



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño de unidad de control
modular para automatización
de procesos**

TESIS

Que para obtener el título de
Ingeniero Mecatrónico

P R E S E N T A

Antonio Fonseca Alvarado

DIRECTOR DE TESIS

M.I. Serafín Castañeda Cedeño



Ciudad Universitaria, Cd. Mx., 2016

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México y la Facultad de Ingeniería quienes dan todo sin pedir nada a cambio.

A los profesores que durante los 5 años de carrera inspiraron y orientaron mi formación; menciono de forma especial al MI. Ulises Peñuelas R., al Dr. Víctor G. Villela, a la MI. Arianna Sánchez G. y al MI. Noé Martínez S.

A los miembros del grupo de investigación en mecatrónica, MRG, de la DIMEI bajo la dirección del Dr. Víctor J. González Villela, en especial al MI. Noé Martínez Sánchez, al Dr. Octavio Díaz H. y al MI. Erick Peña Medina, por sus enseñanzas dentro y fuera del salón de clases que representaron una motivación para el desarrollo de esta tesis.

En forma más personal en cuanto a amistad, a Axel Zurita Lozano, quien desde preparatoria me ha acompañado como colega, compañero y amigo tanto en competencia como en colaboración. Menciono de forma especial a Diana Santa María G., Ruth Isela Barrera R., Laura Recillas, Juan Sidar, a Néstor Rivera y a Mariano Méndez por su amistad y acompañamiento en diversos momentos.

Menciono nuevamente a la MI. Arianna Sánchez, a quien se debe que esta tesis fuese desarrollada, sus enseñanzas como profesora y como directora de tesis complementaron y motivaron de gran forma mi formación; le dedico este logro y quedo agradecido con ella. De igual forma a la empresa SNF de México con la que se colaboró en el desarrollo de su proyecto, que permitió el desarrollo libre de la unidad de control y del sistema mecánico.

Finalmente, a la familia por su apoyo incondicional.

ÍNDICE

Agradecimientos	1
Índice.....	2
Índice tablas	4
Índice de Figuras	6
Introducción.....	10
Cap. 1 Generalidades	11
1.1 Antecedentes de proyecto	11
1.2 Objetivo	12
1.2.1 Objetivos específicos	12
1.3 Alcances	12
1.4 Justificación	12
1.5 Metodología	13
Cap. 2 Marco teórico.....	15
2.1 Efluentes industriales	15
2.1.1 Prueba de jarras.....	15
2.1.2 Método de mezclado en Prueba de jarras.....	16
2.2 Estructura de procesos	16
2.2.1 Procesos discretos	18
2.2.2 Procedimientos de operación.....	19
2.3 Estructura modular de procesos	20
2.3.1 Jerarquía de sistemas basada en componentes	22
2.3.2 Grafcet de procesos.....	25
2.4 Programación de autómatas	28
2.4.1 Estructura de algoritmos	30
2.4.1.1 Paradigma en microcontroladores.....	30
2.4.1.2 Programación orientada por eventos	31
2.4.2 Bloques funcionales.....	33
2.4.3 Estructura de la unidad de control	35
2.4.4 Modelo BITBUS de comunicación	37
Cap. 3 Diseño conceptual.....	39
3.1 Estado de proyecto.....	39
3.1.1 Planeación de trabajo	42
3.1.2 Definición de Fases de trabajo.....	44
3.2 Necesidades y requerimientos	46
3.3 Especificaciones y limitaciones.....	47
3.4 Despliegue de matriz de calidad	50
3.5 Definición del tipo de sistema modular	52
Cap. 4 Diseño de módulos.....	57
4.1 Jerarquía de operaciones de proceso	59
4.1.1 Arquitectura del proceso	60
4.1.2 Operaciones del proceso	67
4.1.3 Jerarquía de componentes	69
4.1.4 Estructura de operaciones.....	81
4.1.5 Cartas de función	95
4.1.6 GRAFCET de operaciones.....	107
4.2 Modularidad del proceso.....	116
4.2.1 Módulos.....	117
4.2.1.1 Especificaciones y limitaciones	118
4.2.1.2 Pines de control y comunicación	120

4.2.2	Estructura de comunicación de módulos	124
4.2.3	Estructura de funciones de operación en la UC	143
4.3	Fase 1: Control de estructura	153
4.3.1	Módulo de estructura	153
4.3.2	Bloques funcionales	155
4.3.3	Sistema electrónico	157
4.4	Interfaz de usuario y módulo maestro	161
4.4.1	Módulos	161
4.4.2	Sistemas electrónicos	164
4.5	Fase 2: Control de batidora	168
4.5.1	Módulo control batidora	168
4.5.2	Bloques funcionales	173
4.5.3	Sistema electrónico	174
4.6	Fase 3: Dosificación floculante y efluente	177
4.6.1	Módulo de dosificación del efluente	177
4.6.2	Módulo de dosificación del floculante	179
4.6.3	Bloques funcionales	183
4.6.4	Sistemas electrónicos	186
Cap. 5	Prueba de modelo funcional	190
5.1	Prueba de módulos	190
5.1.1	Sensores	191
5.1.2	Actuadores	194
5.1.3	Ejecución de tareas	196
5.2	Prueba de comunicación y de operaciones	199
5.3	Prueba de secuencia	205
5.4	Resultados y análisis de pruebas	212
Cap. 6	Conclusiones	219
6.1	Trabajo a futuro	220
	Bibliografía	221
	Apéndices	222
A.	Fundamentación de Proyecto Lodos	223
B.	Estructura mecánica, Proyecto Lodos	231
C.	Interfaz de usuario, Proyecto Lodos	244
	Anexos	260
1.	HERRAMIENTA GRAFCET	260

ÍNDICE TABLAS

Tabla 1. Clasificación del índice de Willcomb	16
Tabla 2. Tipo de POUs	22
Tabla 3. Comparativa entre sistemas de control y cableado	28
Tabla 4. Planteamiento del Proyecto Lodos	43
Tabla 5. Distribución sistemas por cada etapa	44
Tabla 6. Prioridad de las necesidades.....	46
Tabla 7. Distribución de Fases de trabajo	47
Tabla 8. Especificaciones	48
Tabla 9. Limitaciones	48
Tabla 10. Grado de restricción de las especificaciones.....	49
Tabla 11. Especificaciones ponderadas por la matriz de calidad	50
Tabla 12. Especificaciones agrupadas por jerarquía	52
Tabla 13. Concepto inicial de la unidad de control	53
Tabla 14. Relación de las entradas entre el proceso y el sistema mecánico.....	65
Tabla 15. Relación de las posiciones en estructura mecánica con el proceso	65
Tabla 16. Arquitectura del proceso	67
Tabla 17. FCB de la Fase 1, estructura.....	72
Tabla 18. FCB de la Fase 2, Batidora.....	73
Tabla 19. FBC de la Fase 3, dosificación.....	74
Tabla 20. Dependencias de los FBC de las Fases de trabajo	76
Tabla 21. Niveles de jerarquía de los FCB.....	79
Tabla 22. Estados de inicio de sistemas	81
Tabla 23. Relación de los grupos de operación de la UC y del proceso	82
Tabla 24. Componentes de función	82
Tabla 25. Funciones de operación de ejecución.....	83
Tabla 26. Funciones de operación de planeación y configuración.....	84
Tabla 27. Funciones de grupo de operación en UC	85
Tabla 28. Tipos de variables en las funciones de módulo	95
Tabla 29. Variables y funciones de módulo 1	95
Tabla 30. Variables y funciones de módulo 2	96
Tabla 31. Variables y funciones de módulo 3	96
Tabla 32. Variables y funciones de módulo 4	96
Tabla 33. SFC sistema 1a de módulo 1	97
Tabla 34. SFC sistema 1b y 1c de módulo 1	98
Tabla 35. SFC sistema 1d de módulo 1	100
Tabla 36. SFC sistema 2a de módulo 2	101
Tabla 37. SFC sistema 3a de módulo 3	103
Tabla 38. SFC sistema 3b1 de módulo 4.....	103
Tabla 39. SFC sistema 3b2 de módulo 4.....	106
Tabla 40. Variables y funciones de módulo maestro interfaz	110
Tabla 41. Especificaciones iniciales del sistema de control.....	118
Tabla 42. Especificaciones iniciales de actuadores y sensores	118
Tabla 43. Definición de los microcontroladores por limitaciones	119
Tabla 44. Comparación de microcontroladores	119
Tabla 45. Funciones de microcontrolador PIC para las características de módulo.....	120
Tabla 46. Puertos del microcontrolador PIC16F887	122
Tabla 47. Pines de I/O para componentes de ejecución	122

Tabla 48. Pines de control para módulos	123
Tabla 49. Pines de comunicación para módulos	124
Tabla 50. Bloques de Bitbus en los módulos	125
Tabla 51. Estructura de comunicación	129
Tabla 52. Parámetros en datos de comunicación	131
Tabla 53. Set de instrucciones de la estructura de comunicación	141
Tabla 54. Instrucciones de operaciones: Conexión	142
Tabla 55. Instrucciones de operaciones: Configuración parámetros	142
Tabla 56. Instrucciones de operaciones: Secuencia	143
Tabla 57. Módulo estructura	154
Tabla 58. Componentes del sistema electrónico, Módulo 1.....	158
Tabla 59. Módulo interfaz y maestro	162
Tabla 60. Componentes de módulo de interfaz y de maestro	164
Tabla 61. Módulo batidora	168
Tabla 62. Parámetros y constantes de control PID por ZN	169
Tabla 63. Componentes del sistema electrónico, Módulo 2.....	175
Tabla 64. Módulo efluente	178
Tabla 65. Pruebas de dosificación para caracterización de encóder	180
Tabla 66. Pruebas de dosificación por ecuaciones de regresión	181
Tabla 67. Módulo floculante	182
Tabla 68. Componentes del sistema electrónico, Módulo 3 y 4.....	186
Tabla 68. Resultados pruebas de acondicionado sensores	213
Tabla 68. Resultados pruebas de error en dosificación y mezclado.....	213

