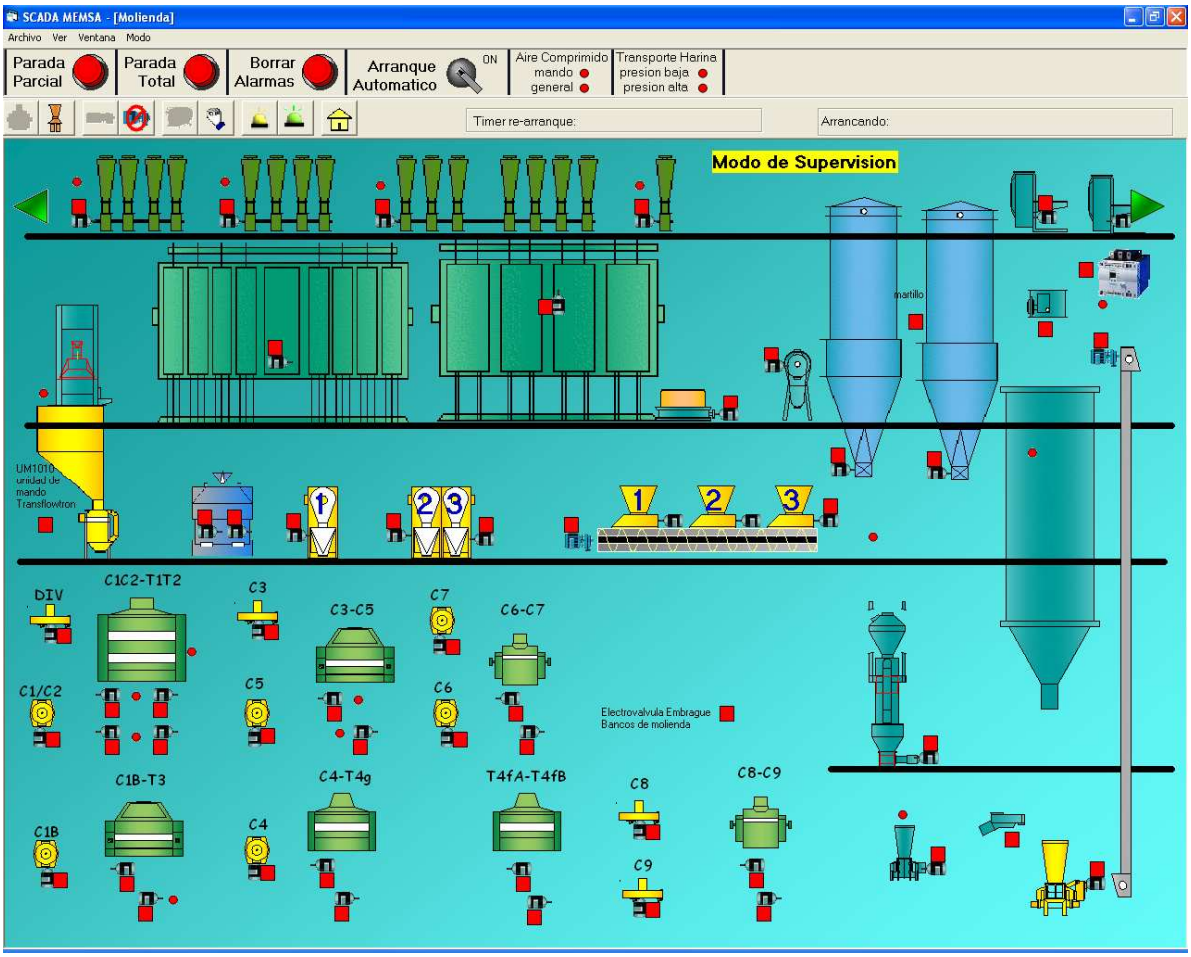


Figura 2.35. Pantalla de molienda con motores visibles

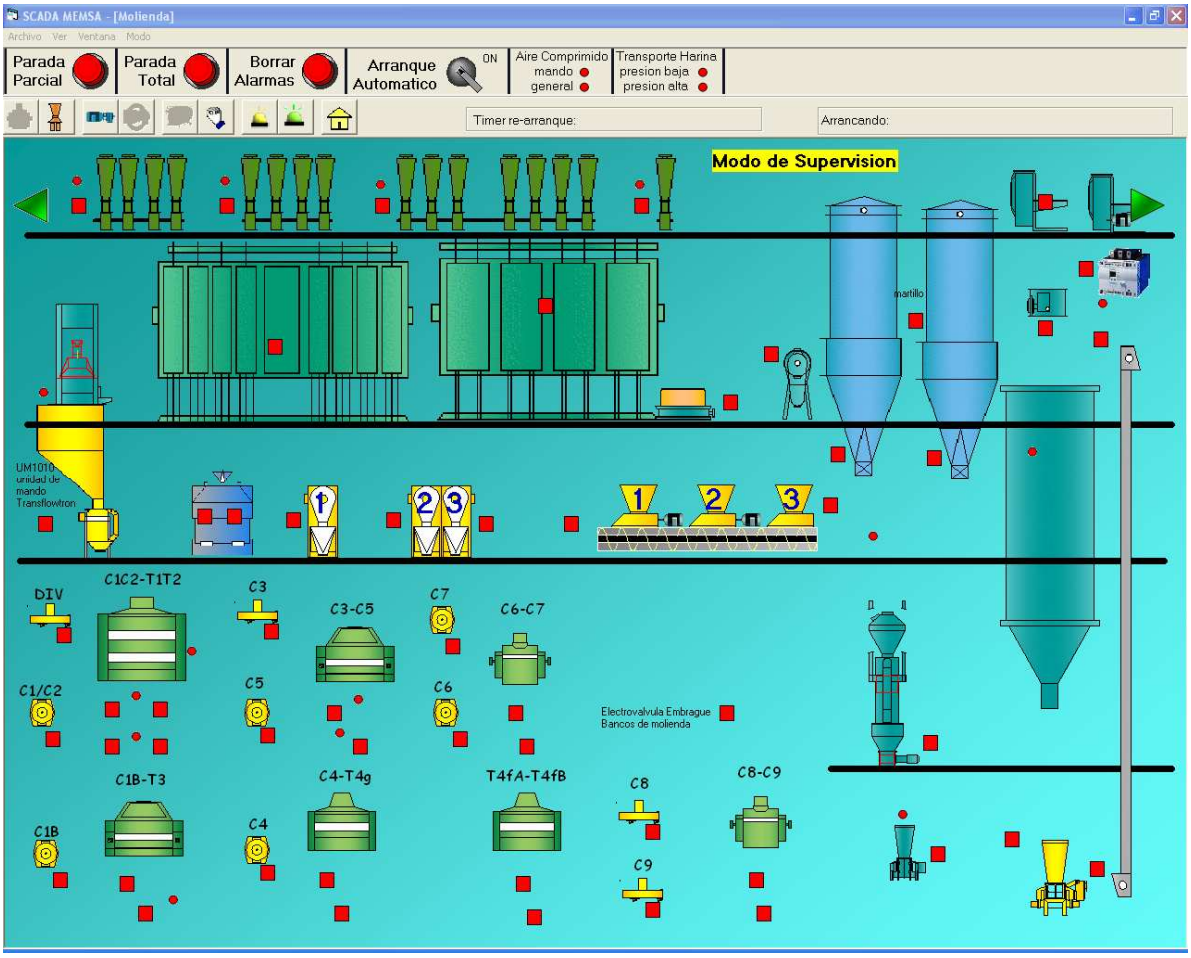


Figura 2.36. Pantalla de molienda sin motores visibles

Ocultar motores

Si los dibujos de los motores están visibles, al pulsar estén botón, éstos se ocultan y se deshabilita el botón y el comando equivalente del menú.

Modo de supervisión

Se ingresa al modo de supervisión cuando se pulsa el botón.

Modo de control

Cuando se hace clic sobre este icono se ingresa al modo de control.

Alarmas

Al hacer clic sobre este icono se muestra la pantalla en la cual se visualizan las alarmas ocurridas.

Eventos

Haciendo clic en este botón se muestra la pantalla en la cual se encuentra la lista de los eventos ocurridos.

Principal

Cuando se pulsa sobre este icono se muestra la pantalla de administración del programa.

2.3.8.3.3 Menú

El menú funciona de la misma manera que los menús desplegables que poseen muchos programas. Es un conjunto de palabras que al hacer clic sobre ellas muestran una lista de opciones o realizan una función.

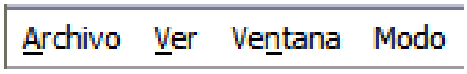


Figura 2.37. Menú del software del sistema SCADA

2.3.8.3.4 Funciones del menú

A continuación se describen las funciones de cada botón del menú:

Archivo

Dentro del menú *Archivo* se encuentra el comando *Salir,* el cual al ser presionado ejecuta la rutina para terminar la aplicación.

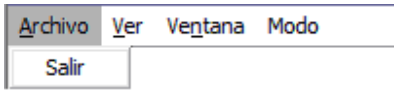


Figura 2.38. Menú Archivo

Ver

Dentro de este menú encontramos los siguientes comandos:

- Mostrar motores: Realiza la misma función que el botón *mostrar motores* de la barra de tareas.
- Ocultar motores: Cumple la misma función que el botón de la barra de tareas que tiene el mismo nombre.
- Velocidad animación: Mediante este menú se puede seleccionar la velocidad de las animaciones de los elevadores. Hay tres opciones: velocidad baja, velocidad media y velocidad alta.

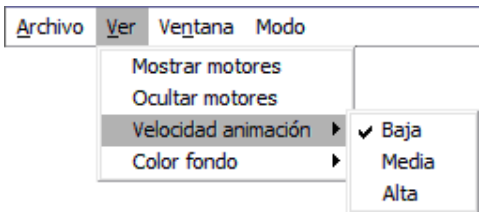


Figura 2.39. Menú velocidad animación

- Color fondo

  Este menú se utiliza para cambiar el color del fondo de las ventanas de manera de tener una mejor visibilidad de las imágenes. Las opciones disponibles son los colores: verde, azul y blanco.

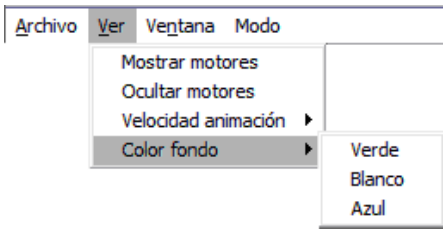


Figura 2.40. Menú color fondo

Ventana

Este menú se utiliza para seleccionar la ventana que se desea observar. Las opciones disponibles son:

- Principal,
- Prelimpieza,
- Molienda,
- Empaque,
- Eventos, y
- Alarmas.

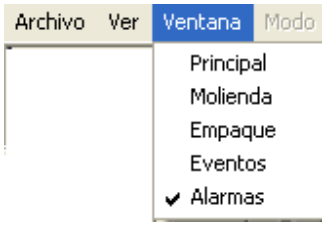


Figura 2.41. Menú ventana del software del sistema SCADA de la sección molienda

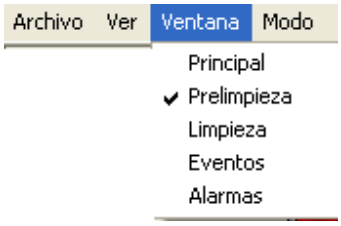


Figura 2.42. Menú ventana del software del sistema SCADA de la sección limpieza

Modo

Este menú permite seleccionar si se desea operar el sistema en modo de supervisión o en modo de control manual.

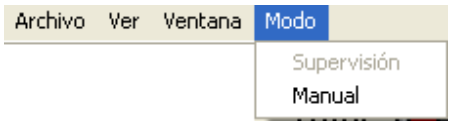


Figura 2.43. Menú modo

2.3.8.4 Botones para prender y apagar motores e indicar su estado

Desde el software del sistema SCADA se va a realizar el control y la supervisión de todos los motores de la planta de producción. Para realizar la supervisión es necesario establecer un método para determinar rápidamente su estado. De esta

manera se pretende conocer que motores se encuentran encendidos y cuales se encuentran apagados en un determinado momento. Para realizar el control es necesario poner activar y desactivar fácil y rápidamente los elementos a controlar.

Una imagen cambiará de color según el estado de los motores. Esta imagen también se empleará como interruptor. De este modo se pretende que la aplicación sea fácil de utilizar y no existan demasiados elementos en la pantalla que dificulten la visualización y operación del sistema.

A continuación se describe la implementación de los indicadores de estado e interruptores de los motores.

2.3.8.4.1 Implementación de botones para prender y apagar motores e indicar su estado

El estado de los motores se indica mediante símbolos gráficos. Cerca de la imagen de los motores está un cuadrado que es de color rojo cuando el motor está apagado y verde cuando está encendido. Estos cuadrados también se utilizan para encender y apagar los motores. Funcionan como interruptores, si el programa se encuentra en modo de control manual, al hacer clic sobre ellos y presionar la tecla adecuada del teclado (S*hift* para apagar y *Control* para encender*)*, cambia el estado del motor.

Cuadrado Verde

El cuadrado aparece de este color cuando el motor está encendido. Al hacer clic sobre el, cuando está activado el modo de control manual y se presiona la tecla *Shift,* se procede a apagar el motor.

Cuadrado Rojo

El cuadro rojo indica que el motor se encuentra apagado. Para encenderlo se hace clic sobre ésta figura y se presiona la tecla *Control.* Además es necesario estar en modo de control manual.

2.3.8.5 Encendido y apagado de motores

Para cumplir con los objetivos de la implementación del sistema SCADA es necesario poder encender y apagar, desde la aplicación, todos los motores de la planta. Para escribir el código que acciona los motores desde la aplicación, primero se analizaron los motores utilizados, sus métodos de arranque y las consideraciones que se deben tomar para encenderlos y apagarlos. Luego siguiendo estas consideraciones se procedió a la elaboración del código.

En la fábrica de Molino Electro Moderno S.A. existen motores que se arrancan directamente y otros cuyo arranque se realiza por el método estrella – triángulo. Uno de los motores de la planta se arranca mediante un arrancador suave debido a su elevada potencia.

Una de las máquinas requiere un tratamiento especial. En la fábrica existe una máquina vibratoria que funciona con dos motores que se deben encender y apagar simultáneamente para que trabaje correctamente. Este hecho se consideró al momento de escribir el código que acciona esta máquina.

Otra consideración que se tomó fue sobre las máquinas vibratorias. En el proceso de producción de harina se utilizan máquinas para tamizar los productos. Estas máquinas trabajan vibrando. Para encender estas máquinas es importante que se encuentren totalmente paradas, es decir, que haya parado su vibración. Por lo tanto, luego de apagar este tipo de máquinas se debe esperar un intervalo de tiempo antes de poder encenderlas nuevamente.

A continuación se describe la implementación del encendido y apagado de los motores de la planta.

2.3.8.5.1 Implementación de encendido y apagado de motores con arranque directo

Los motores con arranque directo se prenden y apagan mediante un contactor activado por el PLC. Para prender o apagar estos motores, se cambia el estado de la salida correspondiente. Cuando el operador ordena prender o apagar un motor, se envía la señal al PLC para que modifique el estado de la salida.

Para encender un motor se coloca en uno lógico la salida correspondiente del PLC mediante el siguiente código:

```
salida(1) = info_motor(Index).numero_opc
valor(1) = 1
MDIForm1.OPCItemWrite 1, salida(), valor()
```

Para colocar en cero lógico una salida del PLC con el objetivo de apagar un motor, se utiliza el siguiente código:

```
salida(1) = info_motor(Index).numero_opc
valor(1) = 0
MDIForm1.OPCItemWrite 1, salida(), valor()
```

2.3.8.5.2 Implementación del encendido y apagado de motores con arranque estrella – triángulo

Los motores de mayor potencia son arrancados empleando el método estrella - triángulo. Los programas de los PLCs realizan las rutinas para la conmutación de los contactores de línea, contactor de conexión estrella y contactor de conexión delta para llevar acabo el arranque del motor.

Para encender estos motores desde el software del SCADA se debe activar la etapa del programa del PLC encargada de realizar esta operación. El código que lo realiza es el siguiente:

```
salida(1) = numero_opc_etapa
valor(1) = 1
MDIForm1.OPCItemWrite 1, salida(), valor()
```

Este código solo se ejecuta luego de verificar que el motor involucrado está apagado. Si el motor estuviese prendido y se envía la orden de arranque, se activaría el contactor de conexión estrella mientras está activado el contactor de conexión triángulo, provocando un corto circuito. El código utilizado para comprobar que el motor está apagado es:

```
If info_motor(Index).valor = False Then
        prender_yd (info_motor(Index).numero_opc_etapa)
End If
```

Para apagar los motores con arranque estrella – triángulo, se coloca en cero las salidas del PLC encargadas de maniobrar los tres contactores involucrados en la operación. De esta manera se puede apagar sin ningún problema el motor en cualquier momento, incluso durante la etapa de arranque, mientras todavía se encuentra en conexión estrella. Cuando se apaga un motor de este tipo también se apaga la etapa encargada del arranque. De este modo, si al momento de apagarlo no ha terminado el proceso de arranque, éste se suspende. El código encargado de realizar esta operación es:

```
salidas(1) = info_motor(Index).numero_opc
salidas(2) = info_motor(Index + 1).numero_opc
salidas(3) = info_motor(Index + 2).numero_opc
```

```
salidas(4) = info_motor(Index).numero_opc_etapa
valores(1) = 0
valores(2) = 0
valores(3) = 0
valores(4) = 0
MDIForm1.OPCItemWrite 4, salidas(), valores()
```

2.3.8.5.3 Implementación de arranque y parada de motor con arrancador suave

Uno de los motores de la fábrica, por ser de elevada potencia, es arrancado empleando un arrancador suave (o *soft starter* en inglés). Para encender el motor es necesario activar un contactor que alimenta al arrancador suave y dar la señal de arranque (*start)*. Estas operaciones se realizan en una de las etapas del programa del PLC, por lo que basta activar esta etapa para poner en funcionamiento al motor. Para activar la etapa del PLC que realiza esta operación se emplea el mismo código utilizado para activar las etapas encargadas de realizar el arranque estrella - triángulo de los motores.

La parada del motor se realiza desactivando el contactor que energiza al arrancador suave y el contactor que envía la señal de arranque al soft starter. El código que realiza esta operación es:

```
valores(1) = 0
valores(2) = 0
salidas(1) = 48
salidas(2) = 35
MDIForm1.OPCItemWrite 2, salidas(), valores()
```

2.3.8.5.4 Implementación de encendido y apagado de motores que se debe realizar la operación de manera simultanea

Algunas de las máquinas de la fábrica trabajan con dos motores, los cuales se deben encender y apagar al mismo tiempo. Para conseguir el arranque y parada simultánea, se debe modificar al mismo tiempo las salidas del PLC encargadas de accionar los contactores correspondientes a dichos motores. Para el arranque, el programa desarrollado en Visual Basic coloca en uno lógico las salidas del PLC correspondiente a los dos motores, y para apagarlos se colocan en cero lógico.

El código encargado de encender los motores simultáneamente es:

```
valores(1) = 1
valores(2) = 1
salidas(1) = 27
salidas(2) = 28
MDIForm1.OPCItemWrite 2, salidas(), valores()
```

Para apagarlos se utiliza el siguiente código:

```
valores(1) = 0
valores(2) = 0
salidas(1) = 27
salidas(2) = 28
MDIForm1.OPCItemWrite 2, salidas(), valores()
```

2.3.8.5.5 Implementación de encendido y apagado de motores que se debe esperar un intervalo de tiempo entre el apagado y encendido

Dentro del proceso de producción, se emplean máquinas que utilizan motores para provocar su vibración, con el objetivo de tamizar los productos. Estas máquinas deben estar en reposo antes de ser encendidas. Por este motivo se debe esperar un periodo de tiempo luego de apagarlas antes de poner arrancarlas nuevamente. Esta seguridad está implementada en el software del sistema SCADA. Si el operador

intenta encender la máquina antes de que transcurra el tiempo de espera, aparece un mensaje indicado que no es posible llevar a cabo la operación. Cuando se apagan los motores de estas máquinas se inicia un temporizador. Mientras este temporizador no llegue al valor de tres minutos, no es posible arrancar los motores nuevamente.