ÍNDICE DE FIGURAS

Figura 1. Modularidad de un proceso.....	17
Figura 2. Estructura general de procesos [10].....	18
Figura 3. Ejemplo de una SFC	21
Figura 4. Esquema genérico del proceso.....	23
Figura 5. Niveles de jerarquía de un proceso	24
Figura 6. FCB genérico	25
Figura 7. Estructura de sistema automatizado.....	27
Figura 8. Unidad de control.....	29
Figura 9. Representación del paradigma deliberativo	31
Figura 10. Control de flujo de datos.....	32
Figura 11. Representación del paradigma dirigido por eventos	32
Figura 12. Despachador de eventos.....	33
Figura 13. Bloque funcional.....	35
Figura 14. Componentes de una planta [36].....	36
Figura 15. Estructura unidad de control	36
Figura 16. Arquitectura BITBUS	37
Figura 17. Ciclo del proceso de prueba de jarras	39
Figura 18. Algoritmo del proceso de prueba de jarras, Apéndice A	39
Figura 19. Distribución de sistema mecánico propuesto, Apéndice A	40
Figura 20. Etapa 1 del proyecto, sistema mecánico, CAD proyecto LODOS, estructura carrusel, diciembre 2013	41
Figura 21. Organización del equipo de trabajo	42
Figura 22. Fases de trabajo.....	45
Figura 23. Matriz de calidad.....	51
Figura 24. Concepto de la estructura de comunicación.....	53
Figura 25. Concepto de estructura de organización de la unidad de control.....	54
Figura 26. Ventaja de unidad de control modular	55
Figura 27. Ciclo de operación en laboratorio y en sistema mecánico	59
Figura 28. Prueba de Jarras	60
Figura 29. Floculante en proceso primario de tratamiento de efluentes	61
Figura 30. Proceso completo del proyecto LODOS.....	62
Figura 31. Estructura mecánica, primera etapa, Apéndice A	64
Figura 32. Distribución de posiciones y del ciclo de operación.....	66
Figura 33. Distribución de sistemas por posición	66
Figura 34. Grupos de operación de un proceso	68
Figura 35. Esquema general de componentes y su distribución, Apéndice B.....	70
Figura 36. Esquema sistema carrusel	70
Figura 37. Esquema sistema orientación de contenedores	71
Figura 38. Esquema sistema posicionamiento batidora	71
Figura 39. Esquema sistema de mezclado.....	72
Figura 40. Esquema sistema dosificación efluente	73
Figura 41. Esquema sistema dosificación floculante	74
Figura 42. Funciones de FBC en Estructura.....	77
Figura 43. Funciones de FBC en Mezclado	78
Figura 44. Funciones de FBC en Dosificación.....	78
Figura 45. Jerarquía de funciones de FBC	80
Figura 46. Estructura de operaciones en UC.....	81

Figura 47. Tiempos de operación: Preparación.....	88
Figura 48. Tiempos de operación: Puesta en Marcha	89
Figura 49. Tiempos de operación: Puesta en Marcha, 2	90
Figura 50. Tiempos de operación: Puesta en Marcha, 3	91
Figura 51. Tiempos de operación: Fin de marcha	92
Figura 52. Tiempos de operación: Fin por error	93
Figura 53. Tiempos de operación: Puesta en marcha, 1	94
Figura 54. Comparación del proceso secuencial con los tiempos de operación	94
Figura 55. SFC 1, sistema carrusel de módulo 1	97
Figura 56. SFC 2, sistema orientación de módulo 1	99
Figura 57. SFC 3, sistema posición batidora de módulo 1	100
Figura 58. SFC 4, sistema mezclado de módulo 2	102
Figura 59. SFC 5, sistema dosificado efluente de módulo 3	103
Figura 60. SFC 6, sistema dosificado vías de módulo 4	105
Figura 61. SFC 7, sistema dosificado floculante de módulo 4	107
Figura 62. Jerarquía de funciones de operación	108
Figura 63. SFC de Módulo, configuración de ejecución d tarea	110
Figura 64. SFC de Módulo, solicitud de tarea.....	111
Figura 65. Grafcet encapsulado 1, Preparación sistema.....	112
Figura 66. Grafcet encapsulado 2, Puesta en marcha.....	113
Figura 67. Grafcet encapsulado 3, Fin de operación	114
Figura 68. Grafcet encapsulado 4, Fin de operación por error.....	115
Figura 69. Diagrama de pines del microcontrolador 16F887, datasheet Microchip	121
Figura 70. Estructura Bitbus de la Unidad de Control	125
Figura 71. Estructura de datos para comunicación RS232.....	126
Figura 72. Estructura de datos para comunicación I2C.....	128
Figura 73. Conversión de datos, operaciones de comunicación.....	133
Figura 74. Identificación de errores, operaciones de comunicación	134
Figura 75. Funciones de emisor-receptor para RS232, estructura comunicación	135
Figura 76. Función de emisor para I2C, estructura comunicación.....	136
Figura 77. Función de receptor para I2C, estructura comunicación	137
Figura 78. Estructura de módulos de operaciones de planeación y ejecución	138
Figura 79. Estructura de módulos de operaciones de configuración.....	139
Figura 80. Estructura del sistema de control modular.....	140
Figura 81. Función de operación en UC: manejador eventos	148
Figura 82. Función de operación en UC: Sensado para manejador eventos	149
Figura 83. Función de operación en UC: Funciones de eventos y sensado	150
Figura 84. Función de operación en UC: Paradigma deliberativo, planeación	151
Figura 85. Función de operación en UC: Paradigma deliberativo, actuación.....	152
Figura 86. Estructura de módulo 1	154
Figura 87. Módulo 1, Bloques funcionales de SFC 1 y SFC 3	155
Figura 88. Módulo 1, Bloque funcional de SFC 2	156
Figura 89. Módulo 1, sistema electrónico, etapa acondicionamiento	159
Figura 90. Módulo 1, sistema electrónico, etapa potencia	160
Figura 91. Módulo 1, sistema electrónico, etapa control	160
Figura 92. Estructura módulo interfaz.....	163
Figura 93. Estructura módulo maestro	163
Figura 94. Interfaz, sistema electronico general.....	165
Figura 95. Módulo maestro, acondicionamiento de lienas de comunicación	166

Figura 96. Módulo maestro, conectores para líneas de comunicación y varios	166
Figura 97. Módulo maestro, componentes de la etapa de control	167
Figura 98. Estructura de Módulo 2.....	169
Figura 99. Módulo 2, prueba de respuesta al escalón, ZN	170
Figura 100. Algoritmo para control PID en microcontroladores	170
Figura 101. Pruebas de control.....	171
Figura 102. Comparación del arranque del control por ganancias tabuladas.....	172
Figura 103. Error en estado permanente	172
Figura 104. Módulo 2, Bloque funcional, SFC 4	173
Figura 105. Lectura de encoder a 300rpm.....	174
Figura 106. Módulo 2, sistema electrónico, etapa acondicionamiento	175
Figura 107. Módulo 2, sistema electrónico, etapa potencia	176
Figura 108. Módulo 2, sistema electrónico, etapa de control.....	176
Figura 109. Estructura de Módulo 3.....	178
Figura 110. Diagrama de evolución sensor ultrasónico HCSR04, datasheet fabricante Cytron technologies, 2016.....	179
Figura 111. Pseudo código para filtrado por promedio, sin picos.....	179
Figura 112. Sensor propuesto para bomba de dosificación efluente	180
Figura 113. Ecuaciones por regresión para caracterización de sensor	181
Figura 114. Estructura módulo 4.....	182
Figura 115. Módulo 3, Bloque funcional, SFC 5	183
Figura 116. Módulo 4, Bloque funcional, SFC 6	184
Figura 117. Módulo 4, Bloque funcional, SFC 7	185
Figura 118. Módulo 4, sistema electrónico, etapa de control.....	187
Figura 119. Módulo 4, sistema electrónico, etapa de acondicionamiento	188
Figura 120. Módulo 4, sistema electrónico, etapa de potencia	188
Figura 121. Módulo 3, sistema electrónico, etapa de potencia	189
Figura 122. Módulo 3, sistema electrónico, etapa de control.....	189
Figura 123. Sistema electrónico de la unidad de control, general.....	190
Figura 124. Módulo 1, placas de acondicionamientos	191
Figura 125. Módulo 1, pruebas de sensado	192
Figura 126. Módulo 3, pruebas de sensado	192
Figura 127. Módulo 3, pruebas de lectura del sensor ultrasónico	193
Figura 128. Módulo 4, pruebas de sensado	193
Figura 129. Distribución de fuentes y placas de acondicionamiento de potencia.....	194
Figura 130. Módulo 1, pruebas de actuadores.....	194
Figura 131. Módulo 2, prueba de actuador	195
Figura 132. Módulo 3, prueba de actuadores	195
Figura 133. Módulo 4, prueba de actuadores	196
Figura 134. Unidad de control y estructura mecánica.....	196
Figura 135. Módulo 1, ejecución de tarea en estructura	197
Figura 136. Ejecución de tarea, posicionamiento y mezclado	197
Figura 137. Ejecución de tarea, posicionamiento batidora y orientación contenedor.....	197
Figura 138. Ejecución de tarea, dosificación de floculante.....	198
Figura 139. Ejecución de tarea, mezclado.....	198
Figura 140. Unidad de control y sistema mecánico en la última Fase de trabajo	199
Figura 141. Estructura de comunicación, simulación de solicitud id de esclavos	200
Figura 142. Estructura de comunicación, prueba de solicitud envío de datos	201
Figura 143. Estructura de comunicación, prueba solicitud tareas	202

Figura 144. Estructura de comunicación, prueba de detección de errores, fin normal	203
Figura 145. Prueba de error, detección de dispositivos.....	203
Figura 146. Prueba de error, solicitud de tarea.....	203
Figura 147. Prueba de error, detección de error y solicitud de paro	204
Figura 148. Prueba de error, verificación de dispositivos.....	204
Figura 149. Estructura de comunicación, detección de errores, final por error	205
Figura 150. Prueba de interfaz, configuración de parametros.....	206
Figura 151. Prueba de interfaz, menu de datos	207
Figura 152. Pruebna de interfaz, ingreso de datos	207
Figura 153. Prueba de interfaz, confirmación de dato	208
Figura 154. Prueba de interfaz, solicitud de secuencia.	208
Figura 155. Prueba secuencia, TO1 dosificación lodo	209
Figura 156. Prueba secuencia, TO2 movimiento contenedor	209
Figura 157. Prueba secuencia, TO3 dosificación y posicionamiento batidora	210
Figura 158. Prueba secuencia, TO4 mezclado	210
Figura 159. Prueba secuencia, TO4 posicionamiento batidora	211
Figura 160. Prueba secuencia, TO4 cambio orientación para recolección muestra.....	211
Figura 161. Prueba secuencia, TO4 orientación para dosificación efluente	212
Figura 162. Detección de errores	217

INTRODUCCIÓN

La automatización modular conlleva el estudio de las operaciones de los procesos, de sus componentes físicos y de las capacidades de los autómatas programables para cubrir las necesidades y especificaciones de dichas operaciones en el ambiente de trabajo. El concepto modular en un sistema de control define un orden de jerarquía y de organización de elementos particulares llamados módulos con funciones de operación bien definidas e independientes, que trabajan en conjunto con otros bajo un planeador quien se encarga de organizar y establecer tiempos de operación para el desarrollo de un proceso. La implementación de sistemas modulares incrementa el número de elementos de control, establece la necesidad de una organización bien estructurada y eleva costos en cuanto al procesamiento, pero permite reducción de tiempos, flexibilidad del proceso, optimiza la ejecución de operaciones y permite el monitoreo de estados de cada módulo para su mantenimiento y configuración.

El presente trabajo desarrolla una metodología basada en el diseño del producto y en análisis de los componentes de un proceso para el diseño de una unidad de control modular por microcontroladores; se define la separación de las funciones de operaciones del proceso con las del sistema modular para permitir desarrollar un modelo genérico de módulo para la gestión de operaciones de configuración, de planeación y de ejecución, esto permite agregar módulos sin tener que definir estructuras particulares o modificaciones globales para el sistema, dependiendo la escalabilidad únicamente de del análisis de componentes y funciones del proceso.

El método se aplica en el diseño de la unidad de control para el proyecto **LODOS** el cual se basa en pruebas de mezclado de polímeros para el pretratamiento de aguas residuales en un sistema mecánico que simula el proceso en laboratorio y en planta. La orientación modular para la unidad de control del proyecto se establece a partir de la evaluación de la automatización por un sistema secuencial centrado en un microcontrolador que fue propuesto al inicio del proyecto, se concluye que no cumple con los objetivos y las operaciones del proceso superan las capacidades del sistema. La evolución de la metodología inicia con el análisis de funciones de operación para saber “qué” automatizar; luego se establece la estructura modular de la unidad de control para definir el “cómo”; finalmente se implementa en cada módulo las funciones de operación del proceso.

CAP. 1 GENERALIDADES

1.1 Antecedentes de proyecto

El tratamiento de aguas residuales provenientes del sector industrial consiste en una serie de procesos físicos, químicos y biológicos que tienen como fin separar y reducir los contaminantes y residuos sólidos del efluente líquido, también llamados biosólidos o lodos, para su disposición en el ambiente u otros procesos en la misma industria. Actualmente los esfuerzos para tratar las aguas residuales industriales están sujetos a regulaciones y estándares gubernamentales debido al daño en el ambiente y la salud, donde la legislación establece límites de seguridad y normas para el desecho de los efluentes líquidos.

La empresa SNF FLOERGER DE MÉXICO se especializa en la fabricación y comercialización de floculantes y coagulantes orgánicos que se usan en procesos de separación de sólidos en efluentes, residuos industriales líquidos. Los servicios que brinda son de asesoría técnica y selección y adaptación de productos [1]. Las aplicaciones de sus servicios incluyen:

- Clarificación.
- Separación solido-líquida en tratamiento de efluentes.
- Ayudantes de flotación.
- Remoción de metales pesados.
- Procesos de separación en minería.
- Superabsorbentes de uso agrícola, entre otros.

Sobre el tratamiento de efluentes, su principal mercado, sus polímeros se adaptan a todos los tipos de tratamiento de aguas, desde la producción de agua potable hasta el proceso de desaguado [2]. El proceso se basa en la floculación de partículas mezcladas en líquidos que por sus características no se pueden decantar, estas son inducidas a la aglomeración [3].

Debido a la diversidad y particularidad que presentan los efluentes, actualmente determina el mejor tipo de floculante mediante la prueba de jarras, que es una serie de pasos para probar en laboratorio diferentes concentraciones de floculantes para una misma muestra de efluente. En dicho proceso se requiere una gran cantidad de pruebas repetitivas debido a la diversidad de los floculantes y a la característica particular de cada planta de tratamiento, lo cual implica costos de tiempo y de recursos humanos.

La empresa ha considerado la necesidad de automatizar la prueba de sus floculantes, además de unirla con el proceso de planta para obtener un análisis más completo. Como parte de sus servicios, su departamento de ingeniería diseña y fabrica equipos para la aplicación en sitio de sus productos y para la adaptación de sus equipos en las condiciones que el cliente requiere. De este departamento fue el que extiende el proyecto “**LODOS**”.

1.2 Objetivo

Desarrollar una unidad de control modular para la automatización del proceso de mezclado de floculantes con efluentes del proyecto LODOS, Apéndice A, etapas 1, 2 y 3.

1.2.1 Objetivos específicos

- Probar la modularidad en sistemas de control para automatización de procesos.
- Diseñar e implementar un modelo funcional de una unidad de control modular, que permita la flexibilidad del proceso, la comunicación y la configuración independiente de cada módulo.
- Diseñar un sistema de comunicación para los módulos que permita la organización de las solicitudes de operación de datos y de tareas, la configuración, el monitoreo de estados y la identificación de errores.
- Diseñar una estructura de módulo genérico que pueda ser implementado en otros sistemas basados en microcontroladores.
- Diseñar e implementar una interfaz intuitiva para el usuario para las operaciones de configuración del sistema.

1.3 Alcances

El presente trabajo pretende automatizar parte de un procedimiento de laboratorio mediante el concepto de módulos para la unidad de control. El diseño se basa análisis de operaciones para descomposición en funciones tanto a nivel de módulo como de proceso. Para la implementación no se participa en el diseño mecánico, por lo que se desarrolla únicamente la electrónica digital y la estructuración de modelos y algoritmos para los módulos. El modelo funcional demuestra que los métodos de automatización orientados al proceso y a la estructuración modular permiten orden, flexibilidad y escalabilidad en comparación con sistemas secuenciales centrados en un solo procesador.

En cuanto a la implementación, no se pretende desarrollar sistemas electrónicos especializados debido a que los actuadores y sensores usados no requieren una etapa de acondicionamiento complejo; para las etapas posteriores del proyecto se planea agregar más módulos por lo que el establecimiento del sistema genérico debe permitir únicamente analizar las operaciones de los nuevos procesos sin alterar a la estructura del sistema modular.

1.4 Justificación

La industria es uno de los motores principales para el desarrollo económico y clave para el progreso social, la creciente demanda de los diversos sectores de consumo ha implicado que aumente la producción para cubrir el mercado, lo que ha implicado un aumento drástico en la contaminación ambiental por los desechos que se generan en los procesos industriales. El problema radica en cómo neutralizar los desechos, su tratamiento o almacenamiento sin afectar al medio ambiente. Entre los principales desechos destaca los que generan sustancias líquidas, debido a que una pequeña porción puede contaminar grandes cantidades de agua, siendo la causa principal de la contaminación de mantos acuíferos y ríos cercanos a las industrias.

No obstante, con la concientización social y gubernamental sobre el impacto ambiental que tienen las industrias, además del auge de nuevas tecnologías, se han creado nuevos sectores comerciales que desarrollan investigaciones para poder generar productos que reducen los contaminantes industriales, por ejemplo, la empresa SNF FLOERGER DE MÉXICO da servicios para el tratamiento de efluentes que permiten a un sector de la industria y al gubernamental reducir niveles de contaminación en las aguas residuales.

Por otra parte, la automatización de procesos pretende mejorar la calidad del trabajo ante operaciones repetitivas y aquellas que representa un factor de riesgo en la integridad del operador. La aplicación de sistemas modulares permite además procesos flexibles que reducen el tiempo de las operaciones y el escalamiento. En este sentido, en colaboración con la empresa y como parte de la responsabilidad social y la ética de la ingeniería en el entorno social-ambiental, el desarrollo de esta tesis se basa en la necesidad de realizar las pruebas de laboratorio de forma automatizada, con el fin de tener un ambiente controlado para determinar el mejor polímero para los efluentes y así poder contribuir a reducir el impacto ambiental.

1.5 Metodología

El propósito del trabajo basado en un método es la normalización de los procedimientos para construir y evaluar un diseño al clarificar las etapas, procesos e información que se abordan en la solución de una necesidad, de modo que se enfatice la dimensión estratégica del proceso de unificación e integración de ideas en términos de objetivos, prioridades y programas de acción.

De la diversidad de metodologías aplicadas en las ciencias de ingeniería para proyectos se opta por utilizar dos metodologías para orientar el diseño de la unidad de control en automatización de procesos. La base es la metodología de Ulrich sobre el *Diseño y Desarrollo de Productos*, para considerar el proyecto como un producto que engloba funciones de planeación, toma de decisiones y optimización de procesos en sus diferentes niveles de organización; por otra parte, se complementa mediante la metodología de N. Iriondo y F. Artaza sobre el *Diseño de Sistemas de Control Basado en Jerarquía de Componentes*, para darle un sentido jerárquico a las funciones de operación del proceso y un sentido modular a la unidad de control. El procedimiento de desarrollo que se plantea se divide sistemáticamente en tres etapas de diseño: conceptualización de problema, diseño de concepto y validación.

La primera etapa se enfoca en el análisis del proyecto y del proceso a automatizar para determinar las necesidades y especificaciones del sistema de control, así como la planeación del desarrollo en Fases de trabajo, tomando en cuenta las especificaciones iniciales antes del desarrollo. Siguiendo la metodología del producto, se analiza primero la estructura sistemática del proceso para determinar las necesidades de la problemática en relación al desarrollo de la unidad de control, y de esta forma determinar las especificaciones y requerimientos para conceptualizar las posibles soluciones con una orientación de modularidad en los procesos. Posteriormente se filtran las especificaciones por una matriz de calidad para darles una estructura y un peso en el desarrollo de la estructura de la unidad de control, tomando en cuenta el sistema mecánico.

Para la etapa de diseño de concepto, por los objetivos del proyecto en cuanto a la automatización del sistema mecánico, se divide el desarrollo de diseño en tres partes: la primera se basa en el análisis de componentes para establecer las funciones de operación en el proceso, a modo de identificar las condiciones, estados y jerarquías que intervienen en el desarrollo de su operación, de esta forma se

establece un modelo de análisis de funciones de operación para el desarrollo de las siguientes etapas; la segunda parte se enfoca en el desarrollo de una estructura de módulo genérica para las operaciones de gestión sin depender de una estructura particular por las operaciones del proceso, también se define y orienta para microcontroladores; finalmente la tercera parte se enfoca en la implementación de los sistemas en módulos, siendo consecuencia de las dos partes anteriores. La distinción de funciones del proceso y del módulo permite que la implementación de nuevos módulos solo dependa del análisis de su proceso.

Finalmente, la etapa de validación es para realización de pruebas del modelo funcional para la ejecución de las operaciones del proceso considerando la estructura modular, el sistema físico y la relación de las especificaciones y restricciones para poder validar la solución y permitir la continuidad del desarrollo de las demás etapas del proyecto.

CAP. 2 MARCO TEÓRICO

2.1 Efluentes industriales

El término agua residual o *efluente* define un tipo de agua que está contaminada con sustancias químicas o elementos orgánicos que proceden de desechos en algún proceso. Debido a su composición reciben una gran importancia tal que se requiere sistemas complejos de canalización, tratamiento y desalojo. Su tratamiento nulo o indebido genera graves problemas de contaminación y de salud [4].

Se entiende como tratamiento físico-químico a aquellos en los que se agregan dosis de sustancias coagulantes y floculantes con la finalidad de disminuir la contaminación. En general estos tratamientos se emplean como etapa preliminar al tratamiento biológico debido a que algunos compuestos pueden ser tóxicos para las bacterias.

Por su origen, las aguas residuales presentan en su composición diferentes elementos orgánicos e inorgánicos que se pueden resumir de forma general como:

- Componentes suspendidos: Gruesos y finos que pueden ser filtrados fácilmente.
- Partículas coloidales: Sólidos suspendidos con sedimentación lenta.
- Componentes disueltos: Inorgánicos y orgánicos que cambian la química de la sustancia.