El cuadro de mensaje que aparece si se intenta encender la máquina antes de que transcurra el tiempo de espera para que alcance el reposo es el siguiente:

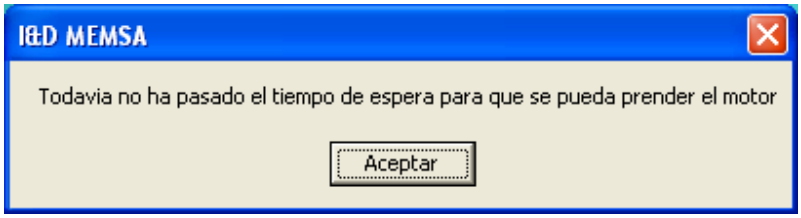


Figura 2.44. Mensaje indicando que no se debe esperar a que transcurra un intervalo de tiempo antes de encender el motor

2.3.8.6 Apertura y cierre de las compuertas

En la planta de producción de Molino Electro Moderno S.A. se emplean varias compuertas. Existen compuertas en las entradas y salidas de los silos de almacenamiento de trigo y en las entradas y salidas de los silos de trigo húmedo. Mediante estas compuertas se selecciona donde almacenar el trigo, de donde sacar el trigo para moler y también permiten trasladar trigo de un silo a otro.

Anteriormente, el PLC realizaba el control de las compuertas. Luego de una ampliación de la planta de producción, el número de compuertas aumentó, y el control mediante botones en el tablero de control se volvió complicado. La empresa elaboró un software mediante el cual se hacía el control desde un computador. El software leía las señales de los sensores y escribía en las salidas del PLC a través del protocolo DDE. Las señales de los sensores y las señales de activación de

compuertas están conectadas al PLC pero éste no realiza el control de dichos elementos.

El sistema SCADA implementado en la fábrica controlará las compuertas de la misma manera, desde los programas que corren en los computadores. En la aplicación desarrollada en Visual Basic se escribirá el código que controle estos elementos.

2.3.8.6.1 Implementación del control de las compuertas

Las compuertas instaladas en la fábrica de Molino Electro Moderno S.A. se accionan mediante cilindros neumáticos. Hay dos tipos de cilindros neumáticos: bi-estables y mono-estables. Las primeras se accionan entregando un pulso a uno de sus terminales y se cierran con pulso en otro terminal. El pulso debe permanecer hasta que la compuerta se abra o cierre completamente. Los cilindros mono-estables trabajan con una sola señal, cuado se coloca un uno lógico se abre y cuando la señal es cero lógico el cilindro se cierra.

Los cilindros poseen dos sensores, uno se activa cuando la compuerta está abierta y el otro cuando ésta se cierra. Estas señales se utilizan para realizar el control de las compuertas desde Visual Basic.

Para accionar un cilindro bi-estable se coloca en uno lógico la salida del PLC correspondiente a la señal de apertura del cilindro, y se mantiene esta señal hasta recibir la señal del sensor indicando que la compuerta está totalmente abierta. Para cerrarla se sigue el mismo procedimiento, colocando en uno lógico la señal del PLC adecuada hasta recibir la confirmación del sensor indicando que ya alcanzó el estado deseado. A continuación se muestra el código utilizado para abrir este tipo de válvulas:

```
valor(1) = 1
numero_opc(1) = info_elemento(Index).numero_opc
MDIForm1.OPCItemWrite 1, numero_opc, valor
```

Cuando se detecta la señal del sensor, se coloca en cero lógico la salida del PLC mediante el siguiente código:

```
salida(1) = info_elemento(Index).numero_opc
valor(1) = 0
MDIForm1.OPCItemWrite 1, salida(), valor()
```

Los cilindros mono estables se abren colocando en uno lógico la señal de activación y se cierran colocándola en cero lógico. Esto se realiza modificando directamente la salida del PLC correspondiente mediante los siguientes códigos:

Código para abrir:

```
valor(1) = 1
numero_opc(1) = info_elemento(Index).numero_opc
MDIForm1.OPCItemWrite 1, numero_opc, valor
```

Código para cerrar:

```
valor(1) = 0
numero_opc(1) = info_elemento(Index).numero_opc
MDIForm1.OPCItemWrite 1, numero_opc, valor
```

Las representaciones gráficas del estado de las compuertas son las siguientes:

Compuerta roja

Se muestra esta figura cuando la compuerta está totalmente cerrada. La imagen se hace visible cuando se detecta la señal del sensor que confirma este estado.

Compuerta verde

La imagen es de color verde cuando se detecta la señal del sensor indicando que la compuerta está completamente abierta.

Se conoce que una compuerta se atoró cuando luego de dar la señal para cambiar su estado transcurre demasiado tiempo y no se activa el sensor adecuado. Si se hace clic nuevamente sobre la imagen de la compuerta se envía al PLC la señal opuesta a la anterior. Si se estaba abriendo la compuerta y se vuelve a pulsar sobre su imagen, se enviará la señal para que se cierre, y viceversa. De esta manera, si la compuerta se atora cuando se estaba abriendo, al enviar nuevamente una señal, la compuerta se cerrará, permitiendo solucionar el problema que provocó que se atore.

2.3.8.7 Implementación de sensores

Para realizar el control de la maquinaria es indispensable utilizar sensores. Estos elementos nos permiten conocer el estado de la maquinaria y dan señales de alarma. Leyendo su estado se pueden realizar acciones de control como paradas totales o parciales o dar señales de alerta al operador para que tome medidas preventivas.

En la fábrica se encuentran instalados varios sensores. Se utilizan para detectar el giro de los elevadores, para determinar niveles mínimo y máximo de producto en tolvas, si una compuerta se encuentra abierta o cerrada, etc. Para indicar su estado se utilizan imágenes de pequeños círculos rojos y verdes.

Círculo rojo

Indica que el sensor se encuentra en cero lógico, y cuando se trata de un fin de carrera de una compuerta indica que está totalmente cerrada.

Círculo verde

Cuando la imagen del sensor tiene este color significa que su estado es uno lógico. Si se trata de un fin de carrera, indica que la compuerta está totalmente abierta.

El código mostrado a continuación se ejecuta cuando se detecta el cambio de estado, de cero a uno lógico, de la señal de un sensor:

```
sensor(206).Picture = LoadPicture(App.Path & "\sensor on.gif")
```

Este código muestra el círculo verde para el sensor 206. Cuando la señal cambia de uno a cero lógico, se muestra el círculo rojo mediante el siguiente código:

```
sensor(206).Picture = LoadPicture(App.Path & "\sensor off.gif")
```

2.3.8.8 Implementación de alarmas

La activación de algunos sensores implica una condición que no permite el correcto funcionamiento de la etapa (prelimpieza, primera limpieza, segunda limpieza o molienda) por lo que provocan la parada total o parcial de ésta. Estos sensores y los protectores térmicos se denominarán alarmas.

Es importante que cuando ocurra una alarma esta se visualice claramente. De este modo, el operador conocerá cual es el problema y podrá resolverlo. Cuando ocurre

una alarma aparecen en pantalla dos indicaciones. Uno es un mensaje de texto indicando al operador el estado de alarma y el código del elemento involucrado. La otra señal es gráfica. Si la señal es de un protector térmico, la imagen de cuadrado utilizada como interruptor del motor cambia de color a amarillo. La imagen que simboliza los sensores también cambia de color rojo a verde.

A continuación se muestra una imagen de la pantalla de molienda en la que aparece el mensaje de texto indicando la condición de alarma:

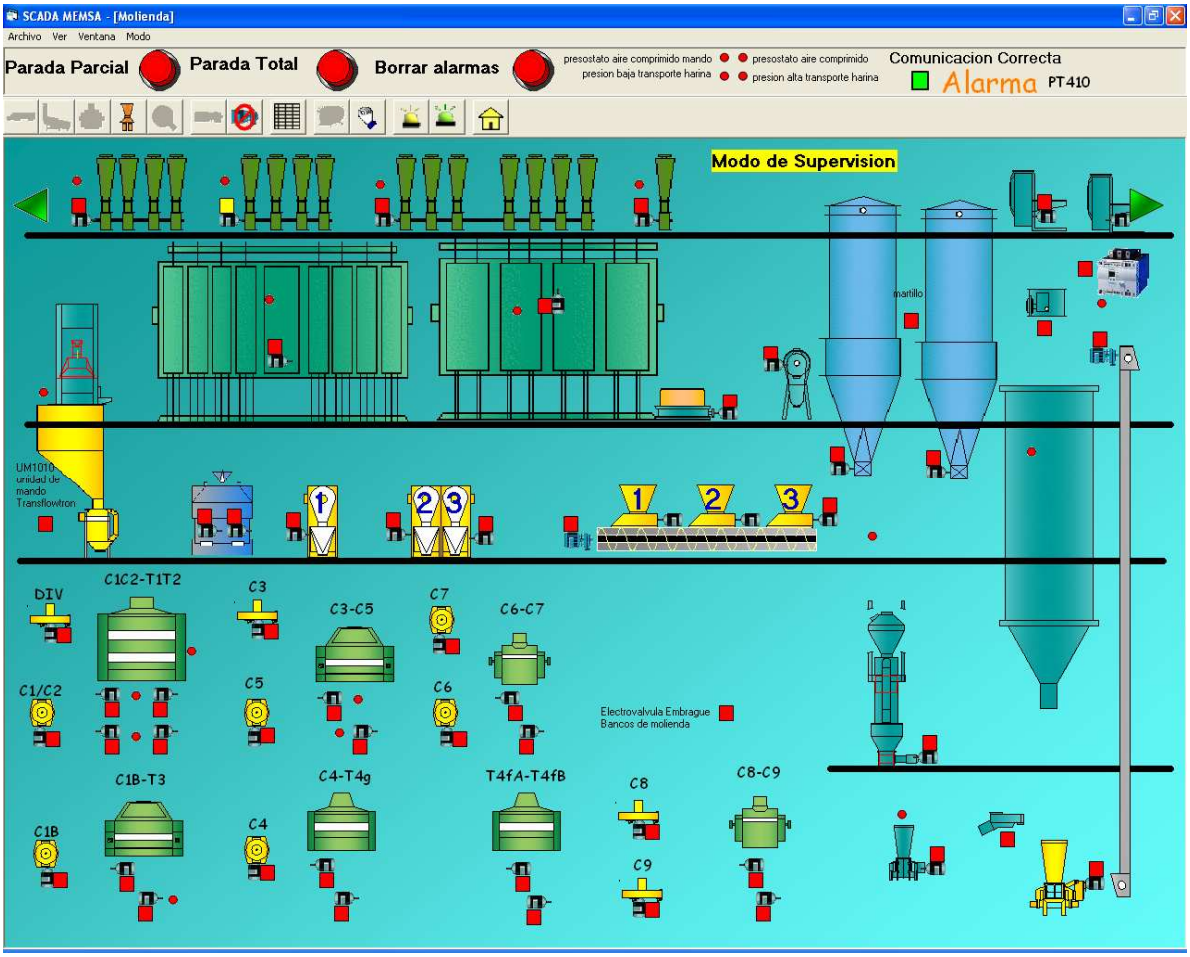


Figura 2.45. Pantalla de molienda indicando activación de alarma: PT410

Cuando se detecta una señal de alarma, lo primero que hace el software es registrar en la base de datos el código del elemento, la hora y el nombre del usuario encargado. Luego muestra las dos indicaciones gráficas y espera que el operador ingrese el motivo que la ocasionó antes de permitir arrancar nuevamente la etapa. Si no se ingresa el motivo, el programa impide realizar cualquier operación y muestra el siguiente mensaje:

Figura 2.46. Mensaje indicando que hay alarmas activadas

Al detectarse una alarma, el PLC coloca en uno lógico una bandera que indica cual protector térmico o sensor se activó. Si alguna de las banderas se encuentra en uno lógico, la lógica programada no permite el arranque de la etapa. Por lo tanto es indispensable borrar esta bandera desde Visual Basic para arrancar. Cuando se hace doble clic sobre la imagen del elemento que provocó la alarma, se borra la bandera utilizando el siguiente código:

```
v1414.Caption = v1414.Caption - 2 ^ 1
numero_opc(1) = v1414c.numero_opc
valor(1) = v1414.Caption
MDIForm1.OPCItemWrite 1, numero_opc(), valor()
```

El código mostrado arriba coloca en cero el bit uno de la localidad de memoria V1414. Cada una de las alarmas tiene una bandera en el PLC. Estas banderas se escriben en localidades de memoria de ocho bits. El programa de Visual Basic borra la bandera adecuada a la señal que se activó. Por ejemplo, para borrar una bandera

localizada en el bit seis de la localidad de memoria V1417 se utiliza el siguiente código:

```
v1417.Caption = v1417.Caption - 2 ^ 6
numero_opc(1) = v1417c.numero_opc
valor(1) = v1417.Caption
MDIForm1.OPCItemWrite 1, numero_opc(), valor()
```

Cuando se han borrado todas las alarmas se ejecuta una rutina en el PLC que cambia el mensaje del panel de operador ubicado en el tablero de control. Esta rutina se ejecuta al pulsar la tecla F5 del panel OP-1510. El código mostrado a continuación envía una señal desde el software del sistema SCADA para hacer correr la misma rutina:

```
numero_opc(1) = C40.numero_opc
valor(1) = 1
MDIForm1.OPCItemWrite 1, numero_opc(), valor()

numero_opc(1) = C40.numero_opc
valor(1) = 0
MDIForm1.OPCItemWrite 1, numero_opc(), valor()
```

Este código coloca en uno lógico y luego en cero el relé de control C40, el cual se encuentra en el programa del PLC, en paralelo con la tecla f5 del panel *Opti-Mate OP-1510*. Es decir, el código simula la acción de pulsar la tecla.

Luego, el programa muestra la pantalla diseñada para ingresar el motivo de la alarma. Para obligar al operador a escribir el motivo, esta pantalla permanece visible hasta que se coloque un mensaje de más de cinco caracteres y se pulse el botón colocado en la pantalla o la tecla *enter* del teclado. Cuando se cumplen estas

condiciones, el programa regresa a la pantalla en la cual se encontraba y se escribe en la base de datos el motivo ingresado por el operador.

Debido a la programación del PLC, para poder arrancar nuevamente los motores se debe pulsar la tecla f1 del panel OP-1510. Desde el software del sistema SCADA se simula esta acción mediante el siguiente código:

```
numero_opc(1) = C41.numero_opc
valor(1) = 1
MDIForm1.OPCItemWrite 1, numero_opc(), valor()
```

Para que el relé auxiliar trabaje de la misma manera que la tecla f1 del panel, su valor debe regresar a cero lógico. En el programa que gobierna al PLC se agrego una bobina reset que se activa al momento de pulsar la tecla f1 o de activar el relé C41. En la imagen mostrada a continuación se observa esta sección del código escalera.

Figura 2.47. Sección del código escalera en el que se ve la acción del relé de control C41

2.3.9 DESCRIPCIÓN DE LAS VENTANAS DEL PROGRAMA Y SUS FUNCIONES

En esta sección se describen las ventanas utilizadas en el software del sistema SCADA. Se describen las funciones de cada ventana y se explica el código empleado para ejecutarlas.

Se explica como se realiza el control de acceso al software, la creación y eliminación de usuarios, y las consultas de eventos y alarmas a la base de datos. En esta sección también se puede visualizar la apariencia de todas las pantallas. Se han colocado imágenes de cada pantalla para que se pueda apreciar su diseño.

2.3.9.1 Ventana de administración del programa

Esta ventana permite acceder al software e ir hacia el resto de las ventanas. Desde ésta se pueden seleccionar las opciones para eliminar y agregar usuarios. El icono para cambiar de contraseña también se encuentra en esta ventana.

Esta es la primera ventana que se muestra al iniciar el programa. Cuando se hace correr el software del sistema SCADA, en esta pantalla aparece un único botón, denominado *ingresar,* y las funciones del menú y de la barra de tareas que permiten acceder a otras ventanas se deshabilitan. Al hacer clic sobre el botón *ingresar*, aparece la ventana diseñada para ingresar el nombre de usuario y la clave.

Si se ingresa un nombre de usuario y una contraseña correcta, el programa muestra nuevamente la pantalla *administración del programa*, pero ahora, se muestran botones adicionales, mediante los cuales se puede acceder a otras pantallas. En la imagen mostrada a continuación se pueden observar estos botones.

A continuación se muestran dos imágenes de la ventana *administración del programa.* En la primera solo el botón *ingresar* está visible y en la segunda se muestra la ventana con todos los botones*:*

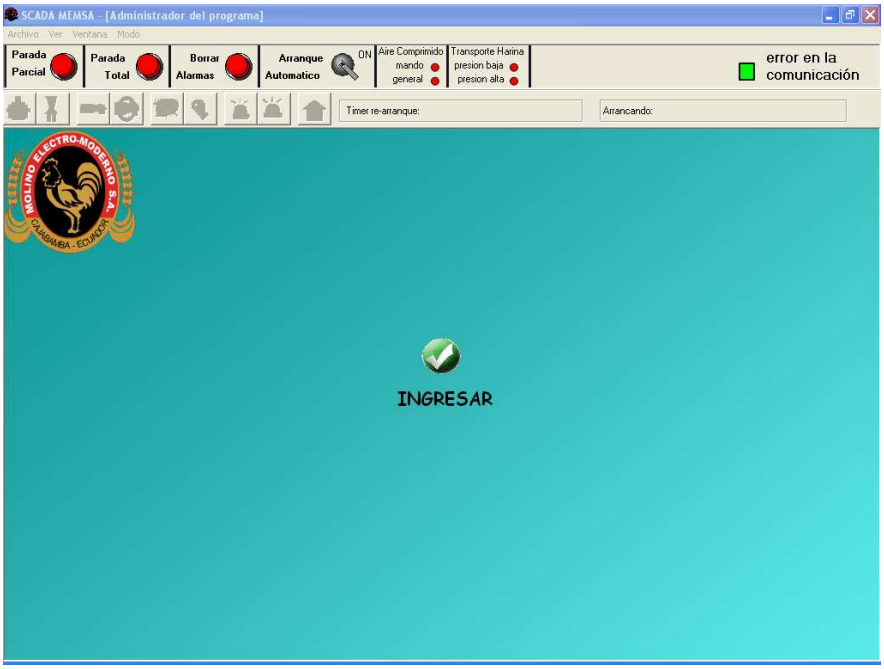


Figura 2.48. Ventana administración del programa con el botón ingresar

Figura 2.49. Ventana administración del programa con todos los botones

La imagen superior muestra la pantalla *administración del programa* con todos los botones visibles. La ventana tiene esta apariencia cuando se ingresa al software como administrador. Si se ingresa como operador, los botones agregar usuario y eliminar usuario no son visibles.

Cuando se pulsa cualquier botón, con excepción de *salir,* el programa muestra la ventana indicada por el texto del botón. Si se hace clic sobre el botón *salir*, se muestra la ventana *administración del programa* con el mismo formato que cuando se arranca por primera vez el software. Es decir, solo el botón *ingreso* es visible y para poder utilizar las funciones del programa es necesario ingresar nuevamente utilizando un nombre de usuario y una clave válidos.

2.3.9.2 Ventana para el ingreso del nombre de usuario y clave

Esta ventana permite realizar el control del acceso de los usuarios. Aquí se deben ingresar los datos del operador. El código de esta pantalla verifica la información ingresada y si es correcta permite el acceso a las funciones del software. En esta pantalla también se define si la persona que ingresó puede utilizar todas las funciones o tiene un acceso restringido.

La ventana *ingreso de clave* consta de dos cuadros de texto y dos botones. Los cuadros de texto se utilizan para que el operador ingrese su nombre de usuario y la clave, respectivamente. Al pulsar el botón *ingresar*, se ejecuta la rutina para la comprobación de los datos, mientras que al pulsar sobre *cancelar*, el programa regresa a la pantalla *administración del programa* sin realizar ninguna operación.

Para realizar la comprobación de la información se pregunta a la base de datos la clave almacena para el nombre de usuario ingresado. El valor leído de la base de datos se compara con el ingresado, si estos son iguales se permite el acceso al

operador. La información se lee de la base de datos utilizando el siguiente comando SQL:

SqlQry = "SELECT clave,nombre_acceso FROM usuarios where usuario_id='" & Trim$(nombre_de_usuario) & "'"

Además de la clave, este comando también lee el nombre del acceso. De esta manera se puede restringir o permitir el uso de las funciones del software. Solo si el acceso es de tipo *administrador,* los botones *agregar usuario* y *eliminar usuario* son visibles.