En función de estos componentes se diseñan, construyen, operan y mantienen sistemas de tratamiento de aguas residuales para recuperar las características del agua para uso posterior al tratamiento. La cuantificación de los componentes mencionados marca condiciones para definir una estrategia adecuada que resuelva el tratamiento económicamente, obteniendo así una calidad de agua residual con posibilidad de uso posterior.

Es importante conducir pruebas piloto paralelamente a la caracterización de las aguas residuales por su contenido y variedad en las especies químicas solubles. Esto puede dar límites de tratabilidad estándar dependiendo de las características del sistema evaluado [5].

2.1.1 Prueba de jarras

Para poder determinar la tratabilidad de los efluentes se recurre a un método de simulación de los procesos de coagulación y de floculación en planta llamada *prueba de jarras*, la cual se realiza a nivel de laboratorio como modelo predictivo; los flóculos formados con diferentes dosis del coagulante dan como resultado valores de turbiedad diferentes con lo que se puede analizar y estimar la concentración que permita obtener agua de buena calidad, fácilmente separable por decantación.

Con los aportes tecnológicos, se ha perfeccionado continuamente el ensayo y se ha aplicado con éxito a diversas condiciones de coagulación de las aguas. El modelo de prueba de jarras se ha orientado principalmente hacia la caracterización y optimización de las variables químicas del proceso de coagulación [6].

2.1.2 Método de mezclado en Prueba de jarras

El procedimiento experimental para la prueba de jarras, sin considerar el análisis de los resultados, está formado por las siguientes etapas [7]:

- **Caracterización de la muestra patrón:** Se determina la turbiedad, pH y color del agua cruda, sin tratar, a la entrada de la planta de tratamiento. Este es el “blanco” o patrón de comprobación.
- **Pruebas de jarras para la selección del mejor coagulante:** Para efectos de comparación, esta prueba permite determinar el mejor coagulante, que es el que reduce en mayor porcentaje los niveles de turbidez y color del agua cuando es combinado con una dosis constante de floculante.
- **Prueba de jarras para el ajuste de la dosis del mejor coagulante:** Con el mejor coagulante se realiza una nueva serie de pruebas de jarras, variando su dosis y utilizando nuevamente el floculante con dosis constante. La finalidad es determinar la mejor dosis del coagulante seleccionado en la prueba anterior.
- **Pruebas de jarras para la determinación de la mejor dosis de floculante:** Con la mejor dosis de coagulante, se procede a simular la Fase de floculación agregando diversas dosis de polímero floculante. El proceso se repite para hasta obtener los flóculos más grandes, consistentes y de mayor velocidad de asentamiento o precipitación.

Se miden los parámetros de turbiedad, pH, color, por el tiempo de formación del *floc*¹ y la evaluación de los resultados se determina de acuerdo al índice de Willcomb [8], Tabla 1, que califica el tamaño visual de los residuos visibles suspendidos en un líquido. Al finalizar esta prueba se pueden cuantificar los polímeros coagulantes y floculantes ideales para la clarificación del agua.

Tabla 1. Clasificación del índice de Willcomb	
Calificador	Índice de Willcomb
0	Floc coloidal. Ningún signo de aglutinación
2	Visible. Floc muy pequeño, casi imperceptible
4	Disperso. Floc bien formado pero uniformemente distribuido (sedimenta muy lentamente o no sedimenta)
6	Claro. Floc de tamaño relativamente grande pero que precipita con lentitud
8	Bueno. Floc que deposita fácilmente pero no completamente
10	Excelente. Floc que deposita fácil y completamente

2.2 Estructura de procesos

La automatización, en los sistemas productivos, se define como una tecnología relacionada con las aplicaciones de sistemas mecánicos y eléctricos integrados con sistemas computarizados para operación y control de la producción. Hoy en día, la automatización se basa en la electrónica

¹ Floc es una masa floculante formado en un fluido a través de la precipitación o agregación de partículas en suspensión. Los flóculos producidos a partir de aguas duras/turbias son generalmente densos

programable y flexible, a diferencia de antes del advenimiento de la electrónica, que era fija por que se basaba en componentes y sistemas mecánicos [8].

Para poder alcanzar un equilibrio entre la repetitividad de las operaciones y el máximo aprovechamiento de los equipos, existen diferentes opciones de las cuales destacan la fabricación modular y la aplicación de los principios de la tecnología de grupo [9]. La fabricación modular consiste en buscar la especialización de la producción en la fragmentación de sus procesos, para que puedan ser combinadas en el máximo número de formas para ofrecer el mayor número de productos o servicios con bajos costos de implementación.

Una descripción rápida de la Tecnología de Grupos permite caracterizar a la fabricación modular como un enfoque que busca la formación de “familias” de procesos con requisitos similares de fabricación. Los grupos de máquinas necesarias para la elaboración de cada familia se organizan en “células”, que son pequeñas líneas de fabricación o montaje; las máquinas de cada célula pueden ser ajustadas para adaptarse a las necesidades de los diferentes lotes de cada familia.

Entonces se entiende como modularidad de un sistema o de un proceso a la capacidad que tiene de ser estudiado, segmentado y conceptualizado en una integración de módulos que operan dentro del sistema de forma independiente de las operaciones de otros módulos, que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo; por tanto cada módulo está compuesto de entradas y salidas específicas con una unidad de procesamiento lógico que desarrolla una tarea. Se muestra en la Figura 1 la composición de un proceso por subsistemas y módulos.

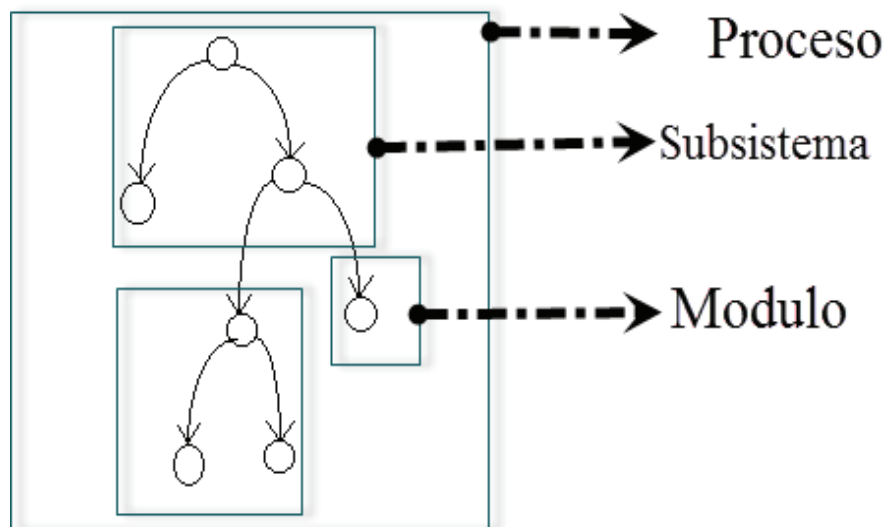


Figura 1. Modularidad de un proceso

La aplicación de la fabricación modular y la de tecnología de grupos sientan las bases de partida para lograr los máximos beneficios de la instalación de la automatización flexible, esto es, equipos que pueden ser reprogramados automáticamente a nivel de proceso de modo que puedan elaborar diferentes productos, por lo que son útiles tanto en configuraciones continuas como intermitentes:

- El primer caso, cuando se reduce la demanda de un determinado artículo, las máquinas pueden ser reconfiguradas para atender a la fabricación de otro, manteniéndose un nivel adecuado de utilización de los equipos. Modularidad a nivel hardware.

- El segundo, los cambios de operaciones entre lotes de productos son simples, pues basta con cambiar los programas que dirigen la actuación de los equipos. Modularidad a nivel Software.

2.2.1 Procesos discretos

Por proceso se entiende aquella parte del sistema en que a partir de la entrada de material, energía e información se genera una transformación, sujeta a perturbaciones del entorno, que da lugar a la salida de material en forma de producto [10], Figura 2. Los procesos industriales comúnmente se conocen como procesos continuos, procesos discretos y procesos batch.

De forma general, los procesos continuos se caracterizan por la salida en forma de flujo continuo de material, como por ejemplo la purificación de agua o la generación de electricidad. Los procesos *batch* son aquellos en los que la salida del proceso se lleva a cabo en forma de cantidades o lotes de material, como por ejemplo la fabricación de productos farmacéuticos o la producción de cerveza.



Figura 2. Estructura general de procesos [10]

Los procesos discretos suelen estar emplazados dentro de una secuencia temporal que puede ser acelerada o parada, donde el proceso de fabricación normalmente es por partes, por eventos o por hilos, y se denominan discretos por que se suele manejar magnitudes del tipo “todo” o “nada”, representando conceptos como, por ejemplo, si una pieza está o no esta, si ha llegado al sitio que le corresponde o no, etc. Además, en los procesos con automatización discreta también hay magnitudes analógicas, pero las digitales representan un porcentaje mucho más alto. La automatización discreta reduce los desechos y defectos en manufactura, reduce drásticamente los plazos de entrega, incrementa la flexibilidad y a la vez genera, de manera constante, productos de alta calidad [11].

En el análisis sistemático de un proceso, es necesario estructurar y desfragmentar la totalidad del proceso para poder entender el concepto de producción como una unión sinérgica de subsistemas jerarquizados. Esto permite tener un mejor control de los subprocesos involucrados y determinar en las primeras iteraciones la eficiencia del sistema de control para posteriormente evaluar los cambios de organización y de sistematización con el fin de mejorar el sistema.

2.2.2 Procedimientos de operación

La automatización de los procesos industriales debe contemplar todos los posibles estados en que se pueden encontrar las máquinas y los equipos para poder asignar un orden de jerarquía en las operaciones. Se debe considerar prioritario que el sistema esté preparado para afrontar los diversos procesos y estados, para ello el programa del autómatas deberá prever qué hacer frente a posibles problemas, a fin de reducir el tiempo de parada al máximo y permitir identificar el proceso en conflicto [11].

Los procedimientos de la automatización, vistas desde el autómatas sobre el proceso, pueden dividirse en procedimientos de parada y puesta en marcha, procedimientos de funcionamiento y procedimiento de falla; estos procedimientos se describen a continuación.

Parada y puesta en marcha:

Referido a los modos de funcionamiento en donde el sistema se encuentra parado, para luego ponerlo en marcha. Puede ser:

- Parada en estado inicial: Corresponde al estado de reposo normal inicial.
- Parada solicitada al final del ciclo: Consiste en hacer que una máquina deje de operar cuando ya ha finalizado su ciclo de trabajo normal, la máquina pasa al estado de parada inicial.
- Preparación para la puesta en marcha después de un defecto: en este estado deben realizarse las acciones necesarias para corregir las fallas. Finalizada la reparación, el operador elegirá como y cuando reiniciar la máquina en el proceso productivo.
- Puesta del sistema en estado inicial: se realiza de forma automática, retornando la máquina a su estado inicial.

Procedimientos de funcionamiento:

- Producción normal: Es el estado de producción común de funcionamiento automático.
- Marcha de preparación: Acciones necesarias para permitir que la máquina entre en funcionamiento.
- Marcha de cierre: corresponde a la Fase de vaciado o limpieza que reciben las máquinas antes de parar.
- Marcha de verificación sin orden: El operador hace funcionar la máquina en forma manual, para verificar su correcto desempeño, también se usa para posicionar la máquina en un determinado estado o lugar.
- Marcha de verificación con orden: la máquina realiza un ciclo de funcionamiento al ritmo dado por el operador, se usa para funciones de mantenimiento y verificación.
- Marcha de test: Es una prueba de comprobación de buen funcionamiento que realiza el autómatas.

Procedimientos de falla:

- Parada de emergencia: implica la parada de los actuadores, además se debe guardar en memoria la posición en la que quedo la máquina.
- Diagnóstico y/o tratamiento de defectos: el autómatas puede guiar al operador para indicarle donde se encuentra el defecto, el personal de mantenimiento deberá reparar la falla.
- Producción a pesar de defectos: corresponde a los casos en que la producción debe continuar a pesar de que el sistema no funcione correctamente. Algunas acciones pueden realizarse manualmente por el operador.

2.3 Estructura modular de procesos

La estructuración de un proceso a automatizar es hecha de modo que se divide el problema en partes más pequeñas, las cuales pueden ser subdivididas. Existen límites para esto: no es práctico continuar dividiendo hasta las últimas secuencias, porque el esfuerzo entonces se orientaría después hacia la integración de esas partes.

Los métodos de programación modernos proveen de herramientas para mejorar la calidad de la planeación del algoritmo de control, y para su correcto funcionamiento en el sentido de repetitividad, robustez, integridad, persistencia y de seguridad. Esto es, se busca una estructuración a nivel software considerando los diferentes niveles de implementación a nivel hardware [12]. Un enfoque en estructuras provee de ventajas como:

- Visión global del sistema, la instalación y el personal de mantenimiento.
- Base para la comunicación interna del multidisciplinario equipo de desarrollo.
- Base para la reutilización del software.
- Documentación inherente y automática.

En este sentido, la aparición del estándar de programación IEC 1131²-3 [13] proporciona lenguajes y métodos estandarizados que permiten resolver un amplio rango de problemas tecnológicos como organización de elementos de software. La estructuración está presentada como la combinación de dos principios que cooperan para la estructura de programación del proceso: la modularidad y la descomposición [13]. Se describen a continuación.

Principios de Modularidad

Con los modernos métodos de desarrollo de software existen cinco tipos asociados a la modularidad. Estos son:

- El lenguaje de programación debe apoyar a las unidades modulares.
- Las unidades deben estar compuestas en tal parte/número que ellas tengan pocas interfaces y pocas interacciones.
- Las interfaces deben ser pequeños, necesitando pocos intercambios de datos.
- El módulo de interacciones requiere una definición explícita, para incrementar su reuso.

² International Electrotechnical Commission, 1993; El estándar 1131 es el primer esfuerzo real para estandarizar los lenguajes de programación usados para la automatización industrial en sistemas de control.

- Los módulos deben mejorar la encapsulación de datos: los datos de aplicación son particionados, y cada partición sólo es accesible a través de un set de funciones las cuales estarán ocultas para usos indeseados.

Principios de Descomposición

Como herramienta de descomposición a nivel software, la IEC 1131 provee de Cartas de Función Secuencial, *SFC*, que describen gráficamente el comportamiento secuencial de un programa de control para dar estructura a la organización interna de un programa descomponiendo el problema de control en partes manejables, mientras se mantiene una visión de conjunto en cada componente del proceso y apoyan las secuencias alternativas y paralelas, por ejemplo, una secuencia es usada para el proceso primario y una segunda monitorea las acciones de operación general del programa.

Los elementos de las SFC consisten en pasos y transiciones; cada paso representa un estado del sistema enlazados con las acciones, y una transición que representa una condición, la cual cuando es cierta causa la desactivación del paso previo y la activación del paso siguiente, Figura 3. Los bloques de acción o transición pueden ser programados en alguno de los lenguajes de la IEC [14], diagrama de contactos, diagrama de bloques funcionales o lista de instrucciones y texto estructurado.

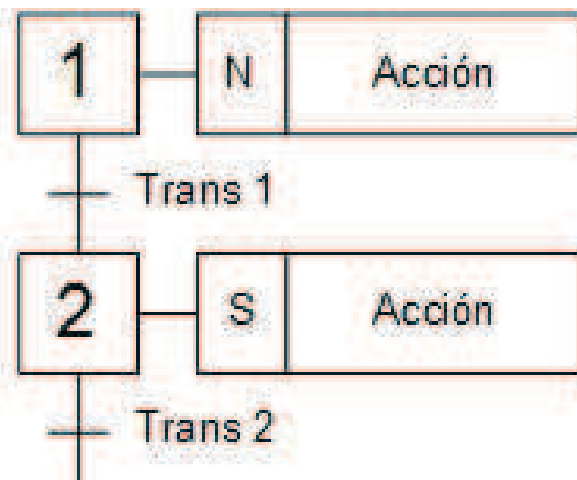


Figura 3. Ejemplo de una SFC

Para desarrollar la definición de las funciones de un proceso los siete pasos siguientes mejoran el camino a seguir para la estructuración del software:

1. Identificación de los interfaces³ externos al sistema de control.
2. Definición de las principales señales intercambiadas entre el sistema de control y el resto de la instalación.
3. Definición de todas las interacciones del operador, prioridades y datos de supervisión.
4. Análisis de la descomposición del problema de control de nivel superior en particiones lógicas.
5. Definición de niveles de jerarquía y organización a nivel software, por ejemplo, programas y bloques funcionales.
6. Definición del tiempo del ciclo de escaneo para los componentes de la aplicación.

³ Entiéndase como interfaces a los elementos con los que el sistema interactúa, pueden ser elementos neutros o actuadores, y forman parte de un subproceso.

7. Configuración del sistema por definición de programas fuente, enlazando programas, con entradas y salidas físicas y asignando programas y bloques funcionales a las tareas.

La IEC 1131-3 ayuda especialmente en los últimos pasos 4-7, donde ocurre la traslación a software. Además de estos 7 pasos, existen algunos principios más que deben ser usados para optimizar el método de estructuración, estos principios son:

- Trabajo puramente simbólico: sin direccionamiento absoluto, esto sólo en la parte de declaración. Ventajas de fácil adaptación a los cambios del entorno, mayor nivel de reutilización de código y menores efectos colaterales.
- Las partes de programa pertenecientes a otra deben ser unidas también en el código fuente.
- Nombrar adecuadamente las variables y los bloques funcionales incrementa transparencia y la lectura global.

2.3.1 Jerarquía de sistemas basada en componentes

Las aplicaciones presentes y futuras se caracterizan por la integración de sistemas flexibles que se adaptan a las estrategias de control y a los cambios que requiera el proceso. Como consecuencia, se requiere de sistemas estructurados abiertamente que se puedan integrar tanto en módulos de producción como en sistemas de procesamiento [15].

La aplicación de estándares también tiene un gran impacto en el rápido crecimiento del mercado de la instrumentación y control de procesos industriales; así, implementar un modelo genérico que sea aplicable a muchos sistemas puede ahorrar una gran cantidad de tiempo en su estudio de sistematización. Por tanto, un modelado del proceso basado en componentes modulares permite la flexibilidad tanto a nivel de implementación como a nivel de planeación.

En este sentido, el estándar IEC 1131-3 describe los programas, funciones y bloques funcionales (FB) como diferentes tipos de unidades de organización de programa (POU) que pueden ser utilizados varias veces a lo largo de las distintas partes que componen una aplicación. La definición de POU favorece la reutilización del software desde el más alto nivel, con los programas, hasta el más bajo nivel, por medio de las funciones y de los FB. En la Tabla 2 se resumen los tipos de POU.

Tabla 2. Tipo de POU		
Tipo POU	Implementación	Comentarios
Programa	Programa instancia ⁴	Máximo nivel de reutilización
Bloque funcional	Bloque funcional instancia	Permite la descomposición de un algoritmo de control complejo en algoritmos simples que pueden ser reutilizados
Función	Función	Para manipulación de datos

Para poder desarrollar el sistema de control es necesario partir de un esquema que modele el proceso basado en componentes genéricos, estos tienen que corresponderse con los componentes

⁴ Instancia: en programación, todo objeto que derive o genere de algún modelo específico.

básicos funcionales (FBC) del proceso perteneciente a un módulo, a nivel de funcionalidad, y con el concepto de FB y de instancia de FB del sistema de control, a nivel de programación.

Un esquema genérico inicia con la conceptualización general del proceso donde se indica los elementos como mecanismos, actuadores y sensores, posteriormente se separa en esquemas planos las diferentes etapas del proceso donde se identifican todos los sistemas, módulos, con sus componentes. En la Figura 4 se muestra un esquema abstracto del desglose de un proceso.