Cuando se pulsa el botón *ingresar,* y el nombre de usuario y la clave son correctos, se muestra un mensaje indicando que se ha ingresado al sistema y cual es el tipo de acceso para el usuario. La imagen de abajo muestra el mensaje para un usuario de tipo administrador.

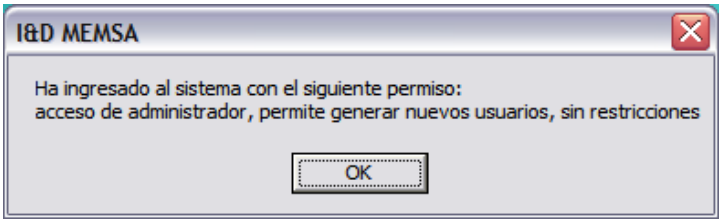


Figura 2.50. Mensaje indicando que se ha ingresado al sistema con acceso de administrador

En cambio, si la clave es incorrecta el mensaje que se muestra es el siguiente:

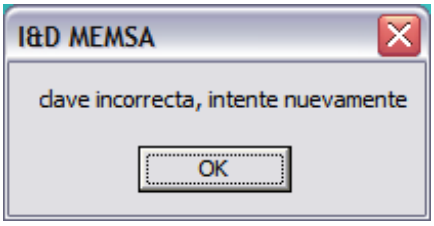


Figura 2.51. Mensaje indicando que la clave ingresada es incorrecta

Si el dato incorrecto es el nombre de usuario, el texto del mensaje dice: *Nombre de usuario incorrecto, intente nuevamente.*

A continuación se muestra una imagen de esta pantalla:

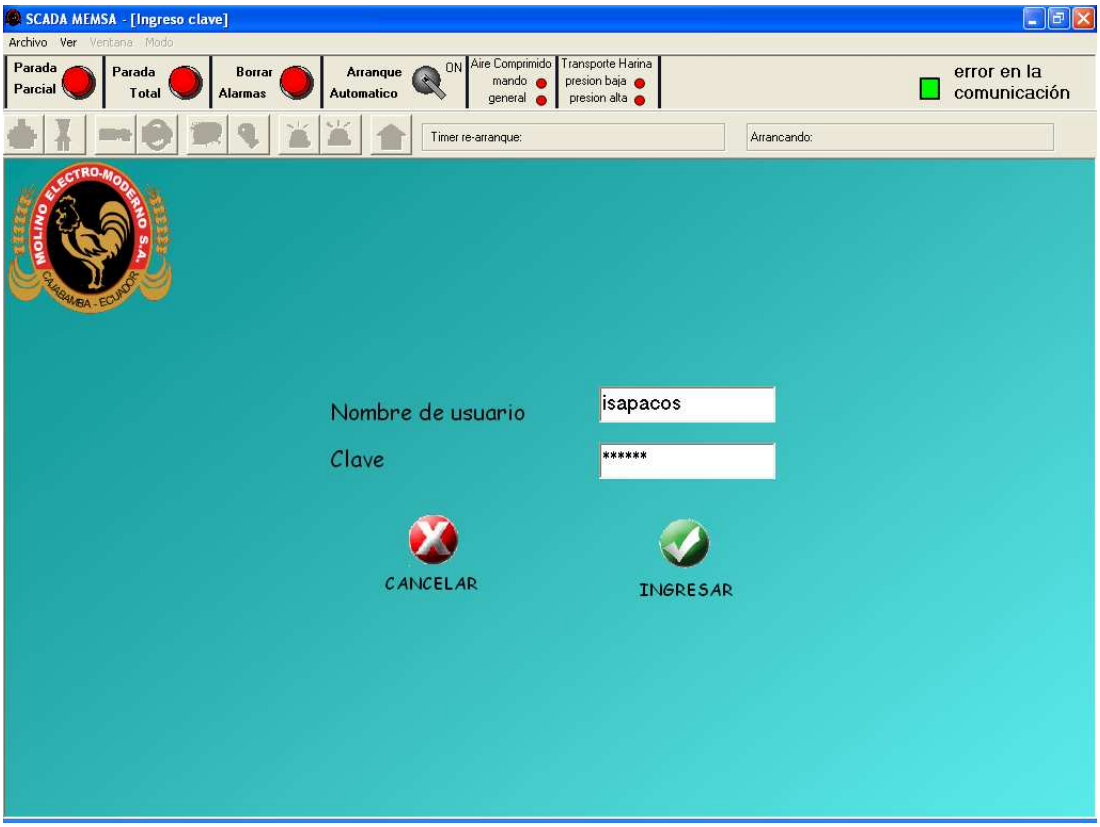


Figura 2.52. Ventana ingreso de clave

2.3.9.3 Ventana para agregar usuarios

A esta ventana solo pueden acceder los usuarios con categoría de administrador. Como su nombre lo indica, se utiliza para agregar nuevos usuarios al sistema. Utilizando esta ventana se puede permitir a nuevos operadores el acceso al software del sistema SCADA.

Esta ventana consta de 5 cuadros de texto y una lista en los que se debe ingresar la siguiente información del nuevo usuario: nombre, apellido, nombre de usuario, contraseña y tipo de acceso. La contraseña se ingresa dos veces para verificar que no hubo equivocación al escribirla.

Cuando se presiona el botón *ingresar* datos, se ejecuta una rutina en la cual se verifica que se haya ingresado toda la información y que el nombre de usuario no se encuentre registrado en la base de datos. Si alguno de los casilleros está en blanco, las dos claves ingresadas son diferentes o el nombre de usuario ya existe, se muestra un mensaje alertando al usuario de cual de estos problemas aconteció.

El botón cancelar permite regresar a la ventana *administrador del programa* sin realizar ninguna operación.

Una vez que toda la información está correcta, ésta se escribe en la base de datos mediante el siguiente comando SQL:

```
SqlQry = "INSERT INTO usuarios(nombre,apellido,usuario_id,clave,nombre_acceso) VALUES ('" & Trim$(nombre) & "' , '" & Trim$(apellido) & "' , '" & Trim$(nombre_de_usuario) & "' , '" & Trim$(clave) & "' , '" & Trim$(nombre_acceso) & "')"
```

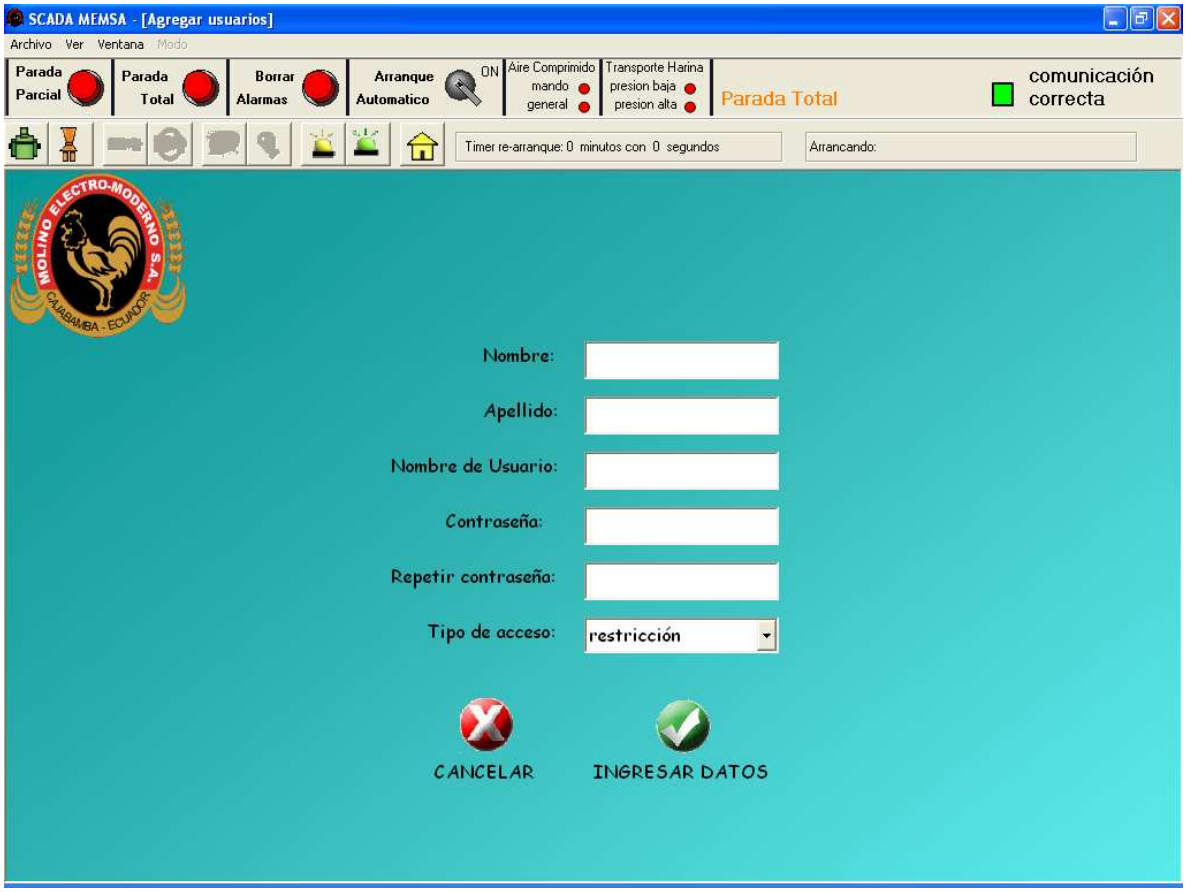


Figura 2.53. Ventana agregar usuarios

2.3.9.4 Ventana para eliminar usuarios

Para utilizar esta ventana también es necesario tener acceso de administrador. Desde aquí se eliminan usuarios de modo que no puedan ingresar al software ni utilizar sus funciones.

Al cargar esta ventana, se muestra una lista de todos los usuarios registrados en la base de datos. Para eliminar alguno de ellos, se lo debe seleccionar y hacer clic en el botón *eliminar.* Si se pulsa el botón *cancelar,* el programa regresa a la ventana *administrador de programa*, sin modificar la lista de usuarios.

Al pulsar el botón eliminar, aparece un mensaje preguntando al operador si está seguro de ejecutar la operación. El cuadro de mensaje posee dos botones mediante los cuales se ingresa la decisión. A continuación se muestra la imagen de este cuadro de mensaje antes de eliminar al usuario isapacos:

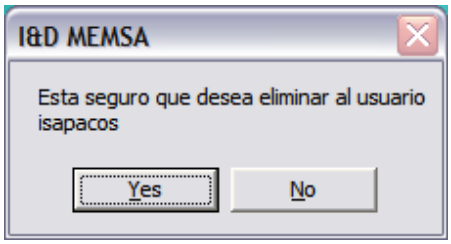


Figura 2.54. Mensaje preguntado si el operador está seguro de eliminar un usuario

Para poder ingresar al software del sistema SCADA, es indispensable que exista por lo menos un usuario registrado. Por esta razón, si en la base de datos, solo existen los datos de un único usuario, el programa no permite que se lo elimine y muestra el siguiente mensaje:

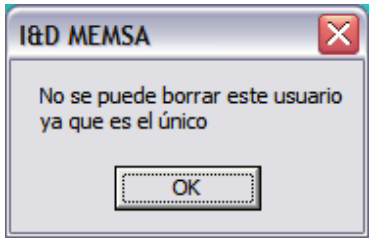


Figura 2.55. Mensaje indicando que no se puede eliminar al usuario por ser el único registrado

El código encargado de eliminar al usuario es:

```
SqlQry = "delete from usuarios where usuario_id= '" & Trim$(usuario_id) & "'"
```

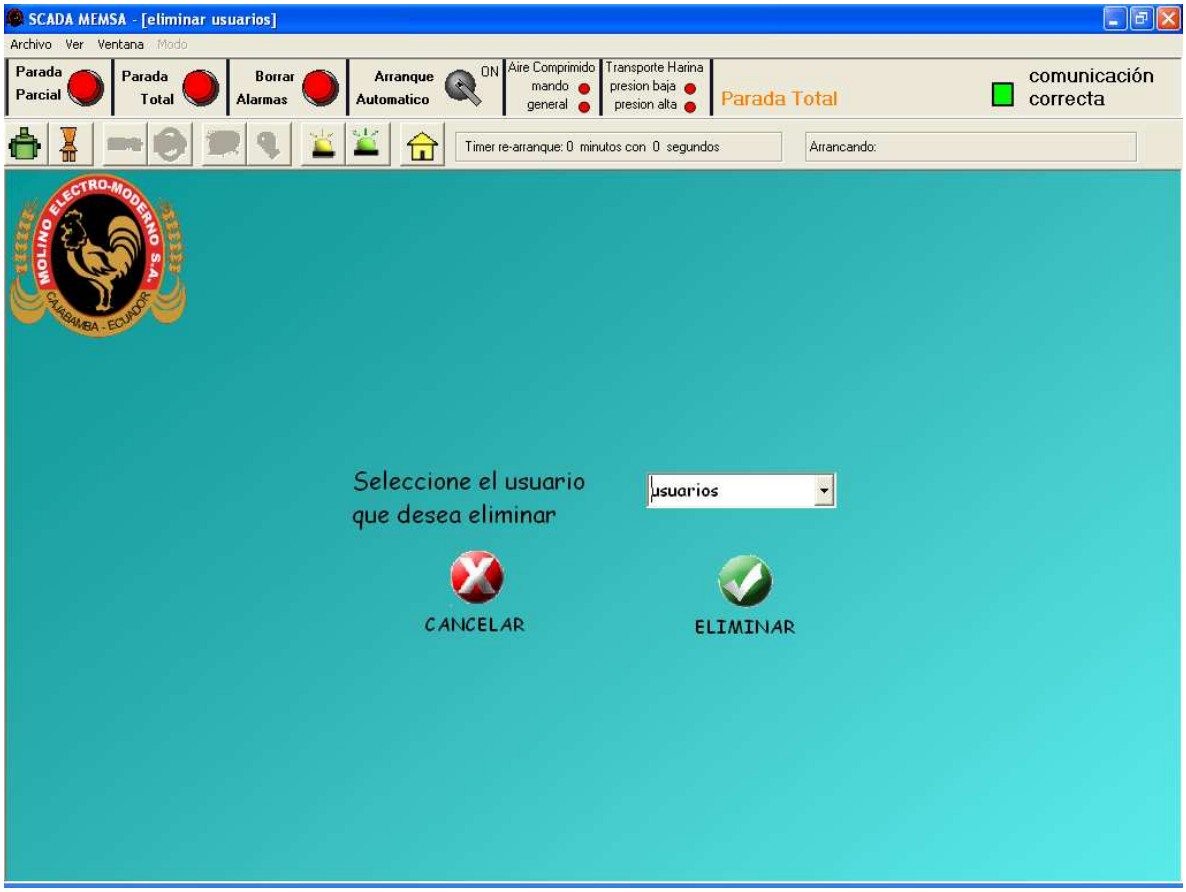


Figura 2.56. Ventana eliminar usuarios

2.3.9.5 Ventana para cambio de contraseña

Por razones de seguridad, los usuarios deben poder cambiar su contraseña cuando les parezca conveniente. La operación de cambio de contraseña se realiza en esta ventana.

En la ventana hay unos casilleros de texto en los cuales se debe ingresar: la clave actual, la clave nueva, y nuevamente la clave nueva para verificar que se ingresó correctamente. Al hacer clic sobre el botón *aceptar,* los caracteres utilizados como clave se escriben en la base de datos. Si se pulsa el botón *cancelar* el programa regresa a la ventana *administración de programa* sin realizar ninguna operación.

Si la clave actual ingresada no es correcta se muestra un mensaje indicándole al usuario para que la ingrese nuevamente. Del mismo modo, si las dos claves nuevas escritas no concuerdan se muestra el siguiente mensaje:

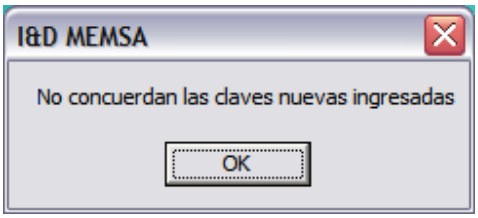


Figura 2.57. Mensaje indicando que las claves ingresadas no son iguales

Cuando toda la información es correcta, se escribe la nueva clave en la base de datos utilizando el siguiente código SQL:

SqlQry = "UPDATE usuarios SET clave= '" & Trim$(clave_nueva) & "' where usuario_id = '" & Trim$(nombre_de_usuario) & "'"

Una vez que el cambio de clave se ha realizado exitosamente, se confirma la operación con el siguiente mensaje:

Figura 2.58. Mensaje indicando que se realizó el cambio de clave

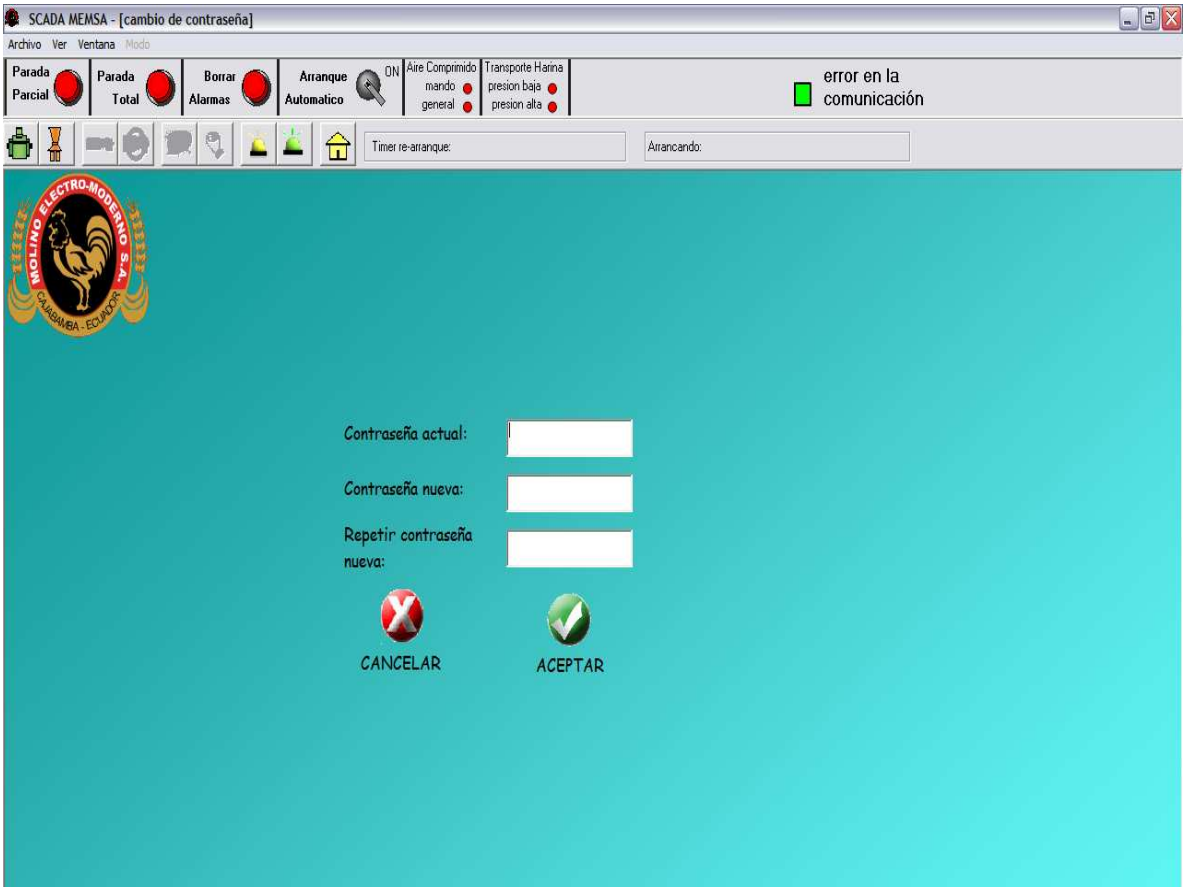


Figura 2.59. Ventana cambio de contraseña

2.3.9.6 Ventanas de operación

Las ventanas de operación son las más importantes del software del sistema SCADA. Desde éstas se controla y supervisa el proceso de producción de harina. En las aplicaciones se crearon: dos ventanas de operación para la sección limpieza y dos ventanas de operación para sección molienda.

En estas ventanas se muestran la maquinaria, los motores, los sensores y demás elementos utilizados en el proceso de molienda de trigo. Mediante los botones e indicadores ubicados en las ventanas podemos: encender y apagar manualmente los

motores, realizar el arranque automático de alguna de las etapas, realizar paradas de emergencia, y revisar información de los motores, actuadores y sensores.

A continuación se muestran las imágenes de estas pantallas:

2.3.9.6.1 Ventanas de la etapa de limpieza

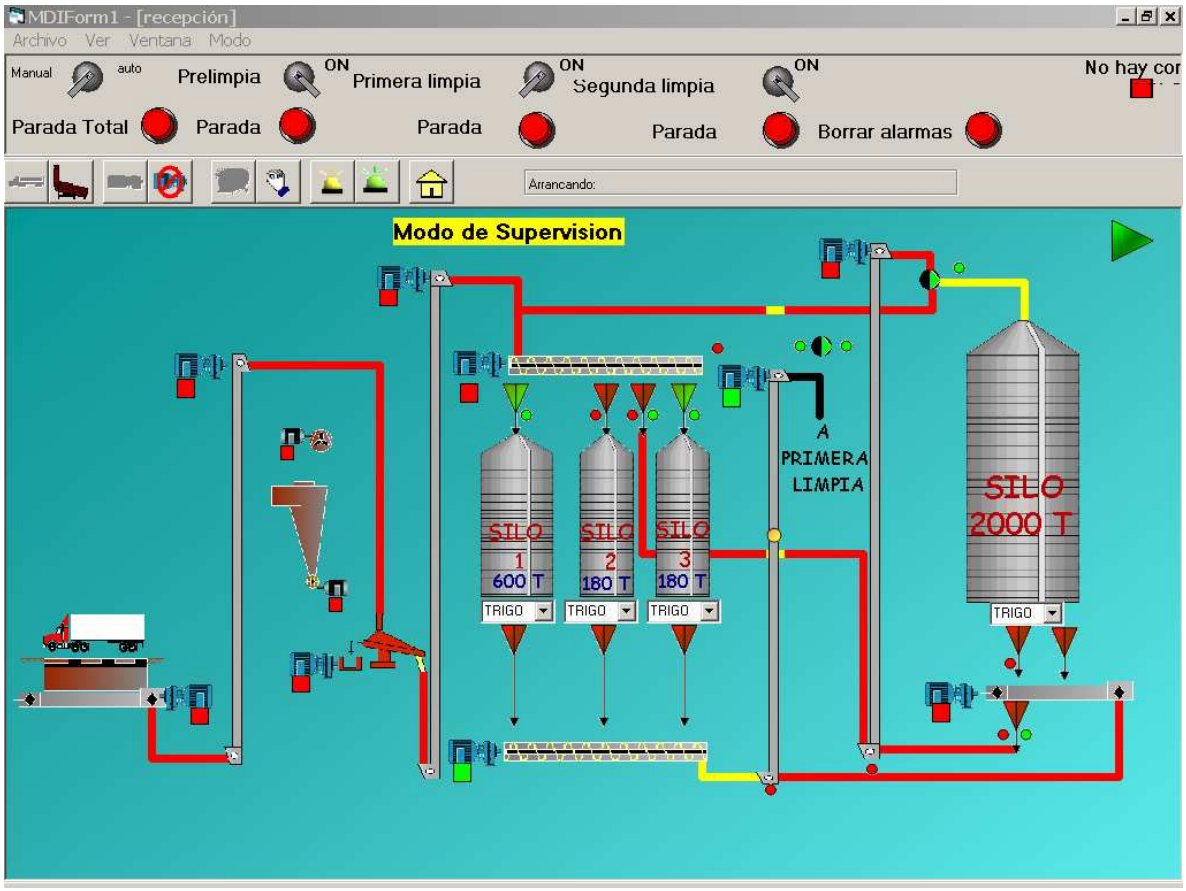


Figura 2.60. Ventana para etapa de recepción, prelimpieza, almacenaje y trasilaje

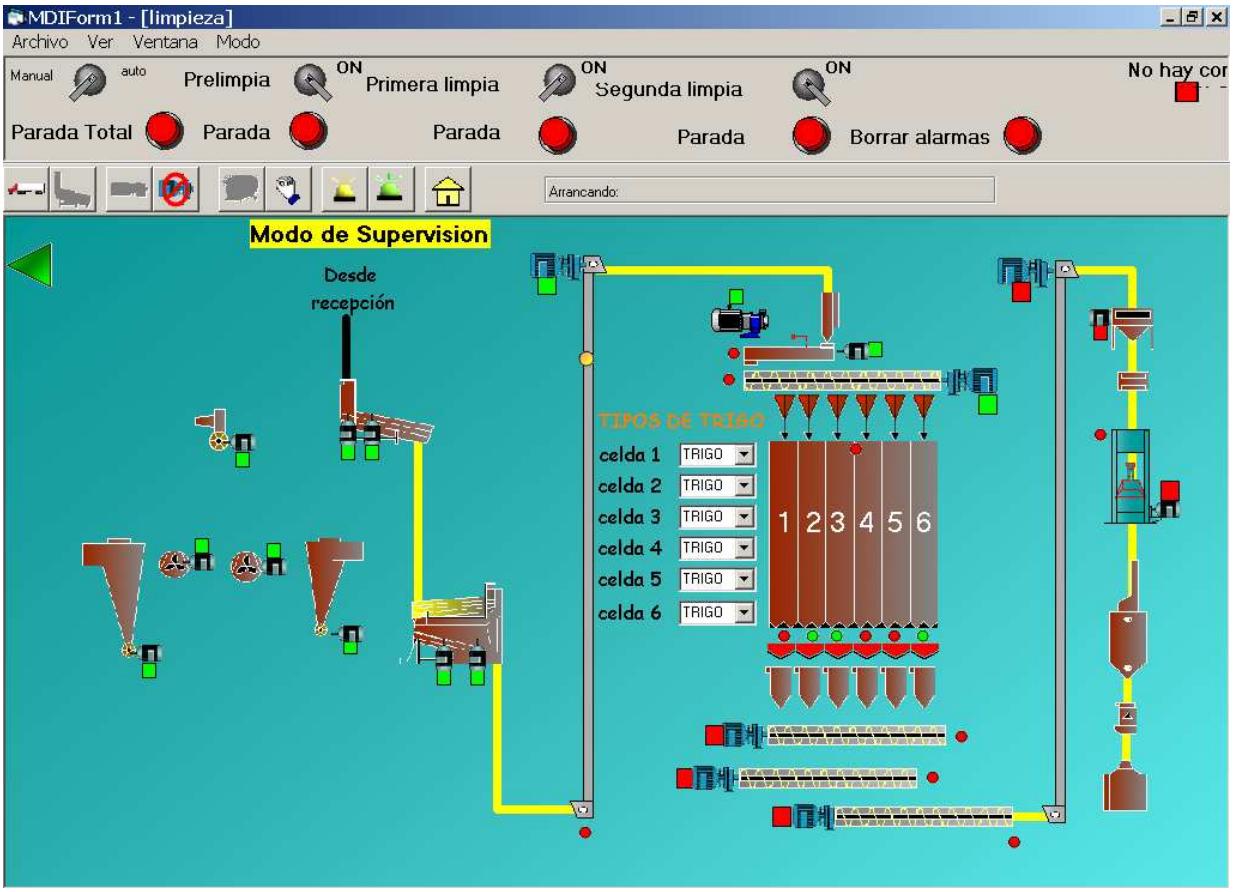


Figura 2.61. Ventana para etapa de primera y segunda limpieza

2.3.9.6.2 Ventanas de la etapa de molienda

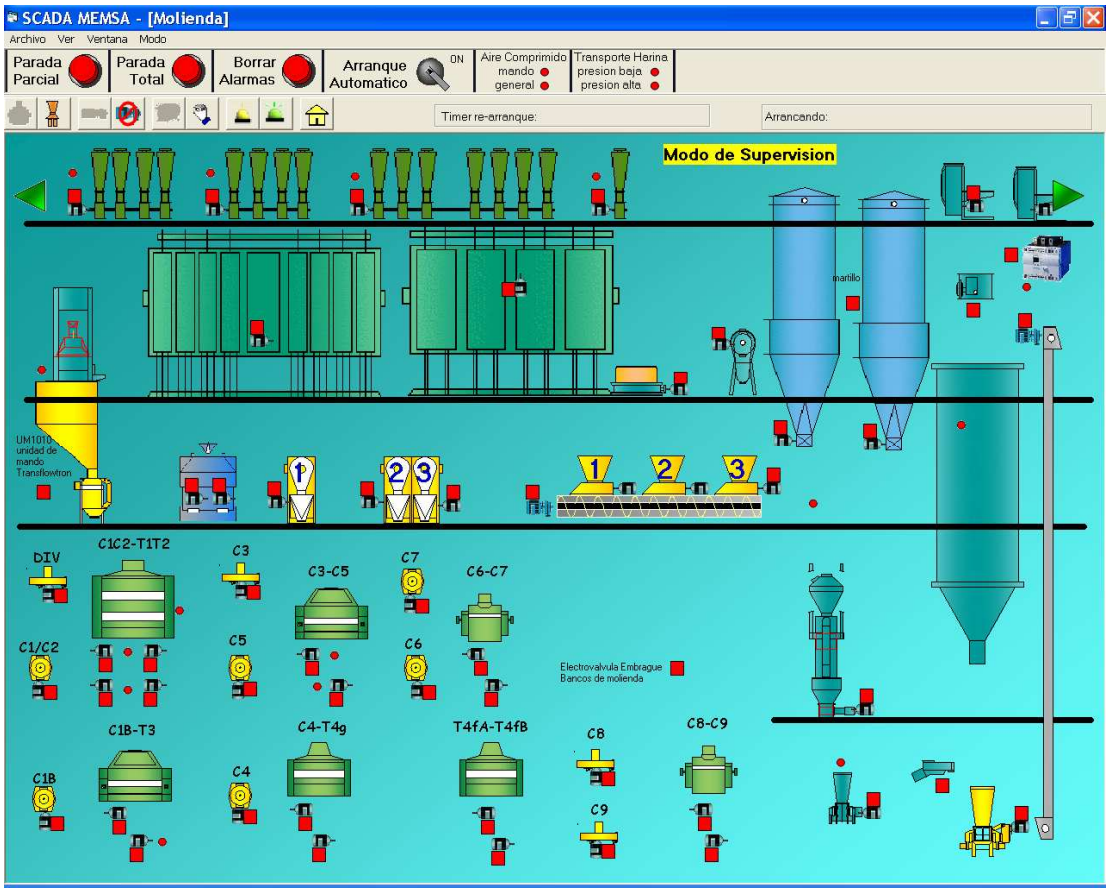


Figura 2.62. Ventana para etapa de molienda

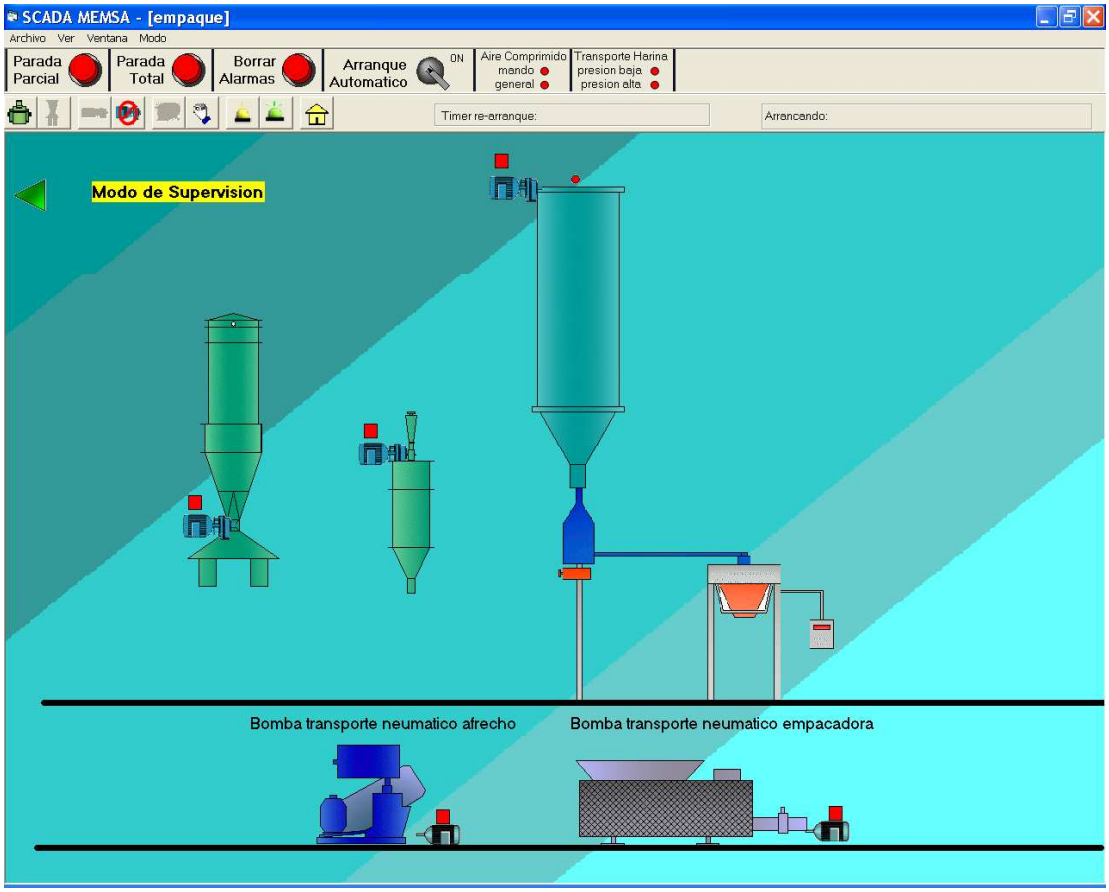


Figura 2.63. Ventana para la etapa de ensacado

2.3.9.7 Ventana para ingresar el motivo que ocasionó una alarma

Cuando ocurre una alarma, el operador debe ingresar el motivo que la causó. Esta información se almacena en la base de datos para poder consultarla posteriormente. De este modo, se lleva un registro de los motivos que ocasionaron las alarmas. Esta ventana permite que el operador ingrese la causa de la alarma de una manera fácil y rápida.

En esta ventana se encuentran un cuadro de texto y un botón. El primero se utiliza para escribir el motivo que ocasionó una alarma y el segundo para guardar esta información en la base de datos. Cuando se pulsa sobre el botón *continuar*, o se

pulsa la tecla *enter* del teclado, se ejecuta una rutina que verifica que el motivo ingresado posea por lo menos 5 caracteres y máximo 400. Si el cuadro de texto está en blanco o tiene menos caracteres de los requeridos, aparece el siguiente mensaje:

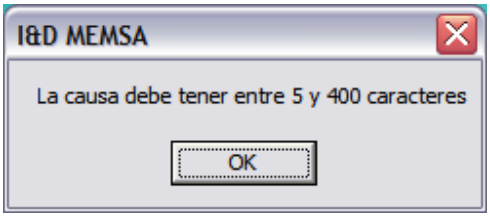


Figura 2.64. Mensaje mostrando que el motivo ingresado no es del tamaño requerido

Si no se ingresa el motivo, el programa no permite regresar a otra ventana, impidiendo de esta manera, encender nuevamente la maquinaria apagada debido a la activación de la alarma. De este modo se obliga al operador a ingresar el motivo que ocasionó dicha alarma.

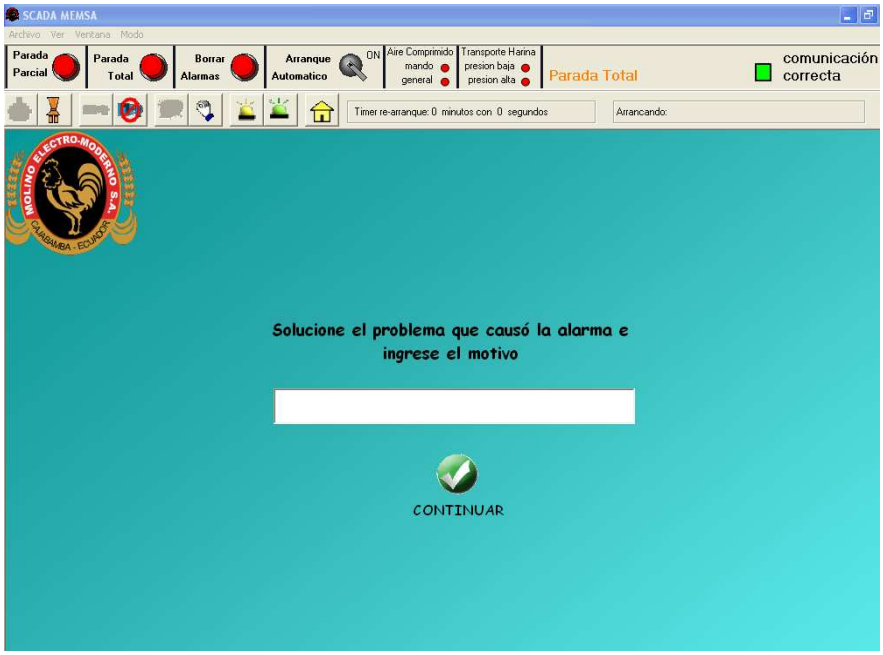


Figura 2.65. Ventana ingreso de motivo

2.3.9.8 Ventana para mostrar el registro de los eventos ocurridos

Uno de los objetivos para la implementación de este sistema es obtener un registro histórico de los eventos ocurridos. De este modo se puede determinar la fecha y hora del encendido y del apagado de toda la maquinaria de la fábrica. Desde esta ventana se puede obtener esta información y enviarla a Microsoft Excel con el objetivo de crear reportes.

En esta ventana, se muestra en una tabla, la información de los eventos ocurridos. Se muestra el código del evento, el código del elemento, una descripción del elemento, la hora de encendido, si se encendió de modo manual o automático, la hora de apagado, el motivo del apagado y el intervalo de duración. El motivo de apagado puede ser: parada parcial, parada total o apagado manual.

Para visualizar los datos se debe pulsar el botón *mostrar eventos ocurridos*. La información se obtiene de la base de datos empleando el siguiente comando SQL:

SqlQry = "SELECT evento_id, elemento_co, descripcion, hora_inicio, causa_inicio, hora_fin,causa_fin,intervalo FROM eventos,elementos_plc WHERE elementos_plc.elemento_id=eventos.elemento_id order by evento_id;"

El código mostrado en la parte superior lee la información de todos los eventos ocurridos. La cantidad de registros de la tabla eventos puede ser muy numerosa, por lo tanto se ha implemento un mecanismo que ayude al operador a encontrar la información deseada.

Para disminuir la información mostrada se puede realizar la búsqueda utilizando el código del elemento y las la fechas de inicio y finalización entre las que sucedió el evento.