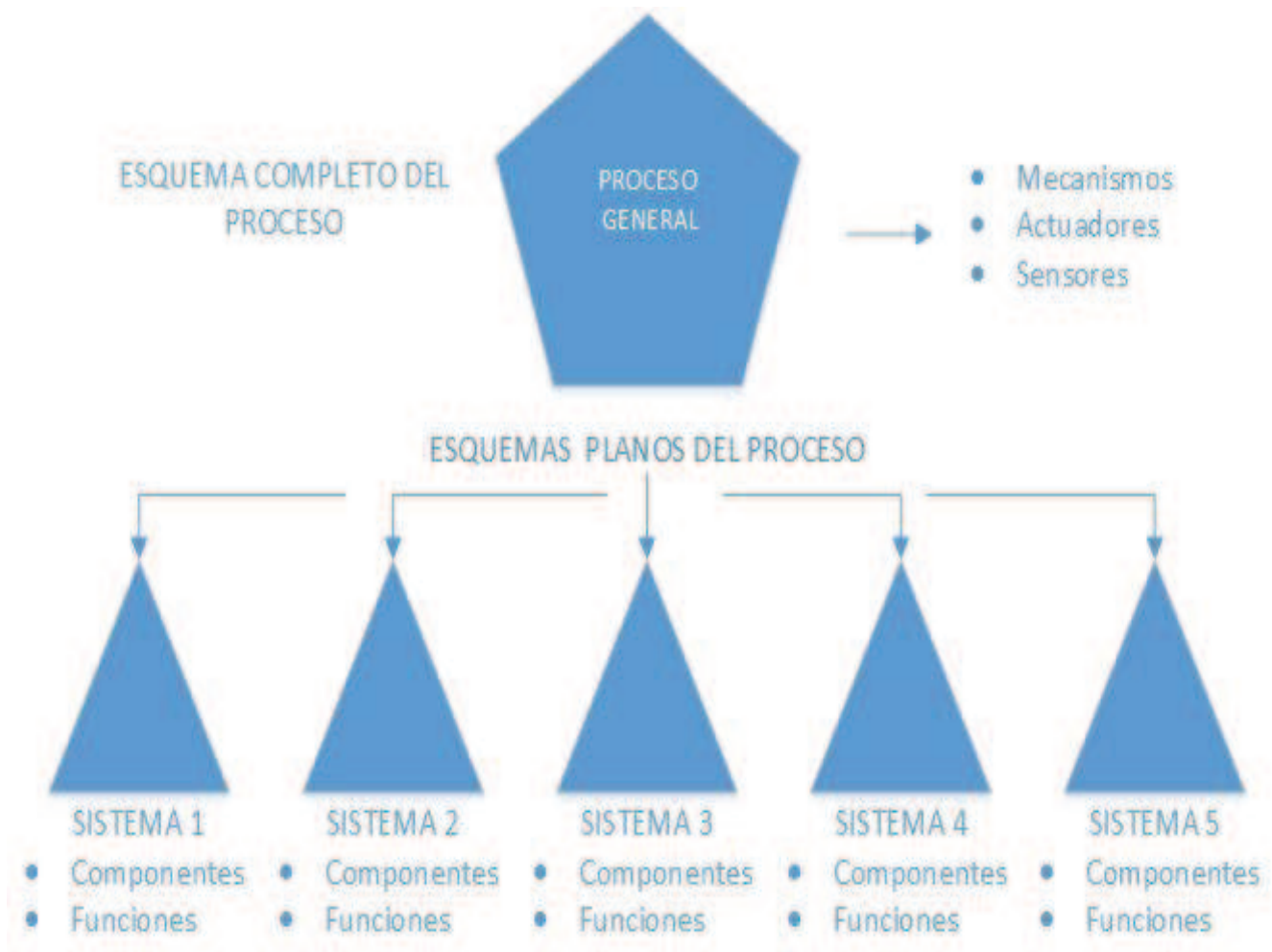


Figura 4. Esquema genérico del proceso

A partir de un análisis esquemático del proceso se identifican el o los sistemas de control, se pasa de un sistema físico a la abstracción de un sistema de control mediante la identificación de sus funciones. El sistema de control se debe especificar como una jerarquía de cuatro niveles:

- Nivel 0: planta completa, que engloba todos los sistemas.
- Nivel 1: subsistemas independientes de planta.
- Nivel 2: elementos funcionales asociados al subsistema de nivel 1.
- Nivel 3: componentes funcionales elementales asociados a cada elemento de nivel 2.

Cada nivel de la jerarquía está formado por un conjunto de FBC con unas entradas y salidas. Todos los FBC independientemente de la capa en que pertenezcan, están definidos de la misma forma y tienen los mismos componentes y características.

En la Figura 5 se muestra que a partir del esquemático del proceso en sistemas, definidas las funciones por cada sistema, se establecen los FBC para posteriormente darles una jerarquía de dependencia por niveles, obsérvese que puede que diferentes sistemas de control pueden tener en común más de un FBC.

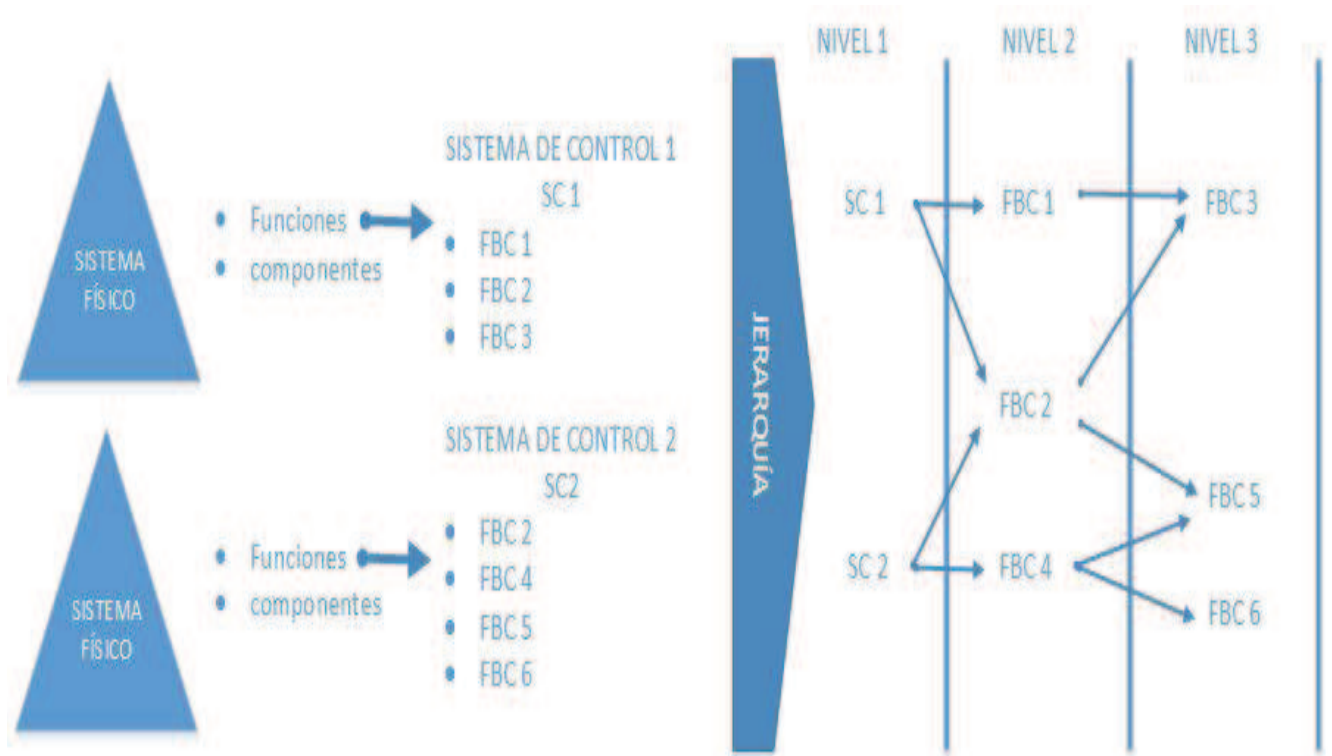


Figura 5. Niveles de jerarquía de un proceso

Los FCB pertenecientes a una aplicación concreta están descritos de la siguiente forma:

- **Tipo:** Tipo de FBC que se está categorizando.
- **Categoría:** El nivel asociado dentro de la jerarquía del FBC.
- **Etiqueta:** Identificador asignado por el programador.
- **Entradas:** Lista de conexiones de entradas de otros bloques o del exterior.
- **Salidas:** Lista de conexiones de salidas a otros bloques o al exterior.
- **Función:** Si el FBC es de nivel 3 se describe el código fuente que ejecuta su funcionalidad, si no, se enumeran los FBC de nivel inferior que contiene.
- **Datos internos:** Lista de variables locales necesarias para poder implementar la funcionalidad del FBC.
- **Parámetros de configuración:** Usados para adaptar la lógica del FCB.

Los FCB pueden reutilizarse dentro del mismo modelo, pero también en otros dominios de aplicación, por ejemplo en un modelo complejo de planta, como en procesos industrializados, con sistemas comunes que tan sólo difieren en el valor de algunos parámetros de configuración en sus funciones, así estos pueden ser definidos una sola vez y después ser utilizados tantas veces como sea necesario.

En la Figura 6 se muestra un FCB genérico para el modelado de procesos, uno donde se describe el FBC de forma individual y otro donde se describe con sus dependencias, esta distinción permite

primero declarar de forma general los FCB del sistema a partir de sus esquemas, para posteriormente agrupar el nivel de dependencias para identificar las funciones y posteriormente definir los niveles de jerarquía.



Figura 6. FCB genérico

El concepto de FBC nos permite definir los FB con base en la funcionalidad debido a que el diseño jerárquico permite su reutilización en cada nivel, en consecuencia, el diseño del algoritmo de control es más fácil de estructurar por la jerarquía y la especificación de las funciones necesarias para los sistemas de control. Por ejemplo, los FB se definen por medio de una declaración de tipo, y se utilizan copias de éste denominadas instancias en diferentes niveles dentro de un FBC.

2.3.2 Grafcet de procesos

Los trabajos efectuados por las comisiones de AFCET⁵ y de ADEPA⁶ han dado como resultado la definición de un diagrama funcional: el **GRAF CET**, del francés, gráfico de mando etapa/transición, Anexo 1. Este diagrama funcional permite describir los comportamientos del automatismo en relación a las informaciones que recibe a partir de SFC, imponiendo un funcionamiento riguroso, evitando de esta forma incoherencias, bloqueos o conflictos en el funcionamiento. En cada nivel de

⁵ Association Française pour la Cybernétique Economique et Technique

⁶ Agence Nationale pour le Développement de la Production Automatisée

descripción, este diagrama puede ser modificado o corregido, sin necesidad de volver a partes ya estudiadas.

La creciente complejidad de los automatismos industriales se traduce en una mayor dificultad para definir de una manera clara y sin ambigüedades las especificaciones funcionales a las que deben responder. Esta dificultad se ve agravada por la utilización de un gran número de informaciones de entrada y salida [13]. La herramienta GRAFCET da solución al desarrollo de programas de control para procesos de una forma gráfica donde se hacen esfuerzos por comprender la estructura del proceso basada en sus componentes. Las distintas formas de descripción de un proceso pueden ser clasificadas de la siguiente forma:

- Una descripción literal únicamente resulta larga, incomoda, a veces imprecisa y, a menudo incompleta.
- Una descripción lógica (logigrama) está casi exclusivamente enfocado a una tecnología determinada, y no es apropiada a la realización mediante programa.
- Una representación por organigrama, más general, se adapta bien a las realizaciones mediante programa, pero resulta pobre en el caso de los secuenciales y no muestra los funcionamientos simultáneos, caso de que los haya.

Las representaciones gráficas de las especificaciones funcionales de un proceso son totalmente independientes de la realización tecnológica. Estas formas de representación se basan en los conceptos de etapa y de receptividad que simplifica en gran medida la síntesis de los automatismos secuenciales, al considerar el hecho de que, entre el gran número de informaciones disponibles, pocas son significativas en un determinado momento.

Los principios que inspiraron la creación del GRAFCET y en los que se basa su aplicación son los siguientes [16]:

- a) Debe caracterizarse el funcionamiento del automatismo con total independencia de los componentes con los que vaya a ser construido. Esto equivale a centrar el interés no tanto en la estructura física o en la tecnología empleada para implementar el automatismo, sino en la función que debe realizar.
- b) El conjunto de un sistema automático se divide en dos partes: parte de control y parte operativa. La parte de control comprende todo aquello que contribuye a la automatización del proceso y la parte operativa incluye el resto del mismo. El conjunto está relacionado con el medio exterior a través de un dialogo con el operador y comunicaciones con otros automatismos que operan en el mismo contexto, Figura 7.
- c) El elemento fundamental de un proceso es la operación (denominada etapa en el lenguaje GRAFCET), entendiendo como tal una acción realizada por el automatismo. En una primera aproximación puede dividirse el proceso en unas pocas operaciones relativamente complejas, estas operaciones podrán ser subdivididas a su vez en operaciones más elementales a medida que avanza el nivel de detalle.
- d) Debe dividirse en proceso en macroetapas⁷ y éstas en etapas más elementales, hasta conseguir que las acciones a realizar en cada una de ellas dependan sólo de relaciones combinacionales

⁷ En GRAFCET para procesos complejos, se empieza por representar un diagrama con las líneas principales a ejecutar en el proceso, definiendo grandes bloques sin desarrollar los detalles del proceso. Permite visión global.

entre entradas y salidas. Cada una de estas etapas elementales tendrá asociada una variable de estado.