Si se desean ver todos los eventos ocurridos entre dos fechas se ingresa la fecha de inicio y la fecha final, sin seleccionar un elemento de la lista de códigos. El comando SQL que se utiliza en esta situación es:

SqlQry = "SELECT evento_id,elemento_co,descripcion,hora_inicio, causa_inicio,hora_fin,causa_fin,intervalo FROM eventos,elementos_plc WHERE elementos_plc.elemento_id=eventos.elemento_id AND hora_inicio< cast('" & MonthView2.Value & "' as date) AND hora_inicio> cast('" & MonthView1.Value & "' as date) order by evento_id;"

Cuando se desea observar los eventos de un elemento ocurridos entre dos fechas se emplea el siguiente comando SQL:

SqlQry = "SELECT evento_id, elemento_co, descripcion, hora_inicio, causa_inicio, hora_fin, causa_fin, intervalo FROM eventos, elementos_plc WHERE elementos_plc.elemento_id=eventos.elemento_id AND eventos.elemento_id='" & elemento_id & "' AND hora_inicio< cast('" & MonthView2.Value & "' as date) AND hora_inicio> cast('" & MonthView1.Value & "' as date) order by evento_id;"

Para obtener la información de la base de datos se debe hacer clic sobre el botón buscar el cual se encuentra ubicado en la parte inferior izquierda de la pantalla. La imagen del botón es la siguiente:

Figura 2.66. Botón buscar

Al lado izquierdo del botón buscar encontramos el botón para exportar datos a Excel. Si se hace clic sobre él, la información mostrada en la tabla de la pantalla es transferida a una hoja de datos en Microsoft Excel. De esta manera se pude editar,

mejorar la presentación e imprimir los registros. A continuación se muestra la imagen de este botón:

Figura 2.67. Botón utilizado para exportar datos a Excel

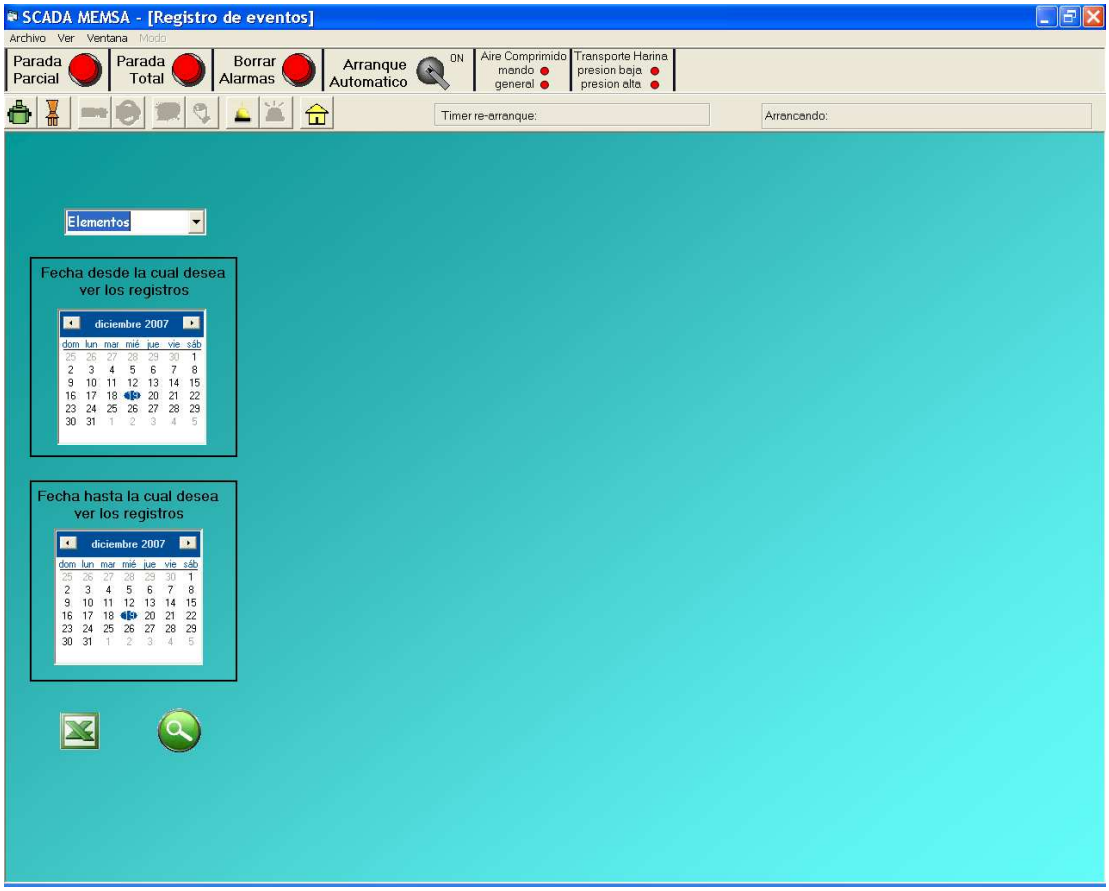


Figura 2.68. Ventana eventos

2.3.9.9 Ventana para mostrar el registro de las alarmas ocurridas

Desde esta ventana se puede solicitar información sobre las alarmas ocurridas. Los datos históricos de alarmas permiten conocer la fecha y la hora en la que éstas sucedieron. A partir de estos datos también se puede determinar si la activación de la alarma ocasionó una parada parcial o total. Otro dato proveniente de la base de datos es el nombre del usuario que se encontraba a cargo de la producción el momento que la alarma ocurrió.

Como se puede observar en la figura *Ventana Alarmas*, la apariencia de esta ventana es muy parecida a la de la ventana *eventos*. En ambas se ha implementando la función de búsqueda y trabaja de la misma manera en las dos ventanas. La diferencia entre las ventanas radica en la información que se muestra en las tablas.

La información que se muestra en esta ventana es: identificación de la alarma, código del elemento, descripción del elemento, fecha en la que ocurrió, motivo, consecuencia de la alarma y nombre del usuario a cargo. En la columna consecuencia de la alarma se indica si la activación de la alarma ocasionó una parada parcial o total. Esta información también se puede enviar a una hoja de datos pulsando sobre el botón: *exportar datos a Excel.*

Los comandos SQL empleados para leer la información de la base de datos también son muy parecidos a los utilizados en la ventana descrita en la sección anterior. La diferencia está en la información que se solicita y la tabla de donde provienen estos datos. Por ejemplo, para solicitar la información de todas alarmas registradas entre dos fechas se utiliza el siguiente comando:

SqlQry = "SELECT alarma_id,elemento_co,hora,causa,usuario_id FROM alarmas,elementos_plc WHERE elementos_plc.elemento_id=alarmas.elemento_id AND hora< cast('" & MonthView2.Value & "' as date) AND hora> cast('" & MonthView1.Value & "' as date) order by alarma_id;"

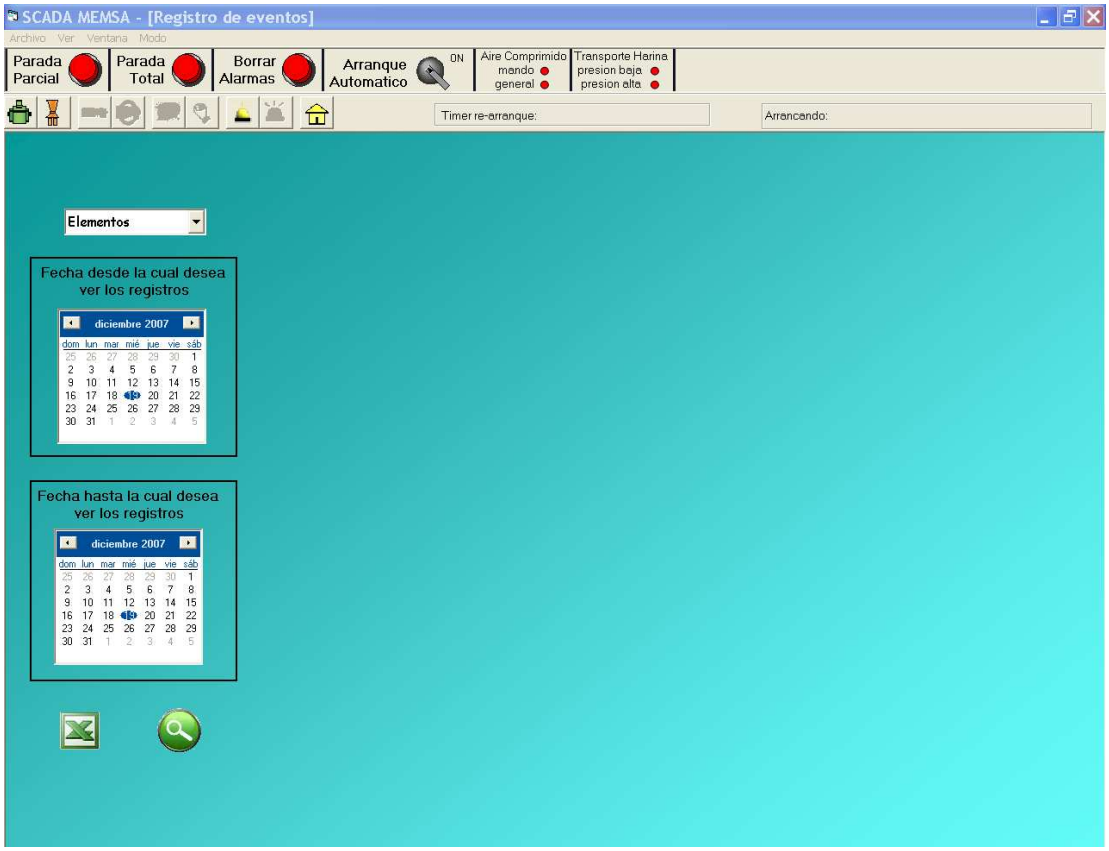


Figura 2.69. Ventana alarmas

2.3.10 IMPLEMENTACIÓN DE OTRAS FUNCIONES

En el software del sistema SCADA se han implementado otras funciones que todavía no se han analizado. En esta sección se las describe y se indica como se implementaron.

A continuación se encuentran: el método utilizado para ingresar al modo de control, como se realiza el registro de los eventos y de las alarmas, como funciona el detector de comunicación, como se realiza el arranque y parada de la producción desde el software del sistema SCADA, y otras funciones importantes para la operación del sistema.

2.3.10.1 Código utilizado para ingresar a modo de control

Es importante que el control de la planta se realice únicamente desde un sistema, ya sea desde el SCADA o desde el panel de control. Si se envían órdenes al mismo tiempo desde dos lugares diferentes se provocarían conflictos. Podría darse el caso que desde el software se envía la orden para realizar el arranque automático de una sección, mientras los selectores del tablero de control ordenan un arranque manual. Para evitar estos inconvenientes, cuando se selecciona el modo de control en el software del sistema SCADA se deben deshabilitar las señales provenientes del panel de control.

Cuando se ingresa al modo de control se activa un relé de control en el programa del PLC. Este relé permite realizar operaciones de arranque y parada automática desde el computador deshabilitando las señales provenientes de los selectores y pulsadores del tablero de control. Además, cuando se activa este contacto, se ejecuta una rutina en el PLC que muestra en el panel *Opti-Mate OP-1510* un mensaje indicando que el control se está realizando desde el software del SCADA.

El código que activa este relé es:

```
numero_opc(1) = C20.numero_opc
valor(1) = 1
OPCItemWrite 1, numero_opc(), valor()
```

Cuando se pasa a modo de supervisión se desactiva el relé utilizando el código mostrado a continuación:

```
numero_opc(1) = C20.numero_opc
valor(1) = 0
OPCItemWrite 1, numero_opc(), valor()
```

Los programas que gobiernan los PLCs se cambiaron de modo que al activar C20 se deshabilitan las entradas del panel de operador y permite la ejecución de órdenes desde el programa desarrollado en Visual Basic. A continuación se muestra un ejemplo de la modificación hecha. Este mismo cambio se efectuó para todos los selectores y pulsadores de los dos paneles de control.

Figura 2.70. Modificación realizada a diagrama escalera para deshabilitar selectores del tablero de control

2.3.10.2 Implementación del registro de eventos y alarmas en la base de datos

En secciones anteriores se ha descrito la importancia de tener datos históricos de los eventos y de las alarmas ocurridas. En esta sección se describe el método utilizado para registrar esta información en la base datos. Se muestran los comandos SQL utilizados para registrar los eventos y para registrar las alarmas

En la base de datos se registran los eventos y las alarmas ocurridas utilizando dos tablas. Siempre que se activa o desactiva un actuador, ya sea por medio del panel de operador, desde el software del SCADA o por medio de la lógica del PLC, se registra en la base de datos la información de esta operación en la tabla *Eventos*. Cuando ocurre una alarma, también se registra en la base de datos información que identifica este acontecimiento. Estos datos se guardan en la tabla *Alarmas.*

A continuación se describe el método utilizado para realizar los registros.

2.3.10.2.1 Registro de eventos

Cuando se detecta el cambio de estado de una salida del PLC se ejecuta una rutina en el programa desarrollado en Visual Basic. Se determina si el nuevo valor de la salida es cero o uno lógico y según esto se procede a escribir en la base de datos la información de encendido o apagado del actuador.

El código SQL empleado para escribir en la base de datos la información de encendido del motor es el siguiente:

"Insert into eventos(elemento_id,hora_inicio) values ((select elemento_id from elementos_plc where numero_vb ='" & Index & "'),current_timestamp)"

Este código se puede desglosar en dos partes. La primera (select elemento_id from elementos_plc where numero_vb ='" & Index & "') obtiene el valor de elemento_id del actuador que se encendió. El resto del código escribe este valor en la columna elemento_id de la tabla eventos, y la hora actual en la columna hora_inicio de la misma tabla.

Para poder escribir la hora de apagado en el casillero correcto, es necesario conocer en que fila se escribió la información de encendido. Para lo cual se utiliza el siguiente código SQL:

"select count(*) from eventos"

Este código se ejecuta justo después de escribir la hora de encendido del elemento. De esta manera obtenemos el número de fila en la cual debemos escribir la hora de apagado cuando este evento suceda. Este valor se guarda en un arreglo. Cada elemento tiene un casillero definido en el arreglo, de tal manera, al momento del apagado de dicho elemento, se conoce la fila en la cual se debe escribir la información. El siguiente código realiza esta operación:

evento_id(Index) = MyRecSet.Fields(0)

Cuando se apaga el elemento se escribe la hora de finalización del evento mediante el siguiente comando SQL:

"update eventos set hora_fin= current_timestamp where evento_id='" & evento_id(Index) & "'"

Este código escribe la hora actual en la columna hora_fin de la tabla eventos. Esta información se escribe en la fila cuyo número se encuentra almacenado en evento_id(Index). A continuación, se calcula el tiempo que estuvo encendido el motor, realizando una resta entre la hora de finalización del evento y la hora de inicio del evento, y esta información se guarda en la columna intervalo. El código mostrado a continuación realiza esta operación.

"update eventos set intervalo= ((select hora_fin from eventos where evento_id='" & evento_id(Index) & "')-(select hora_inicio from eventos where evento_id='" & evento_id(Index) & "')) where evento_id='" & evento_id(Index) & "'"

2.3.10.2.2 Registro de alarmas

Cuando se detecta una alarma, inmediatamente se guarda en la base de datos la información sobre este suceso. Se escribe la identificación del elemento involucrado, la hora en la que ocurre la alarma y el nombre del usuario a cargo de la operación del sistema SCADA. El código que escribe esta información es el siguiente:

"Insert into alarmas(elemento_id,hora,usuario_id) values ((select elemento_id from elementos_plc where numero_vb ='" & codigo & "'),current_timestamp,'" & Form2.Text1.Text & "')"

Luego de solucionar el inconveniente, el operador debe hacer doble clic sobre la indicación de la alarma en el software del SCADA. En ese momento aparece una pantalla en la cual se debe introducir el motivo que ocasionó la alarma. Esta información se guarda en la base de datos mediante el siguiente código:

"update alarmas set causa= '" & causa.Caption & "' where alarma_id='" & alarma_id(index_amarillo) & "'"

En las alarmas se sigue el mismo procedimiento que en los eventos para determinar la fila correcta en la cual se escribe. Cuando ocurre la alarma se guarda en otro arreglo el número de la fila y luego se utiliza este valor para escribir la causa de la alarma. El nombre del arreglo utilizado es alarma_id.

2.3.10.3 Exportación de información a Microsoft Excel

En el software del sistema SCADA se implementó una función que permite exportar la información a una hoja de cálculo de Microsoft Excel. Los datos mostrados en las ventanas *Alarmas y Eventos* se pueden enviara hacia este programa. Desde ahí se puede manejar la información extraída de la base de datos de una manera simple. Una vez que la información se encuentra en este programa se pueden realizar reportes y utilizar todas las funciones de este software. Además, desde Excel se pueden imprimir los registros de los eventos y de las alarmas.

En las ventanas *Alarmas* y *Eventos* se colocó un icono con el símbolo de Excel. Cuando se pulsa este icono se ejecuta la rutina para exportar los datos. Primero se verifica si existen datos listos para exportar. Luego, si los hay, se procede a abrir el programa Microsoft Excel y a enviar la información obtenida de la base de datos. En la hoja de datos se visualizan los mismos datos mostrados en las pantallas *Eventos* o *Alarmas* del programa del sistema SCADA

A continuación se encuentra el código utilizado para exportar los datos:

```
Dim prgExcel As Object
If datos_listos = True Then
datos_listos = False
llamar_datos (SqlQry)
Set prgExcel = CreateObject("Excel.application")
   prgExcel.Workbooks.Add
   prgExcel.Sheets(1).Name = "Alarmas"
   Dim i As Integer
   Dim j As Integer
   j = 1
   Do While Not MyRecSet.EOF
      For i = 0 To MyRecSet.Fields.Count - 1
         prgExcel.Cells(j + 1, i + 1) = MyRecSet.Fields(i).Value
      Next
      DoEvents
      MyRecSet.MoveNext
      j = j + 1
   Loop
   For i = 1 To MyRecSet.Fields.Count
      prgExcel.Cells(1, i) = MyRecSet.Fields(i - 1).Name
      prgExcel.Cells(1, i).EntireColumn.AutoFit
   Next
   prgExcel.Cells(1, 1).Select
   prgExcel.Selection.EntireRow.Insert
   prgExcel.Visible = True

Else
MsgBox "No hay datos para exportar", vbOKOnly, "I&D MEMSA"
End If
```

2.3.10.4 Implementación de cuadro de información

En las pantallas de operación del software del sistema SCADA se puede visualizar mucha información de la planta de producción. Aparte del estado de los actuadores y sensores, se puede obtener información sobre la maquinaria. Por ejemplo, se puede visualizar el código de cada motor, una breve descripción del mismo y la cantidad de tiempo que ha trabajado.

El cuadro de información se utiliza para mostrar datos importantes de los motores, sensores y demás elementos. Para los actuadores, desde la base de datos se lee el código del elemento, la descripción de éste y se calcula el tiempo de funcionamiento. En el caso de los sensores se lee el código, la máquina a la cual está asociado y el motivo que causa su activación. La información aparece cuando se hace clic sobre la imagen de los elementos, y desaparece cuando se mueve el cursor hacia una nueva posición. Estos datos se muestran organizados mediante un cuadro que aparece cerca de la ubicación del cursor. En la siguiente figura se muestra el cuadro utilizado.

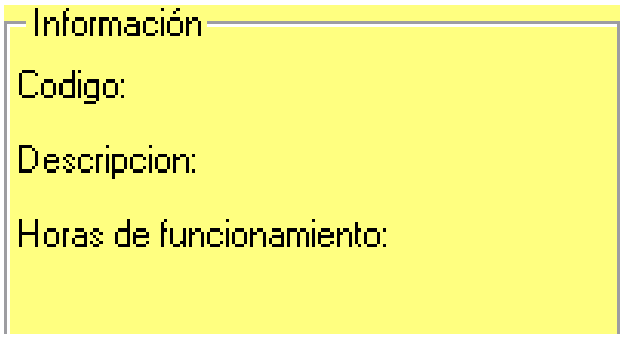


Figura 2.71. Cuadro utilizado para mostrar información

El comando utilizado para leer el código y la descripción de los actuadores es el siguiente:

"SELECT elemento_co,descripcion FROM elementos_plc where numero_vb='" & Trim$(motor(Index).Index) & "'"

Para obtener las horas de funcionamiento el código empleado es:

SqlQry = "SELECT sum(intervalo) FROM eventos where elemento_id=(select elemento_id from elementos_plc where numero_vb='" & Trim$(motor(Index).Index) & "')"

Este código realiza la suma de los intervalos de tiempo en los que el motor ha estado funcionando. En la base de datos se calcula el valor del intervalo de un evento el momento en el que el actuador se apaga. Para obtener el valor real del tiempo de funcionamiento de un motor que está encendido, incluyendo el tiempo desde su último arranque, es necesario calcular el intervalo actual. Para lo cual se debe escribir en la base de datos la hora actual como hora de finalización del evento y luego calcular el intervalo. Luego de realizar la suma de los intervalos se deben borrar estos datos que se escribieron para el motor que se encuentra encendido. El código que escribe la hora actual en la columna hora final de un evento es el siguiente:

SqlQry = "update eventos set hora_fin= current_timestamp where evento_id='" & evento_id(Index) & "'"

Para calcular el intervalo el código utilizado es el siguiente:

SqlQry = "update eventos set intervalo= ((select hora_fin from eventos where evento_id='" & evento_id(Index) & "')-(select hora_inicio from eventos where evento_id='" & evento_id(Index) & ")) where evento_id='" & evento_id(Index) & "'"

Luego de calcular la suma de los intervalos se borra el valor colocado en la columna hora_fin mediante el siguiente comando SQL:

SqlQry = "update eventos set hora_fin= null where evento_id='" & evento_id(Index) & "'"

Para borrar el valor del intervalo se utiliza el comando:

SqlQry = "update eventos set intervalo= null where evento_id='" & evento_id(Index) & "'"

Cuando se hace clic sobre la imagen de un sensor se muestra su información. Para obtener los datos de los sensores desde la base de datos se utiliza el siguiente comando SQL:

SELECT elemento_co,descripcion,señalización FROM elementos_plc where numero_vb='" & Trim$(sensor(Index).Index) & "'"

2.3.10.5 Implementación del arranque y parada de la planta desde el computador

Mediante la implementación del sistema SCADA se pretende que todas las operaciones de arranque y parada de la planta de producción se realicen desde los computadores. Por lo que el software del sistema SCADA debe permitir realizar estas operaciones. En el software se han colocado iconos y se ha realizado la programación necesaria para cumplir este objetivo.

Para que los operadores puedan adaptarse fácilmente al uso del sistema SCADA, en las pantallas se han colocado los mismos selectores ubicados en los tableros de control. Por lo que el procedimiento para realizar el arranque y la parada de la maquinaria desde el software del sistema SCADA es el mismo que utilizando el tablero de control.

En la siguiente imagen se muestran los selectores y pulsadores colocados en las pantallas de la sección molienda.

Figura 2.72. Selectores y pulsadores de la pantalla de la sección molienda

Los selectores y pulsadores de la sección limpieza se muestran en la siguiente imagen:

Figura 2.73. Selectores y pulsadores de la pantalla de la sección limpieza

Cuando el PLC detecta que se pulsó uno de los botones de paradas de emergencia activa un relé de control. Desde el software del sistema SCADA también se activa el mismo relé de control cuando el operador hace clic sobre el pulsador. Por ejemplo, la rutina que realiza la parada total de la sección molienda se ejecuta cuando se activa el relé de control C201. Por lo tanto, cuando se desea realizar la parada total desde la aplicación desarrollada en Visual Basic se coloca en uno lógico este relé de control. El código mostrado a continuación realiza esta operación.

```
numero_plc(1) = C200.numero_opc
valor(1) = 1
OPCItemWrite 1, numero_plc(), valor()
```

El PLC realiza el arranque automático de una sección cuando verifica que el operador ha ordenado esta acción mediante los selectores del tablero de control.

Cuando se ordena la acción desde el SCADA, se activan relés de control colocados en los diagramas escalera en paralelo con las señales de los selectores. Estas señales se habilitan cuando se activa el relé de control C20 al ingresa a modo de control. Cuando se activa C20 se deshabilitan las señales provenientes de los tableros. De esta manera, cuando el software se encuentra en modo de control, las operaciones de arranque se pueden realizar únicamente desde el computador. La imagen a continuación muestra una sección del diagrama escalera en la cual se pueden observar los relés de control ubicados en paralelo con la señales de los selectores.

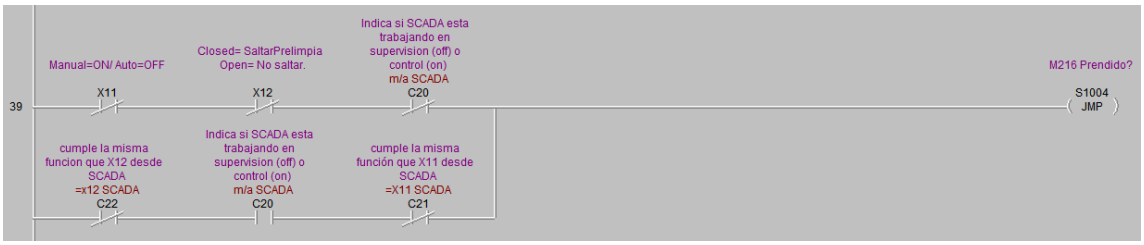


Figura 2.74. Diagrama escalera con relés de control en paralelo con señales de selectores del tablero de control.

El código mostrado a continuación activa desde el software del sistema SCADA el relé auxiliar C22 para realizar el arranque automático de la etapa de pre limpieza.

```
numero_opc(1) = C22.numero_opc
valor(1) = 1
OPCItemWrite 1, numero_opc(), valor()
```

Cuando se desea detener el arranque automático de la etapa de pre-limpieza el relé de control C22 se coloca en cero lógico mediante el código mostrado a continuación:

```
numero_opc(1) = C22.numero_opc
valor(1) = 0
```

OPCItemWrite 1, numero_opc(), valor()

El arranque de las otras etapas se realiza de la misma manera pero activando el relé de control correspondiente a cada una. Por ejemplo, para realizar el arranque automático de la etapa segunda limpieza se coloca en uno lógico el relé C24.

2.3.10.6 Implementación de detector de comunicaciones

La comunicación entre el computador y el PLC puede desaparecer por varias razones. Por ejemplo, se puede dañar o desconectar el cable utilizado, o se puede quitar la alimentación del PLC. Cuando se interrumpe la alimentación al PLC se apaga toda la maquinaria de la sección que controla. Pero el OPC server, y por lo tanto la aplicación del sistema SCADA, no reciben los nuevos valores de las salidas del PLC y debe asumir que no han cambiado. Cuando esto sucede no se registra en la base de datos el apagado de los motores, por lo que el valor de las horas de funcionamiento de los actuadores se vuelve incorrecto. Por esto es importante conocer el estado de la comunicación para escribir en la base de datos la hora de apagado de los motores.

Una de las funciones del OPC driver es informar el estado de la comunicación. Cuando existe un cambio en el valor de algún dato, el OPC Server envía también una variable en la que se indica si la calidad del valor enviado es buena o no. Es decir, cuando existe comunicación con el PLC, la calidad de los datos es buena, y si no hay comunicación, la calidad es mala. El nombre de esta variable es *Qualities* y si se revisa su valor se determina el estado de la comunicación. Cuando su valor es uno, la comunicación es correcta. Si su valor es diferente de uno existe un error en la comunicación.

En la parte superior de la pantalla se muestra un texto indicando si existe o no comunicación. Esto también se muestra gráficamente. El cuadrado de color verde indica una comunicación correcta. Mientras que cuando es rojo representa un error.

En las imágenes mostradas a continuación se puede observar las indicaciones en ambos casos.

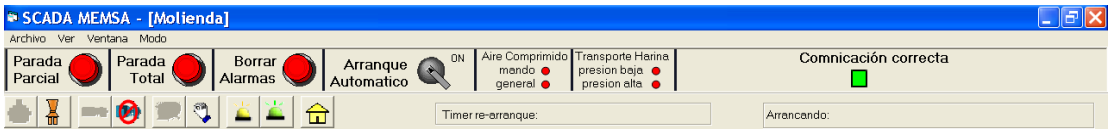


Figura 2.75. Indicación de comunicación correcta

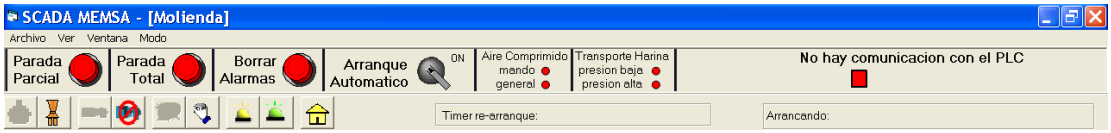


Figura 2.76. Indicación de error en la comunicación

Cuando no hay comunicación, se considera que todos los actuadores se han apagado y se escribe en la tabla correspondiente la hora de finalización. Además, se impide realizar cualquier maniobra. La ventana *administración de programa* se vuelve visible pero todos sus botones aparecen deshabilitados. Cuando la comunicación regresa a su estado normal, se habilitan los botones para poder acceder a la pantalla deseada.

2.3.10.7 Implementación de rutinas para cerrar la aplicación

Si se cierra el software del sistema SCADA se detiene el proceso de producción. De esta manera se evita perder información valiosa. Cuando no está funcionando el software no se puede escribir en la base de datos información del encendido y apagado de las máquinas, por lo que el valor de las horas de funcionamiento no sería correcto. Además, si ocurre una alarma tampoco quedaría registrada.

Cuando arranca el software del sistema SCADA se coloca en uno lógico el relé de control C50. Esta señal indica al PLC que puede hacer funcionar la maquinaria si el operador lo desea. Al cerrar la aplicación, este relé se coloca en cero lógico. Cuando la señal tiene este valor el PLC apaga la maquinaria que se encuentre funcionando y no permite que arranque nuevamente hasta que entre en funcionamiento el software.

Para evitar que se produzca el cierre accidental del programa, y por tanto la parada de la maquinaria, se ha implementando una seguridad. Cuando el operador selecciona la opción salir, ya sea desde el menú o pulsando sobre el botón X ubicado en la parte superior derecha de la pantalla, aparece un mensaje confirmando la acción. El operador puede seleccionar no cerrar la aplicación y continuar trabajando normalmente. Si el usuario está seguro de cerrar el programa deberá pulsar sobre el botón *si*, y el programa se cerrará. El mensaje que aparece se muestra en la siguiente figura:

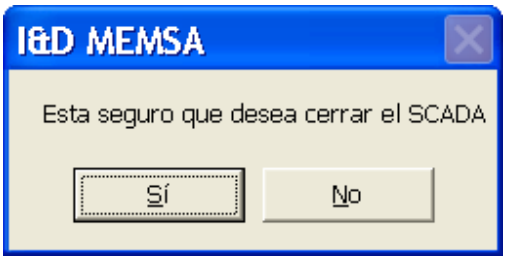


Figura 2.77. Mensaje preguntando al operador si desea cerrar la aplicación

Si se seleccionó cerrar la aplicación se escribe en la base de datos la hora de apagado de todos los motores que estén funcionando en ese momento.

2.3.10.8 Elaboración de mensajes en el panel de operador para indicar funcionamiento de SCADA

Es importante que el operador pueda determinar fácilmente si el control de la maquinaria se está realizando desde el computador o desde el tablero de control. Es

decir, si el software del sistema SCADA está trabajando en modo de supervisión (control desde el tablero de control) o en modo control (control desde el computador). En las pantallas de operación de la aplicación del sistema SCADA se muestran mensajes indicando el modo de operación. También se debe indicar el modo en el que se encuentra el software en el tablero de control. Por lo que se muestra un mensaje en uno de los paneles conectados al PLC.

Cuando se ingresa al modo de control se activa el relé C20. De esta manera se avisa al PLC que el encendido y apagado de la maquinaria se lo va a realizar desde el computador. En el programa del PLC, cuando el estado de este relé de control cambia de cero a uno, se ejecuta una rutina en la cual se muestra en el panel *Opti-Mate OP-1510* un mensaje indicando al operador que debe realizar las operaciones desde el computador. Este mensaje permanece en la pantalla mientras el software del sistema SCADA esté en modo de control. Cuando el modo seleccionado es supervisión, nuevamente se permite realizar las operaciones desde el tablero de control y el mensaje desaparece.

4 PRUEBAS Y RESULTADOS

4.1 INTRODUCCIÓN

Con el objetivo de verificar el correcto funcionamiento del sistema SCADA implementando en Molino Electro Moderno S.A. se han realizado varias pruebas de las diferentes funciones del software. Luego se han analizado los resultados y se han realizado correcciones cuando han sido necesarias. En el presente capítulo se describen las pruebas realizadas y los resultados obtenidos.

4.2 PRUEBAS DE COMUNICACIÓN

Para que el sistema funcione correctamente es indispensable establecer las comunicaciones y que éstas funcionen bien. Debe existir comunicación entre el PLC y el OPC server, entre el OPC server y el programa desarrollado en Visual Basic y entre la aplicación del sistema SCADA y la base de datos. En esta sección se describen las pruebas realizadas para la transmisión de información entre los elementos indicados anteriormente.

4.2.1 COMUNICACIÓN ENTRE LA BASE DE DATOS Y EL SOFTWARE DEL SISTEMA SCADA

Para comprobar el funcionamiento de la comunicación entre la base de datos y el software del sistema SCADA se realizó el procedimiento descrito a continuación: se creó un nuevo usuario desde el software del sistema SCADA de modo que la información ingresada se escriba en la base de datos; luego se leyó la tabla usuarios de la base de datos desde el command prompt utilizando un comando SQL para comprobar que la información se escribió correctamente.

Una vez que se verificó que la información de la base de datos concordaba con la ingresada se procedió a probar la lectura de datos desde la aplicación. Se leyeron desde la aplicación y directamente en la base de datos los eventos de un motor ocurridos en un día. Luego se comparó la información para determinar si la lectura de los datos fue correcta.

Los resultados de estas pruebas fueron satisfactorios. Desde la aplicación desarrollada en Visual Basic se puede leer y escribir información a la base de datos creada en PostgreSQL.

4.2.2 COMUNICACIÓN ENTRE OPC SERVER Y PLC

Para comprobar que la comunicación entre el OPC server y el PLC es correcta se utilizó la herramienta OPC Quick Client. Ésta es una aplicación incluida en el OPC server. En ésta se pueden observar los valores de todas las etiquetas que se han agregado en el servidor OPC. También se puede determinar si la comunicación es correcta ya que muestra la calidad de los datos (good o bad). Para determinar que el canal de comunicación está funcionando y que se están recibiendo los datos basta verificar que la calidad de los datos sea buena. Para comprobar que los datos recibidos son iguales a los datos reales se compara la información obtenida mediante el OPC Quick Client y el valor actual de la variable observado en el software de programación del PLC (Directsoft).

La comunicación entre los servidores OPC y los PLCs funciona correctamente. La calidad de los datos es buena y el valor que se lee en el servidor concuerda con el valor real de las variables. A continuación se muestra una imagen del programa OPC Quick Client en la cual se pueden observar el valor y la calidad de algunas variables.

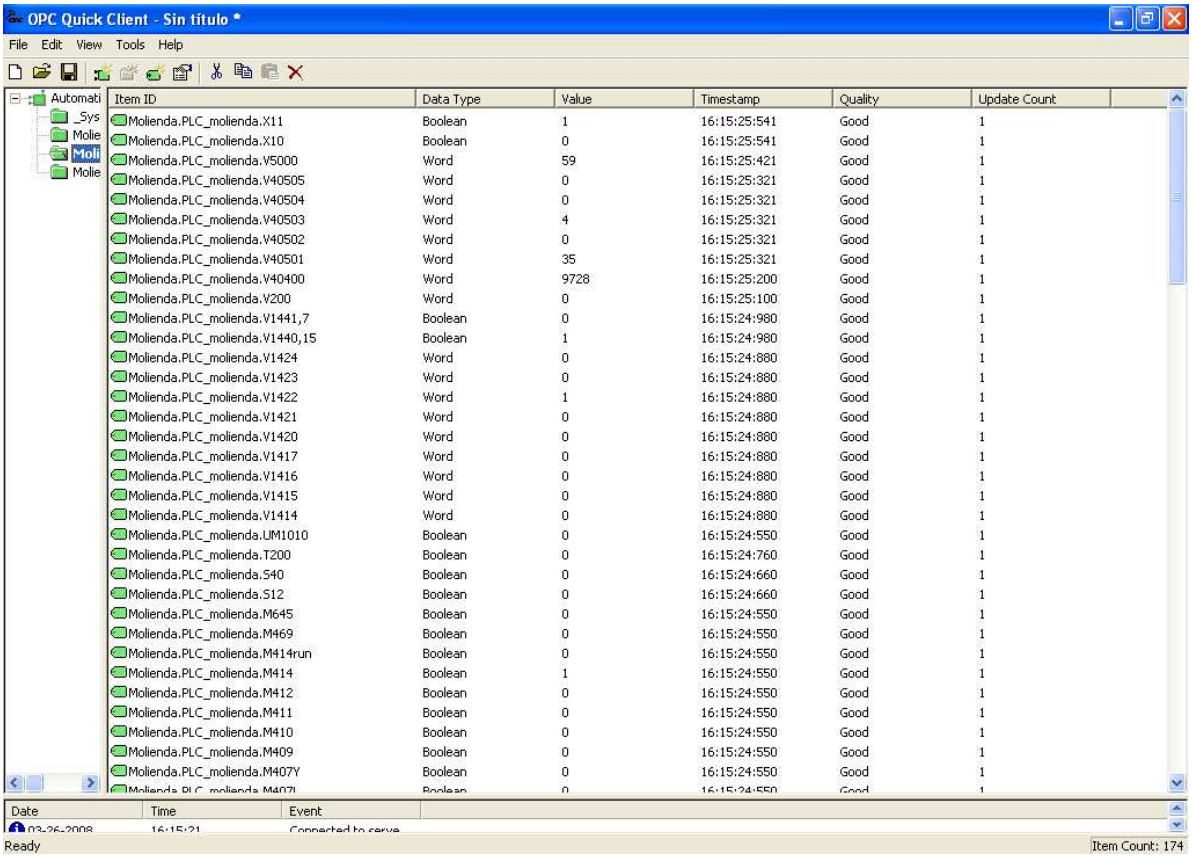


Figura 3.1. Programa OPC Quick Client mostrando la calidad y valor de algunas variables

4.2.3 COMUNICACIÓN ENTRE VISUAL BASIC Y OPC SERVER

La comunicación entre el software del sistema SCADA y el servidor OPC se comprobó corriendo la aplicación y observando que indique el estado real de la planta. Se verificó que se muestren todos los motores apagados cuando están en este estado. También se prendieron algunos motores y se activaron protectores térmicos para verificar que en la pantalla se muestren las indicaciones visuales adecuadas.

Luego de realizar esta prueba se comprobó que la comunicación es correcta. Los indicadores de las pantallas coinciden con el estado real de la planta. Además, se

comprobó que se puede realizar el apagado y encendido de motores desde la aplicación. Lo que implica que se cambie el valor de las salidas del PLC a través del servidor OPC. Por lo tanto se concluye que la escritura y la lectura de datos entre la aplicación y el OPC sever funciona del modo esperado.

4.2.4 DETECTOR DE COMUNICACIÓN ENTRE VISUAL BASIC Y SERVIDOR OPC

En las aplicaciones se implementó un detector de comunicación. Este indica si la comunicación entre los programas y el servidor OPC funciona correctamente. Para probar esta función se arrancó la aplicación y se desconectó el cable de comunicación. En ese momento se verificó que en la pantalla aparezca el mensaje y la indicación gráfica avisando al operador la falla en la comunicación. También se probó que cuando la comunicación falla las funciones del software se deshabiliten.

Luego de realizar esta prueba se concluye que el detector trabaja adecuadamente. Cuando la comunicación falla se muestra un mensaje alertando al operador del problema y se deshabilitan las funciones impidiendo que se trate de enviar datos al PLC.

4.3 PRUEBAS DE OPERACIÓN

En esta sección se muestran las pruebas realizadas para determinar si las órdenes enviadas desde la aplicación del sistema SCADA se ejecutan correctamente. Estas pruebas consistieron en realizar desde el software del sistema SCADA el encendido y apagado manual de motores, arranques automáticos de cada sección y paradas parciales y totales. A continuación se describe dichas pruebas y los resultados obtenidos.

4.3.1 ENCENDIDO MANUAL DE MOTORES

Para verificar que el encendido manual de motores funciona correctamente se procedió a realizar esta operación a todos los motores uno por uno. Se enviaba la orden de encendido y se comprobaba que el motor deseado arrancaba. Al momento de realizar esta prueba también se comprobó que los motores con arranque estrella – triángulo se enciendan de modo adecuado.

Luego de terminar la prueba a todos los motores de la planta se determinó que el encendido manual de los motores funciona correctamente. También se verificó que los motores que arrancan mediante el método de estrella – triángulo se prenden del modo esperado.