- e) Establecer un gráfico de evolución que indique la secuencia de operaciones (secuencia de etapas) y las condiciones lógicas para pasar de una a otra (condiciones de transición en el lenguaje GRAFCET). Como resultado de esta Fase se obtienen las ecuaciones lógicas de las variables de estado y por tanto, queda resuelta la parte secuencial del automatismo.
- f) Establecer para cada operación elemental (etapa) las relaciones lógicas entre entradas y salidas, utilizando eventualmente otras variables internas combinacionales.
- g) Finalmente implementar el sistema en el lenguaje apropiado para el controlador que se haya escogido como unidad de control, utilizando las variables y condiciones de estado, y cableando o programando las relaciones lógicas entre las Fases e y f.

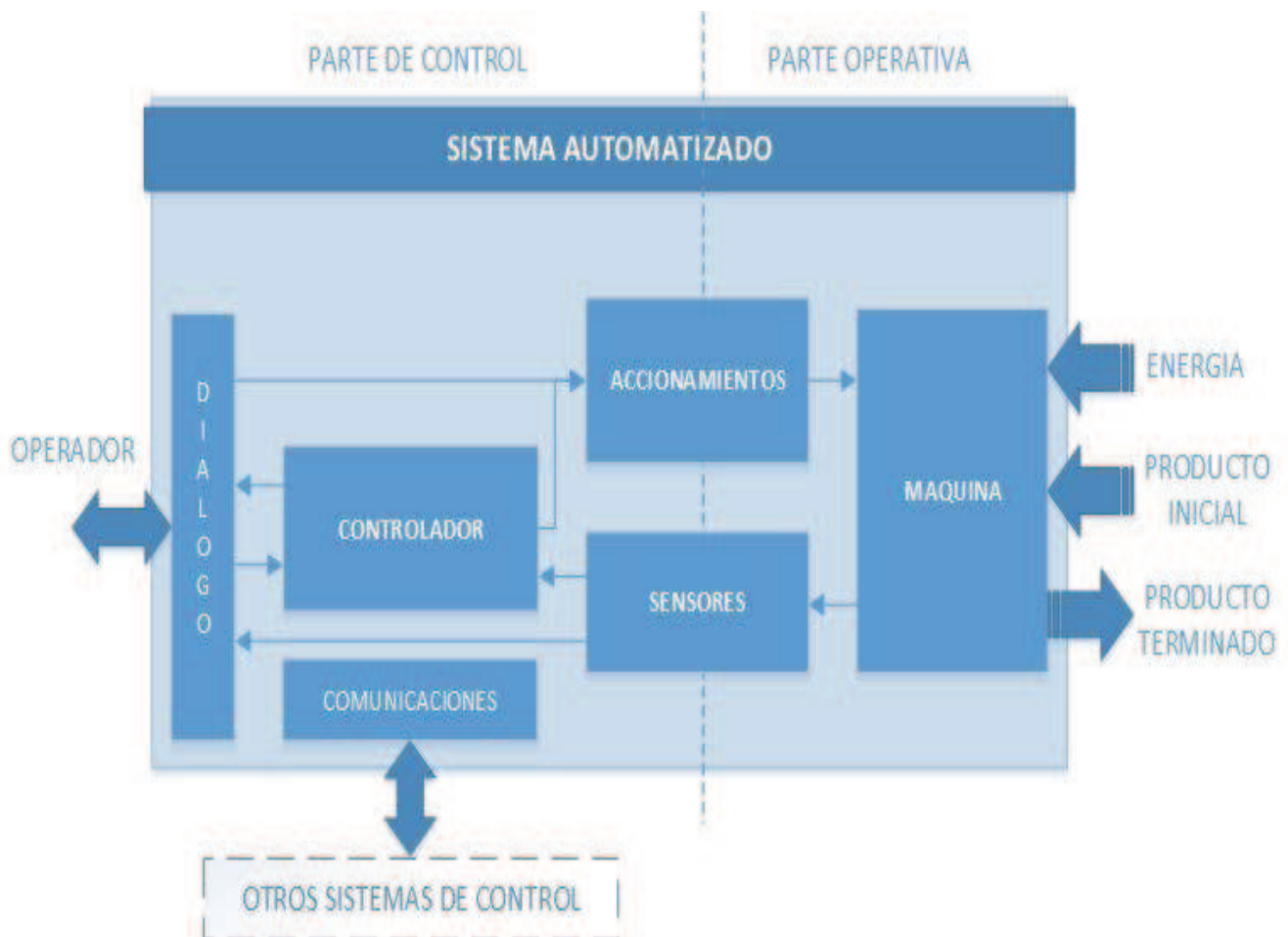


Figura 7. Estructura de sistema automatizado

Considérese que el método está basado en una pregunta clave que permite identificar la parte secuencial de un proceso; la pregunta es: ¿Cuántos estados debe memorizar el sistema, para poder fijar su comportamiento posterior, partiendo de cualquier estado inicial? La respuesta a esta pregunta nos permitirá identificar las etapas y, en consecuencia, las variables y condiciones de estado.

No se pretende minimizar el número de variables de estado ni reducir costes de la unidad de control, por lo que puede no resultar óptimo desde el punto de vista de minimizar el hardware. Sin embargo, el coste y volumen de un sistema dependen cada vez menos del número de variables empleadas, sobre todo si se emplean autómatas programables modulares y en cambio, adquieren cada vez más importancia otros aspectos como el propio coste de diseño de cada módulo, tiempo de desarrollo de software, fiabilidad y factibilidad de test y mantenimiento, aspectos que permite optimizar el método propuesto.

2.4 Programación de autómatas

En un principio, antes de la llegada de los sistemas electrónicos, esta adaptación era fija para los procesos con sistemas cableados que sólo permitían la configuración de parámetros; con el avance de las tecnologías en electrónica se generaron herramientas para la estructuración del software que permite procesos complejos. Por este atributo programable el autómata necesita, para su adaptación en el proceso, de un operador humano que defina como se requiere la evolución del mismo y la configuración de sus parámetros. Se define este conjunto de software como la integración de programas destinados a permitir o facilitar la utilización del hardware para la producción y explotación de las aplicaciones [17]. Según esta definición, se clasifica estas herramientas en dos grandes grupos:

- **Sistemas operativos**, residentes en el propio autómata, que se encargan de establecer las secuencias de intercambios de información, interpretar y ejecutar órdenes de usuario, y vigilar el correcto funcionamiento del equipo; se obtienen las variables de salida o de control.
- **Software de edición/depuración de programas**, que permite al usuario la introducción del programa sobre un soporte físico, la modificación del mismo en la puesta punto a punto, la obtención de la documentación pertinente y la creación de copias de seguridad.

En la Tabla 3 se muestra las características de los sistemas de control con base en la modularidad en su contra parte de sistemas fijos cableados.

Tabla 3. Comparativa entre sistemas de control y cableado		
Característica	Sistema cableado	Sistema de control
Flexibilidad de adaptación al proceso	BAJA	ALTA
Hardware estándar para distintas aplicaciones	NO	SÍ
Posibilidades de ampliación	BAJAS	ALTA
Interconexiones y cableado exterior	MUCHO	POCO
Tiempo de desarrollo del proyecto	LARGO	CORTO
Posibilidades de modificación	DIFÍCIL	FÁCIL
Mantenimiento	DIFÍCIL	FÁCIL
Herramientas para prueba	NO	SÍ
Modificaciones sin parar el proceso	NO	SÍ
Estructuración en bloques independientes	DIFÍCIL	FÁCIL

Desde el punto de vista de su papel dentro del sistema de control, el autómata programable es la unidad de control incluyendo, total o parcialmente, las interfaces con las señales de proceso. Por otro lado, se trata de un sistema con un hardware estándar con capacidad de conexión directa a las señales de campo como transductores y periféricos electrónicos [18], Figura 8.

El concepto de hardware estándar se complementa con el de modularidad, entendiendo como tal el hecho de que este está fragmentado en partes interconectables que permiten configurar un sistema a la medida de las necesidades y además permite la comunicación física con el proceso bidireccionalmente:

- Recoge el conjunto de variables, digitales y analógicas, que definen el estado del mismo, señales de entrada.
- Envía otro conjunto de variables que modifiquen dicho estado en un sentido predeterminado, señales de salida.

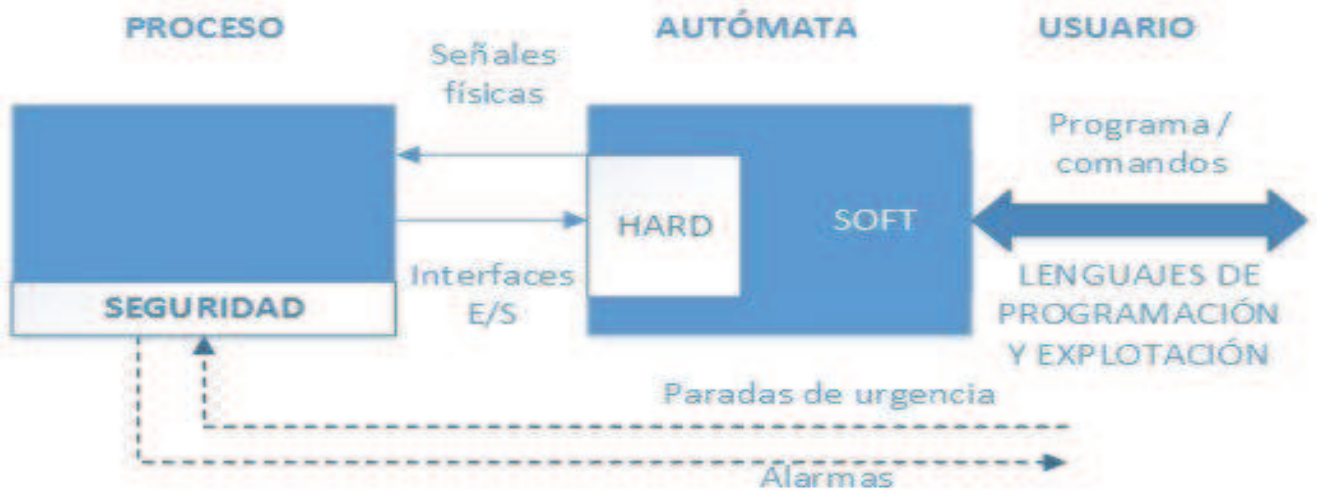


Figura 8. Unidad de control

Los intercambios de información entre los elementos del sistema de control necesitan de una codificación del mensaje que sea compresible a las partes que intervienen [19]. Esta compatibilidad se consigue mediante el empleo de dos tipos de lenguajes, de ***lenguajes de programación*** que se define como el conjunto de símbolos y textos inteligibles por una unidad de programación que le sirven al usuario para codificar en un autómata las leyes de control deseadas, mientras que ***lenguaje de explotación*** será el conjunto de órdenes y comandos que el usuario puede enviar desde la misma unidad o una terminal para recoger estados o modificar variables del proceso. La programación del autómata pasa por los siguientes pasos:

1. Identificar los componentes, señales, de entrada y salida al autómata.
2. Determinar qué debe hacer el sistema de control y en qué orden, por ejemplo, mediante un diagrama o un grafo GRAFCET.
3. Representar mediante un modelo el sistema de control, identificando las funciones que intervienen, las relaciones entre ellas y la secuencia que deben seguir. Esta representación puede ser algébrica o gráfica.
4. Asignar direcciones de entrada/salida o internas a cada uno de los componentes que aparecen en el modelo.
5. Codificar la representación anterior en instrucciones o símbolos inteligibles por la unidad de programación.
6. Transferir las instrucciones obtenidas a la memoria del autómata desde la unidad de programación.
7. Depurar el programa y obtener una copia de seguridad.