4.3.2 APAGADO MANUAL DE MOTORES

Luego de realizar las pruebas de encendido de los motores se procedió a probar el apagado de los mismos. Se ordenó el apagado de cada motor y se fue comprobando que en realidad se apague el motor deseado. En el caso de los motores con arranque estrella – triángulo se verificó que se apaguen todos los contactores utilizados. También se realizaron pruebas apagando el motor mientras está en la conexión estrella para determinar que se pueda apagar en esta condición y que no sea necesario esperar hasta que se encuentre en conexión triángulo.

Luego de realizar esta prueba a todos lo motores se obtuvieron resultados satisfactorios tanto para los motores con arranque directo como para aquellos con arranque estrella – triángulo. Para estos últimos se comprobó que se apagan los tres contactores utilizados para el arranque y que se puede apagar el motor en cualquier momento.

4.3.3 ENCENDIDO AUTOMÁTICO DE LA ETAPA DE MOLIENDA

Para iniciar la producción de harina la maquinaria de la etapa de molienda debe ser arrancada en un orden definido. Para probar que se puede realizar del arranque automático desde la aplicación del sistema SCADA se procedió a enviar la orden de arranque de la etapa y verificar que se cumpla la secuencia correcta.

Luego de enviar la orden de arranque automático se fue verificando que se prenda el motor correspondiente según la secuencia establecida. Durante la prueba se verificó que se prende el motor en la planta y que en la pantalla del SCADA también se indica su estado real.

En la pantalla se colocó un casillero en el cual se muestra un mensaje indicando qué motor está arrancando en ese momento. También se comprobó que el mensaje indicado es el correcto.

4.3.4 ENCENDIDO AUTOMÁTICO DE LAS ETAPAS DE PRE-LIMPIEZA, PRIMERA LIMPIEZA Y SEGUNDA LIMPIEZA

Del mismo modo que en la etapa de molienda, la maquinaria de las etapas de pre-limpieza, primera limpieza y segunda limpieza deben ser arrancadas siguiendo secuencias definidas. Para probar que las etapas arrancan del modo deseado se enviaron las órdenes desde el software del sistema SCADA y se verifico el cumplimiento de las secuencias.

Primero se envió la orden de arrancar la etapa de prelimpieza y se verificó que arranque correctamente. Luego se repitió este procedimiento para las otras dos etapas. En esta aplicación también se colocó un casillero en el cual se muestra un mensaje indicando que maquinaría está arrancando en ese momento. También se verificó que el mensaje mostrado sea correcto.

Luego de realizar estas pruebas se comprobó que el arranque de las tres etapas de limpieza se realiza correctamente desde el software del sistema SCADA. Las secuencias establecidas se cumplen y en las pantallas se puede observar como cambia el estado de la maquinaria que se va encendiendo. El mensaje de texto que indica el nombre de la maquinaria que se está prendiendo en ese momento también es correcto.

4.3.5 PARADAS DE LA ETAPA DE MOLIENDA

En la etapa de molienda pueden producirse dos tipos de paradas: total y parcial. Cuando la parada se debe a la activación de una alarma se produce una de las dos paradas dependiendo de la máquina involucrada. Como su nombre lo indica, la parada total apaga casi toda la maquinaria de la etapa. La parada parcial apaga solo algunas de las máquinas. Cuando se ejecuta una parada total tres motores siguen funcionando durante un intervalo de tiempo para evitar que se atore el producto.

En algunas ocasiones es necesario ordenar la ejecución de una de estas paradas. En las pantallas del software se colocaron botones para cumplir estas funciones. Se colocó un botón para ordenar una parada total y uno para ordenar una parada parcial.

Para probar esta función se encendió la maquinaría de la etapa de molienda y se enviaron las ordenes de las paradas. Primero se pulsó sobre el botón de parada parcial y se verificó que se apague la maquinaria que debe apagarse en este tipo de parada. Luego se encendió nuevamente toda la maquinaria y se realizó la parada total para verificar el apagado de las máquinas.

Los resultados de esta prueba fueron satisfactorios. Cuando se envió la orden para realizar la parada parcial se apagó toda la maquinaria que debía hacerlo. La parada total también funcionó correctamente ya que se apagó la maquinaria esperada.

4.3.6 PARADA DE LAS ETAPAS DE PRE-LIMPIEZA, PRIMERA LIMPIEZA Y SEGUNDA LIMPIEZA

Desde el tablero de control se pueden realizar las paradas de las tres etapas de limpieza. En el tablero hay tres pulsadores que activan cada una de las paradas. En las pantallas del software de la etapa de limpieza también se colocaron tres pulsadores que cumplen la misma función que los del tablero.

Cuando se realiza la parada de una etapa se debe apagar toda la maquinaria involucrada en la etapa. Las máquinas de las otras dos etapas debe permanecer en el mismo estado en que se encontraban.

Para probar que se pueden realizar las paradas de las etapas de limpieza desde el software del sistema SCADA se procedió de la siguiente manera: se encendió toda la maquinaria de las tres etapas y se ordenó desde la aplicación la parada de la etapa de pre-limpieza; luego se encendió nuevamente toda la maquinaria y se ordenó la parada de la etapa de primera limpieza; para probar la parada de la segunda limpieza se procedió de igual modo.

Luego de realizar estas pruebas se concluye que esta función del software tabaja correctamente. Desde la aplicación se puede realizar la parada de las tres etapas de limpieza. También se comprobó que si están funcionando dos o tres etapas de limpieza y se para una de ellas, la otra etapa, en el primer caso, o las otras dos etapas, en el segundo caso, siguen funcionando.

4.3.7 SEGURIDADES PARA EVITAR OPERACIONES NO DESEADAS

Para evitar operaciones no deseadas es necesario que el programa esté funcionando en modo de control y pulsar el botón adecuada del teclado para encender o apagar un actuador. Para verificar que esta función trabaja adecuadamente se trató de encender y apagar actuares sin cumplir estas condiciones. Es decir, se trató de

realizar las operaciones cuando el software estaba en modo de supervisión y también se trataron de realizar las operaciones sin oprimir las teclas adecuadas.

No fue posible realizar las operaciones sin cumplir con las condiciones explicadas en el párrafo anterior. Por lo tanto, las seguridades implementadas en el software del sistema SCADA para evitar operaciones no deseadas funcionan correctamente.

4.4 PRUEBAS DE SENSORES Y ALARMAS

Todos los sensores y alarmas se activaron manualmente uno por uno para determinar si el software del sistema SCADA muestra su estado real. De este modo también se verificó que las aplicaciones realicen las funciones de borrar las banderas de las alarmas del PLC y que permitan ingresar el motivo que ocasionó una alarma.

Cuando ocurre una alarma se activa una bandera en el PLC que impide encender la maquinaria. Para realizar el re-arranque de la producción primero se debe borrar dicha bandera. Esta función se realizaba presionando teclas de los paneles ubicados en el tablero de control. Para prescindir totalmente del uso del tablero esta función se implemento en el software.

Para realizar estas pruebas primero se utilizó un PLC de repuesto que había en las bodegas de la fábrica. Se programó la lógica que gobierna la maquinaria de la etapa de limpieza, se conecto un computador en el cual se hizo correr la aplicación y se procedió a activar los sensores y las alarmas. Se verificó que en la pantalla se muestre el estado de la entrada activada y que se muestre el mensaje mostrando que alarma se activo. Luego se ejecutó la rutina para ingresar el motivo que ocasionó la alarma y se comprobó que funcione correctamente. Para finalizar esta prueba se comprobó que se borre la bandera del PLC y que se puede arrancar nuevamente la maquinaria. Este mismo procedimiento se realizó programando la lógica de la etapa de molienda y haciendo correr en el computador la aplicación de esta etapa.

Luego se aprovechó una parada para mantenimiento para realizar las pruebas con los PLCs conectados a las máquinas y a los sensores y protectores térmicos. En esta ocasión se activaron los protectores térmicos primero con la maquinaría apagada y luego cuando estaba encendida.

Luego de realizar estas pruebas se concluye que estas funciones del software trabajan satisfactoriamente. En las pantallas de las aplicaciones se muestran las alarmas activadas y el mensaje de texto colocado en la parte superior de la pantalla indica su código. Al hacer doble clic sobre las alarmas aparece la pantalla utilizada para ingresar el motivo. Esta pantalla permanece visible hasta que el operador coloque en el casillero correspondiente el motivo de la alarma. Al realizar estas operaciones el software se encarga de borrar las banderas del PLC para permitir arrancar nuevamente la maquinaria. Estas funciones trabajan del modo esperado cuando la máquina se encuentra prendida y cuando no está funcionando.

4.5 PRUEBAS DE LAS FUNCIONES DEL SOFTWARE

En esta sección se describen las pruebas realizadas a las funciones del software. Se probó el correcto funcionamiento del menú, de la barra de tareas y del ingreso y de la eliminación de usuarios. También se comprobó que el software permita cambiar la clave de los usuarios y salir del programa. Además, al final de la sección se encuentran las pruebas realizadas para comprobar el funcionamiento del indicador de información y de las animaciones.

4.5.1 MENÚ

El menú se encuentra en la parte superior de la pantalla. Permite al usuario ejecutar funciones del software y navegar entre las diferentes ventanas de las aplicaciones.

Las funciones del menú se probaron ejecutándolas una por una. Se verificó que todas las opciones trabajen del modo esperado. Se hizo clic sobre ellas y se observó que cumplan su función. Por ejemplo, luego de hacer clic sobre la opción *ocultar motores* ubicada dentro del menú *ver*, se determinó que las imágenes de los motores ubicadas en las pantallas de operación se vuelvan invisibles. También se comprobó que las funciones se habiliten y deshabiliten cuando deben hacerlo. Por ejemplo, se comprobó que la función que permite acceder a la ventana *molienda* se deshabilite cuando ésta ventana está activa.

Todas las funciones del menú funcionan correctamente. Todas las opciones ejecutan la función debida y el menú nos permite navegar entre las ventanas de las aplicaciones del modo previsto.

4.5.2 BARRA DE HERRAMIENTAS

La barra de herramientas es un conjunto de iconos o imágenes que al pulsar sobre ellos con el mouse ejecutan una función. En la barra de tareas se colocaron las funciones de uso más frecuente con la finalidad de que el operador pueda acceder a ellas rápidamente. La barra de herramientas se encuentra en la parte superior de la pantalla debajo del menú.

Del mismo modo que con el menú, para probar las funciones de la barra de tareas se ejecutaron todas una por una. Se hizo clic sobre sus iconos y se verificó que cumplan la función esperada. También se verificó que los iconos ve habiliten y deshabiliten en el momento adecuado. Por ejemplo, el icono para seleccionar modo de control se debe deshabilitar al escoger este modo y se debe habilitar el icono para escoger modo de supervisión.

Estas funciones trabajan del modo esperado. Todos los botones de la barra de tareas realizan la función para la cual fueron creados.

4.5.3 INGRESO DE USUARIO

Cuando se abre una de las aplicaciones del sistema SCADA aparece en la pantalla un solo botón. Al hacer clic sobre éste se ejecuta la función que permite al usuario ingresar al software y obtener acceso a sus funciones.

En esa prueba se verificó que cuando el programa se arranca no sea posible visualizar otra ventana ni utilizar las funciones de la barra de herramientas ni del menú antes de ingresar. Se comprobó además que solo se pueda ingresar utilizando un nombre de usuario y una contraseña que se hayan creado previamente. Para finalizar se comprobó que las opciones agregar usuarios y eliminar usuarios solo se habilitan cuando se ingresa utilizando un nombre de usuario cuyo acceso es de administrador.

Las pruebas descritas en el párrafo anterior resultaron exitosas. El control de acceso es efectivo ya que resulta indispensable utilizar un nombre de usuario y una clave adecuada para utilizar las funciones del sistema.

4.5.4 CREACIÓN DE UN USUARIO

Es necesario que el software permita crear nuevos usuarios. Un usuario con acceso de tipo administrador puede acceder a esta opción haciendo clic sobre el icono ubicado en la ventada de administración del programa. Al hacer clic aparece una ventana creada para este fin. En ella se debe ingresar alguna información del nuevo usuario antes de poder crearlo. Luego de llenar toda la información requerida el usuario es creado pulsando sobre el botón colocado en la parte inferior derecha de la pantalla.

Para probar el funcionamiento de esta opción se procedió a crear varios usuarios con diferentes tipos de accesos. Luego se comprobó que el software permita acceder a sus funciones utilizando los nombres de usuarios y las claves recién creadas.

También se comprobó que el software no permita crear un usuario nuevo si no se ingresa toda la información requerida o si el nombre de usuario ingresado ya existe.

Luego de realizar las pruebas descritas se comprobó que la opción funciona correctamente. El ingreso de nuevos usuarios trabaja del modo esperado y solo los usuarios con acceso de tipo administrador la pueden utilizar.

4.5.5 ELIMINACIÓN DE USUARIOS

Es necesario poder eliminar usuarios a los cuales ya no se les quiere permitir acceder al software. Esta opción está disponible solo para los usuarios con tipo de acceso de administrador. Para acceder a ella se debe hacer clic sobre el botón colocado en la ventana administración del programa.

Para probar esta función se ingresó a la ventana correspondiente y se eliminó uno de los usuarios creados. Luego se trató de ingresar al software utilizando el nombre de usuario recientemente eliminado. También se leyeron los registros de la tabla usuarios de la base de datos para comprobar que el usuario eliminado ya no se encuentre en la lista.

El resultado de esta prueba fue que no se pudo ingresar al software utilizando el nombre de usuario eliminado y que se borró el registro de la base de datos. Por lo tanto se concluye que esta función trabaja adecuadamente.

4.5.6 CAMBIO DE CLAVE

Esta función permite a los usuarios del software cambiar su clave de ingreso. Para ingresar a la ventana donde se ejecuta esta operación se debe hacer clic en el botón ubicado en la ventana administrador del programa.

Para probar que esta función trabaja del modo esperado se procedió a cambiar la contraseña de uno de lo usuarios y luego se ingresó nuevamente al software. Primero se intentó ingresar con la clave anterior y luego con la nueva. También se intento cambiar la contraseña sin ingresar toda la información requerida.

Los resultados de estas pruebas fueron satisfactorios. Una vez cambiada la contraseña es indispensable utilizar la nueva clave para acceder a las funciones de la aplicación ya que la contraseña anterior ya no lo permite. Para realizar el cambio de clave se debe ingresar la clave anterior, la nueva clave y una confirmación de la nueva clave. Si no se ingresan todos estos datos o los datos ingresados no concuerdan no es posible realizar la operación.

4.5.7 SALIR DEL PROGRAMA

Se implementó un control para evitar que las aplicaciones se cierren de modo accidental. Al momento de pulsar sobre la X en la parte superior derecha de la pantalla o sobre la opción salir ubicada en el menú aparece un cuadro con un mensaje preguntando al operador si desea cerrar la aplicación.

Para probar que esto funciona correctamente se procedió a cerrar la aplicación utilizando los dos métodos descritos en el párrafo anterior. Luego se pulsó sobre la opción *sí* y se verificó que la aplicación se cierre. Después se cerró nuevamente la aplicación pero en esta ocasión se pulsó sobre la opción *no* y se verificó que el programa continúe funcionando.

El control para impedir el cierre accidental de la aplicación funciona correctamente. Si luego de enviar la orden para cerrar el programa se selecciona la opción *no* la aplicación sigue funcionando normalmente. En cambio, si se pulsa sobre la opción *sí* el programa se cierra.

4.5.8 INDICADOR DE INFORMACIÓN

En el indicador de información se muestran datos de los actuadores y sensores. Al hacer clic sobre un motor aparece en pantalla un cuadro en el que se encuentra organizada la siguiente información: el código del elemento, su descripción y el número de horas de funcionamiento. En el caso del martillo neumático y de las compuertas se muestra la misma información pero en vez de las horas de funcionamiento se muestra el número de conmutaciones. Cuando se hace clic sobre los sensores en el cuadro de información se muestra: su código, el nombre de la máquina en la cual se encuentra conectado y que es lo que provoca su activación (por ejemplo: exceso de producto).

Para probar que el cuadro de información funcione del modo esperado se pulsó sobre todos los actuadores y sensores y se comprobó que la información mostrada sea correcta. También se realizó la siguiente prueba: se pulsó dos veces consecutivas sobre los motores cuando estaban prendidos y se verificó que la indicación de tiempo de funcionamiento incremente.

Estas pruebas fueron exitosas. El cuadro de información funciona del modo esperado. La información mostrada es correcta tanto para los actuadores como para los sensores.

En la siguiente imagen se puede observar el cuadro de información que aparece luego de pulsar sobre el motor M410 de la etapa de molienda.

Figura 3.2. Cuadro de información mostrando datos del motor M410

4.5.9 ANIMACIONES

Las animaciones permiten determinar rápidamente el estado de algunas de las máquinas y mejoran la apariencia de las pantallas. Para probar que funcionan correctamente se procedió a encender la máquina a la cual simulan y comprobar que la animación de la pantalla represente el estado real de dicha máquina. También se verificó que el tamaño de las animaciones sea adecuado y que permitan una correcta visualización de la pantalla.

Luego de realizar las pruebas se comprobó que las animaciones cumplen con su objetivo. Indican el estado de la maquinaria, son fáciles de visualizar y mejoran la apariencia de la pantalla.

4.6 PRUEBAS DE LECTURA DE REGISTROS DE LA BASE DE DATOS

En el software del sistema SCADA se colocaron dos ventanas para leer los registros de la base de datos. En la una se leen los registros de las alarmas y en la otra los

registros de los eventos. Para probar que estas funciones trabajen del modo esperado se ingresó a cada ventana y se solicitó información.

Primero se probó la lectura de los registros de alarmas. Se ingresó a la ventana respectiva, se seleccionó el periodo de tiempo del cual se deseaba conocer las alarmas y se pulsó sobre el botón ubicado en la parte inferior del formulario. Luego se leyeron directamente de la base de datos los registros pertenecientes a dicho periodo y se compararon con los obtenidos desde el software del sistema. También se comprobó que los registros mostrados sean reales. Es decir, que las alarmas registradas hayan sucedido y que todas las alarmas que sucedieron se encuentren registradas.

Luego se realizó el mismo procedimiento para probar la lectura de los registros de los eventos. En este caso también se verificó la veracidad de la información mostrada y se comprobó que los datos mostrados en la aplicación concuerden con los almacenados en la base de datos.

Los resultados de estas pruebas fueron satisfactorios. Los registros de las alarmas y de los eventos se pueden leer desde las ventanas alarmas y eventos respectivamente. La información mostrada concuerda con la almacenada en la base de datos y con los hechos ocurridos en la planta de producción.

4.7 PRUEBA DE EXPORTACIÓN DE INFORMACIÓN A MICROSOFT EXCEL

En las ventanas eventos y alarmas se agregó una función para exportar la información obtenida de la base de datos a una hoja de cálculo de Microsoft Excel. En la parte inferior izquierda se colocó un botón con el símbolo de Excel. Al pulsar sobre él la información de los eventos o de las alarmas ocurridas es enviada a dicho programa.

Para determinar si esta función trabaja del modo esperado se realizaron pruebas utilizando los registros de las alarmas y de los eventos. Desde cada una de estas ventanas se solicitaron registros de la base de datos. Luego se exportaron a Excel y se compararon los datos mostrados en la aplicación.

Luego de realizar esta prueba se concluye que esta función trabaja del modo esperado. Los datos exportados a Excel son iguales a los datos mostrados en la aplicación.

En las imágenes a continuación se pueden ver registros de eventos mostrados en la ventana de la aplicación y los mismos registros luego de ser exportados a Excel.

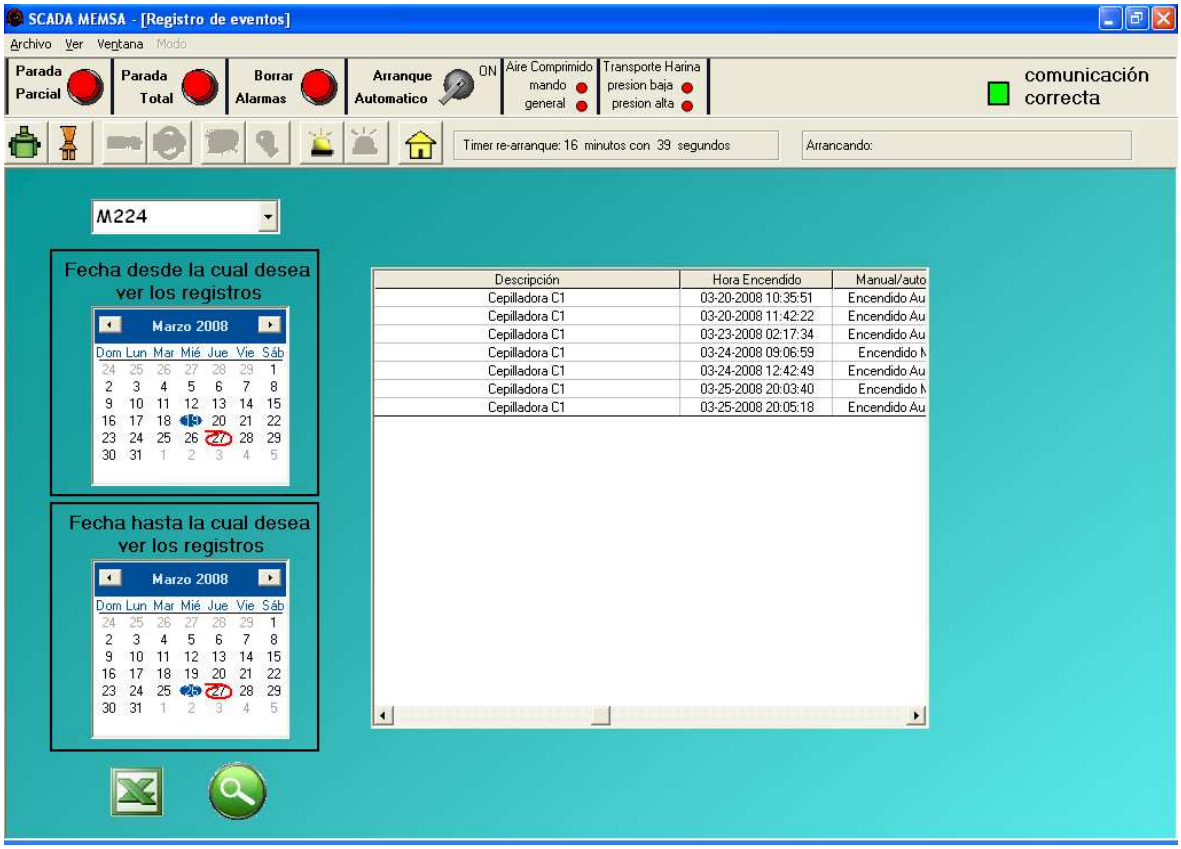
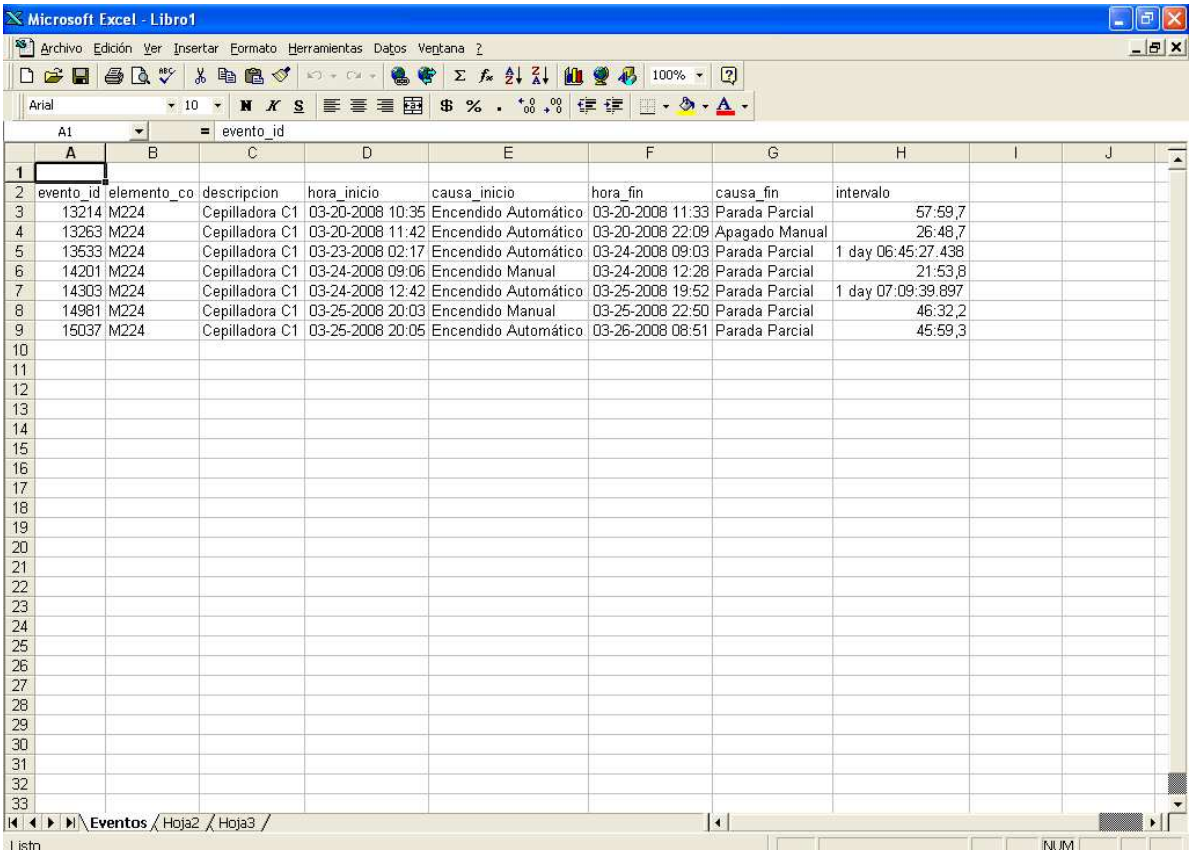


Figura 3.3. Registros de eventos mostrados en la aplicación del sistema SCADA



Microsoft Excel - Libro1

Archivo Edición Ver Insertar Formato Herramientas Datos Ventana ?

Arial 10 100%

A1 = evento_id

	A	B	C	D	E	F	G	H	I	J
1										
2	evento_id	elemento_co	descripcion	hora_inicio	causa_inicio	hora_fin	causa_fin	intervalo		
3	13214	M224	Cepilladora C1	03-20-2008 10:35	Encendido Automático	03-20-2008 11:33	Parada Parcial	57:59,7		
4	13263	M224	Cepilladora C1	03-20-2008 11:42	Encendido Automático	03-20-2008 22:09	Apagado Manual	26:48,7		
5	13533	M224	Cepilladora C1	03-23-2008 02:17	Encendido Automático	03-24-2008 09:03	Parada Parcial	1 day 06:45:27.438		
6	14201	M224	Cepilladora C1	03-24-2008 09:06	Encendido Manual	03-24-2008 12:28	Parada Parcial	21:53,8		
7	14303	M224	Cepilladora C1	03-24-2008 12:42	Encendido Automático	03-25-2008 19:52	Parada Parcial	1 day 07:09:39.897		
8	14981	M224	Cepilladora C1	03-25-2008 20:03	Encendido Manual	03-25-2008 22:50	Parada Parcial	46:32,2		
9	15037	M224	Cepilladora C1	03-25-2008 20:05	Encendido Automático	03-26-2008 08:51	Parada Parcial	45:59,3		

Eventos / Hoja2 / Hoja3 /

Listo NUM

Figura 3.4. Registros de eventos exportados a Microsoft Excel

4.8 PRUEBA GENERAL DEL SISTEMA SCADA

El sistema SCADA se puso en funcionamiento en la planta de producción de Molino Electro Moderno S.A. luego de realizar las pruebas descritas en este capítulo y de comprobar que el sistema funciona correctamente. En esta etapa se continuó con el monitoreo del sistema con el objetivo de detectar problemas y solucionarlos.

El sistema SCADA ha funcionado desde Noviembre del 2007. A partir de esta fecha se han corregido algunos errores y se han realizado modificaciones solicitadas por el jefe de mantenimiento y por los operadores del molino. Algunos de los cambios

realizados han requerido suspender el funcionamiento del sistema y borrar los registros de eventos y alarmas.

Desde el mes de Febrero del 2008 el sistema ha funcionado continuamente. En este período las aplicaciones se han cerrado accidentalmente por los operadores pero volvieron a trabajar pocos minutos después. Esto ocasionó que se registren en la tabla eventos el encendido y apagado de motores al momento de abrir y cerrar la aplicación respectivamente.

Durante el periodo en el cual ha funcionado el sistema se ha procurado realizar todas las operaciones de arranque y parada de la maquinaria desde las aplicaciones. También se han eliminado las señales de alarma debido a la activación de sensores y protectores térmicos desde el software del sistema SCADA. En este periodo se ha enseñado a los operadores a realizar todas estas operaciones desde los computadores. Lamentablemente ellos todavía no han aprendido a utilizar todas sus funciones por lo que algunas de las operaciones, especialmente por las noches, fueron realizadas desde los tableros de control.

A pesar de estos inconvenientes se ha podido determinar que el sistema SCADA implementado en Molino Electro Moderno S.A. trabaja correctamente. Todas las funciones de las aplicaciones se han probado individualmente una por una y también se probó el funcionamiento en conjunto de todas. El tiempo que el sistema ha funcionado ha permitido realizar arranques automáticos de todas las etapas, paradas parciales y totales, encendido y apagado manual de motores, apertura y cierre de compuertas, manejo de alarmas, etc. Todas estas operaciones se han podido realizar sin inconvenientes.

Los objetivos planteados al inicio del proyecto se cumplieron a cabalidad por lo que se obtuvieron las ventajas que se esperaban. Durante el tiempo que ha estado funcionando el sistema se ha observado que su implementación facilitó la tarea del

operador. Las mayores ventajas para ellos se dan en los arranques y las paradas manuales de motores y en el manejo de las alarmas.

Antes de la implementación del sistema SCADA para realizar el encendido y apagado de motores era necesario conocer el código del motor. Los operadores no saben estos códigos de memoria por lo que tenían que consultarlos. Provocando demoras para realizar la operación. Además, en los tableros de control no existían indicadores de estado de todos los motores por lo que en algunos casos no había forma de comprobar que el motor cambió de estado.

El sistema SCADA ha eliminado estos inconvenientes. Para encender y apagar motores no es necesario conocer su código. En las pantallas se puede determinar que motor se va a prender o a apagar según su ubicación en la planta. Sin embargo, si el operador no está seguro de cuál de las imágenes representa el motor deseado basta con hacer clic sobre la representación de un motor para obtener información. De una madera fácil y rápida se puede determinar su código y la máquina que acciona. Por otro lado, una vez realizada la operación se puede comprobar el nuevo estado del motor observando el color del interruptor colocado en las pantallas cerca de cada motor.

El manejo de las alarmas desde los tableros de control representaba un problema. Debido a la poca información que se puede mostrar en los paneles utilizados solo se podía visualizar un mensaje indicando la última alarma que se activó. Si se activaban varias alarmas era necesario solucionar uno de los problemas para poder conocer que otra alarma se había activado. Por este motivo no era posible solucionar varios problemas de manera simultánea. Esto ocasionaba demoras para el arranque de la planta y por lo tanto disminución de la producción.

El sistema SCADA permite observar todas las alarmas que se activan simultáneamente. Las pantallas de operación de las aplicaciones muestran todas las señales de alarmas activadas pudiendo así solucionar los problemas. Además, el

registro de las alarmas permite conocer los problemas acontecidos en horas anteriores para evitar que vuelva a suceder.

El programa de mantenimiento preventivo de Molino Electro Moderno S.A. se realiza periódicamente, cada cierto tiempo, sin contemplar las horas de funcionamiento de las máquinas. Gracias al registro de los eventos realizado por el sistema SCADA se puede conocer el número de horas de funcionamiento de los motores así como también el número de conmutaciones de las compuertas neumáticas. Gracias a esta información se podría implementar el mantenimiento preventivo considerando el trabajo realizado por cada máquina.

El personal de la planta de producción de Molino Electro Moderno S.A. está satisfecho con la implementación del sistema. Las ventajas que éste representa facilitan su trabajo y ha disminuido los tiempos requeridos para ciertas operaciones. A continuación se muestra una carta del Sr. Agustín Cueva, Jefe de Mantenimiento de esta empresa en la cuál manifiesta su satisfacción respecto a la implementación del sistema SCADA.

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El sistema SCADA implementado en la fábrica de Molino Electro Moderno S.A. cumple todos los objetivos planteados al inicio del proyecto.

- Las dos aplicaciones desarrolladas en Visual Basic permiten supervisar y controlar la maquinaria de la planta y por ende el proceso de producción de harina de trigo.

- La creación de dos aplicaciones permite una rápida visualización del estado de la maquinaria. En una aplicación se pueden controlar y supervisar las máquinas encargadas de la etapa de limpieza y en la otra las máquinas encargadas de la molienda.

- Las pantallas de interfase de operador permiten controlar y supervisar los siguientes procesos: manejo de los silos, acondicionamiento de materia prima, molienda del trigo y empaque de harina.

- Los botones colocados en las pantallas de interfase de operador permiten realizar desde el computador las siguientes operaciones: arranque automático del molino, parada de emergencia total y parcial, arranque y parada manual de todos los motores, y apertura y cierre de las compuertas neumáticas.

- El control de acceso implementado controla que solo personas autorizadas puedan ingresar a las aplicaciones del sistema SCADA. De este modo se evita que personal no autorizado realice operaciones no deseadas.

- El operador puede obtener rápidamente datos de los motores, compuertas y sensores a través del cuadro de información implementado en las aplicaciones.

- Los registros de los eventos y de las alarmas ocurridas pueden ser visualizados a través de dos ventanas diseñadas para este fin. Esta información es almacenada en la base de datos cuando se enciende o apaga un motor, se acciona una compuerta neumática, o se activa una alarma.

- Los resultados obtenidos luego de realizar pruebas al sistema SCADA fueron positivos. Todas las funciones implementadas trabajan del modo esperado.

- El protocolo OPC (OLE for Process Control) permitió realizar fácilmente la lectura y escritura de datos entre las aplicaciones desarrolladas en Visual Basic y los PLCs. Este protocolo además presenta la ventaja de que una vez adquirido el servidor OPC, éste se puede utilizar para desarrollar aplicaciones utilizando otros programas como por ejemplo LabView.

- El desarrollo de las aplicaciones para el sistema SCADA utilizando el software de programación Visual Basic resultó demorado y requirió varios meses para depurar completamente. Sin embargo, el hecho de que el costo de su licencia es muy inferior a la de programas como LabView o Wonderware Intouch representa una gran ventaja para el desarrollo de aplicaciones para sistemas SCADA.

- El empleo del programa PostgresSQL para la creación y mantenimiento de la base de datos permitió disminuir los costos de desarrollo del sistema. Este programa es completamente gratuito y ha trabajado muy bien durante todo el proceso de desarrollo y ejecución del sistema SCADA. El empleo de

PostgreSQL disminuyó el precio total del sistema manteniendo la funcionalidad que se tendría con un programa pagado.

- La implementación del sistema SCADA ha facilitado el trabajo del operador. El encendido y apagado manual de motores y otros actuadores es mucho más sencillo. Además, ahora se puede visualizar rápidamente el estado de la maquinaria de la planta. El cuadro de información también ha ayudado a los operadores ya que les permite identificar rápidamente el código y ubicación de los distintos actuadores y sensores.

5.2 RECOMENDACIONES

- En el proceso de producción de harina hay demasiado polvo en el lugar donde se encuentran ubicados los computadores en los que corren las aplicaciones del sistema SCADA, por lo que se recomienda a Molino Electro Moderno S.A. la adquisición de computadores industriales a prueba de polvo para evitar daños de estos equipos.

- Debido a que los operadores que utilizan el software del sistema SCADA poseen escasos conocimientos de computación se recomienda impartirles charlas sobre el uso de las funciones del software de modo que puedan utilizar las aplicaciones con mayor destreza.

- Debido a las características de los computadores utilizados las aplicaciones del sistema SCADA se demoran entre dos y tres minutos en arrancar. Por lo que se recomienda utilizar computadores más veloces para disminuir estos tiempos.

REFERENCIAS BIBLIOGRÁFICAS

[1] SHROYER, Tim. http://www.oznet.ksu.edu/aawf/July/july_3.htm

[2] Información proporcionada por Ing. Wilson Miranda, encargado de producción de Molino Electro Moderno S.A.

[3] SHROYER, Tim. http://www.oznet.ksu.edu/aawf/July/milling.htm

[4] SHROYER, Tim. http://www.oznet.ksu.edu/aawf/July/july_5.htm

[5] ANÓNIMO. http://www.optimate.com/products/optimate/1124.html

[6] ANÓNIMO. http://www.optimate.com/products/optimate/1224.html

[7] ANÓNIMO. http://www.optimate.com/products/optimate/1500.html

[8] ANÓNIMO. http://www.optimate.com/products/optimate/om9001.html

[9] ANÓNIMO. http://es.wikipedia.org/wiki/SCADA

[10] ANÓNIMO. http://www.automatas.org/redes/scadas.htm

[11] ANÓNIMO. http://en.wikipedia.org/wiki/RTU

[12] ANÓNIMO. http://www.alfinal.com/Temas/sistemascada.shtml

[13] ANÓNIMO. http://www.opcfoundation.org/Default.aspx/01_about/01_history .asp?MID=AboutOPC

[14] **MATTHEW**, Neil. **STONES**, Richard. Beginning Databases with PostgreSQL: From novice to professional. Segunda Edición. Apress. Estados Unidos. 2005.

[15] OPC Task Force. OPC Overview. Versión 1.0. 1998

[16] ANÓNIMO. http://www.catapultsoftware.com/products/scada/architecture .asp

[17] ANÓNIMO. http://web2.automationdirect.com/static/specs/directsoftcables.pdf

[18] **STEPHENS**, Ryan. **PLEW**, Ronald. **MORGAN**, Bryan. **PERKINS**, Jeff. Teach yourself SQL in 21 Days. Segunda Edición. Estados Unidos.

[19] **FORTE**, Stephen. Build a Database App with VB6. VBPJ. Publicación Noviembre. 1998.

[20] ANÓNIMO. http://vbasic.astalaweb.com/C_Grids/1_Grids.asp

[21] ANÓNIMO. http://www.recursosvisualbasic.com.ar/htm/ocx-componentes-activex-dll/57-dll-para-imprimir-flexgrid.htm

[21] ANÓNIMO. http://www.todoexpertos.com/categorias/tecnologia-internet/programacion/visual-basic/respuestas/491154/gifs-e-iconos-animados

[22] ANÓNIMO. http://www.vb-mundo.com/ocx.asp

[23] ANÓNIMO. http://www.recursosvisualbasic.com.ar/htm/trucos-codigofuente-visual-basic/28.htm

[24] ESCOBAR, Andrés. http://www.lawebdelprogramador.com/temas /enlace.php?idp=2680&id=93&texto=visual+basic

[25] ANÓNIMO. http://en.wikipedia.org/wiki/OLE_for_process_control

[26] ANÓNIMO. http://www.kepware.com/Menu_items/ industry_OPC_Foundation.html

[27] ANÓNIMO. http://web1.automationdirect.com/adc/Overview/Catalog /Software_Products/Directsoft_PLC_Programming_Software

[28] ANÓNIMO. http://www.issociate.de/board/post/316615 /How_To_Connect_PostgreSQL_thru_VB_without_DSN.html

[29] ANÓNIMO. http://www.postgresql.org/

[30] VOON KIONG, Liew. http://www.vbtutor.net/vbtutor.html

[31] ANÓNIMO. http://www.dbforums.com/showthread.php?p=6298549

[32] ANÓNIMO. http://archives.devshed.com/forums/databases-124/vb-odbc-connection-to-postgresql-dao-ado-rdo-dsn-197317.html

[33] ANÓNIMO. DirectSoft Programming Software User Manual.

[34] ANÓNIMO. DL405 User Manual 3rd Edition.

[35] ANÓNIMO. DirectSoft DDE Server User Manual.