2.4.1 Estructura de algoritmos

El controlador de un sistema de control del autómatas programable tiene almacenado en su memoria un algoritmo o programa de instrucciones, este algoritmo contiene un programa principal o parte fundamental, y subprogramas o partes secundarias. Para las aplicaciones, el usuario operador necesita de una comunicación con la máquina para programar, depurar el programa y para acceder a los estados de planta.

La ejecución de un programa dentro de un controlador sigue un esquema cíclico. Un ciclo de un algoritmo se compone de los siguientes pasos [20]:

- Operaciones de gestión del sistema: Consiste en el tratamiento de la información, de las peticiones y las llamadas efectuadas por el terminal de programación y el envío de mensajes al terminal.
- Lectura del estado de las entradas: Consiste en la lectura de los registros de los módulos de entrada en el almacenamiento de estos datos en una memoria.
- Ejecución del programa almacenado: Empieza por la primera línea y sigue ejecutando línea por línea hasta la última. Durante esta ejecución no se considera una posible variación en el estado de las entradas. Una vez ejecutada la última línea se actualizan los resultados de los estados en la memoria.
- Estructura de datos de salida: Se efectúa una transferencia de información de la memoria a los módulos de salida. El último valor que tome la variable de salida será almacenado en la memoria.

La ejecución del programa principal y los subprogramas puede realizarse de dos maneras:

- Ejecución cíclica: Consiste en encadenar los ciclos del programa uno tras otro en forma continua y sin interrupciones. Después de actualizar las salidas, el sistema pasa a realizar el cíclico nuevamente y así sucesivamente. Es el método de ejecución por defecto para procesos continuos.
- Ejecución periódica: consiste en ejecutar el programa durante cierto tiempo o también durante un número definido de veces. Estos criterios pueden ser definidos por el operador, comúnmente en procesos discretos.

2.4.1.1 Paradigma en microcontroladores

El *paradigma* debe ser concebido como un conjunto de métodos, reglas y generalizaciones utilizadas conjuntamente por aquellos entrenados para realizar el trabajo científico de investigación. En el contexto de unidad de control, los paradigmas de programación nos indican las diversas formas que, a lo largo de la evolución de los lenguajes, han sido aceptadas como estilos para programar y para resolver los problemas por medio de una computadora. [21]

Las acciones que pueden ser desempeñadas por los sistemas con microcontroladores, como robots autónomos, se clasifican en tres categorías según la función que desempeñan dentro del sistema. Estas tres clasificaciones, denominadas primitivas de la robótica, son:

- **Sensar:** se refiere a las funciones que obtienen información de los sensores del robot, los cuales pueden ser dos tipos dependiendo de la información que proporcionan, propioceptivos para estados internos del robot (como encoders) y exteroceptivos para información del ambiente (como sonares).
- **Planear:** se refiere a una función de planeación, cuando se generan subtareas o se toman decisiones, a partir de la información obtenida de los sensores o de alguna representación del conocimiento generado anteriormente.
- **Actuar:** es el grupo de funciones donde se envían comando a los actuadores para ejecutar el movimiento de alguna parte del robot, generalmente basados en una ley de control que permite que el actuador llegue a un valor deseado en un tiempo óptimo. [12]

Utilizando estas funciones primitivas, se definen el **Paradigma Deliberativo** para organizar la inteligencia de los robots autónomos que representa un proceso continuo cuyos parámetros dependen del usuario y trata de emular la forma en que el humano piensa desde una perspectiva introspectiva, este paradigma forma parte de los planteados por Murphy [13]; dicho modelo se basa en la forma en que interactúan las funciones entre sí mediante sus entradas y salidas, Figura 9. Cuando se completan los objetivos planeados y se han ejecutado las instrucciones, este proceso termina y se repite nuevamente hasta que el autómatas termine una secuencia iterativa [17].

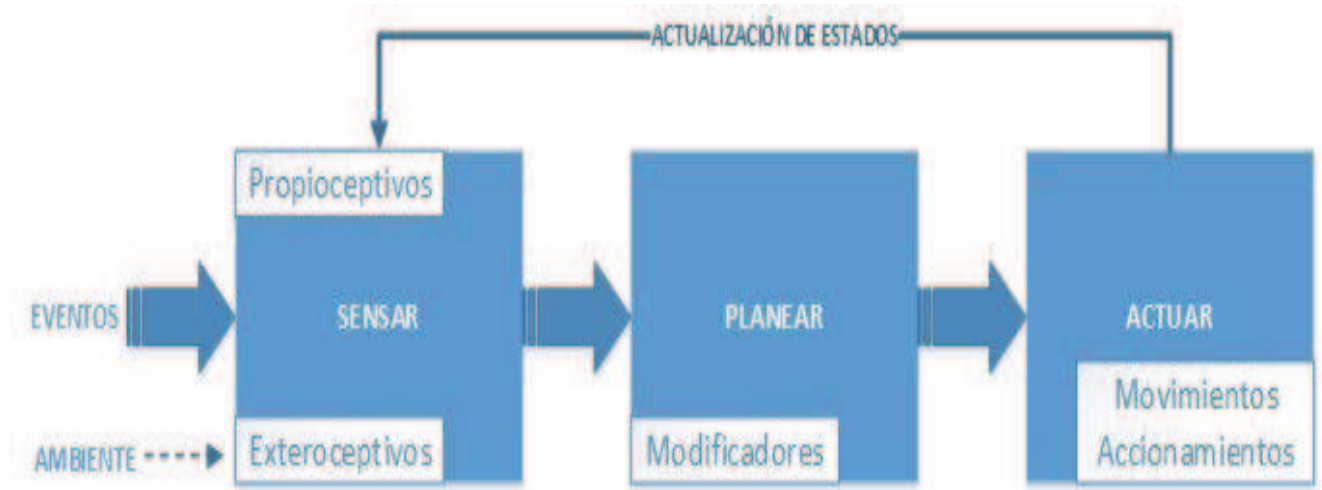


Figura 9. Representación del paradigma deliberativo

2.4.1.2 Programación orientada por eventos

Es importante destacar que hacer referencia a la programación orientada implica una forma de pensar acerca del proceso de descomposición de problemas y de desarrollo de soluciones de programación. Tradicionalmente, se enfrentarse a esta complejidad mediante la programación estructurada, que consiste en descomponer el problema en sub problemas hasta llegar a acciones muy simples y fáciles de codificar, se trata de descomponer el problema en acciones, en verbos [18].

A finales del año de 1970 se definieron nuevos modelos de control del flujo de datos que implementaron lo que llamaron transacciones, un patrón de diseño de manejadores de eventos, que divide el flujo de paquetes de entrada en varios sub flujos discretos de salida de tal forma que sigue niveles de descomposición del problema original, Figura 10.

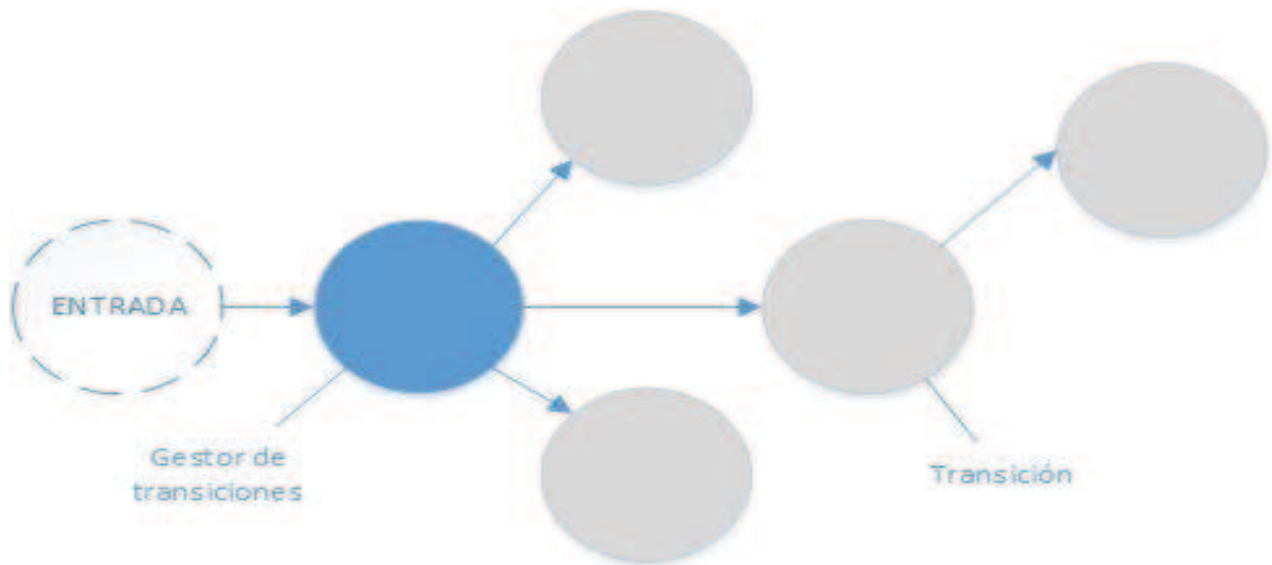


Figura 10. Control de flujo de datos

El diagrama de la Figura 10 es el arquetipo del **paradigma dirigido por eventos** en el que se define una transición inicial cuando algún elemento de datos, control, señal, evento, o cambio de estado es enviado al proceso que conforma al gestor de transiciones. Este debe ser capaz de obtener y responder a las transiciones, analizar cada una para determinar su tipo, despachar de acuerdo al tipo de entrada y su nivel de jerarquía, y completar el proceso de cada una [18]. Para esto, el principal patrón estructural que constituye la esencia del paradigma dirigido por eventos es el llamado patrón **Handler** o manejador, Figura 11.

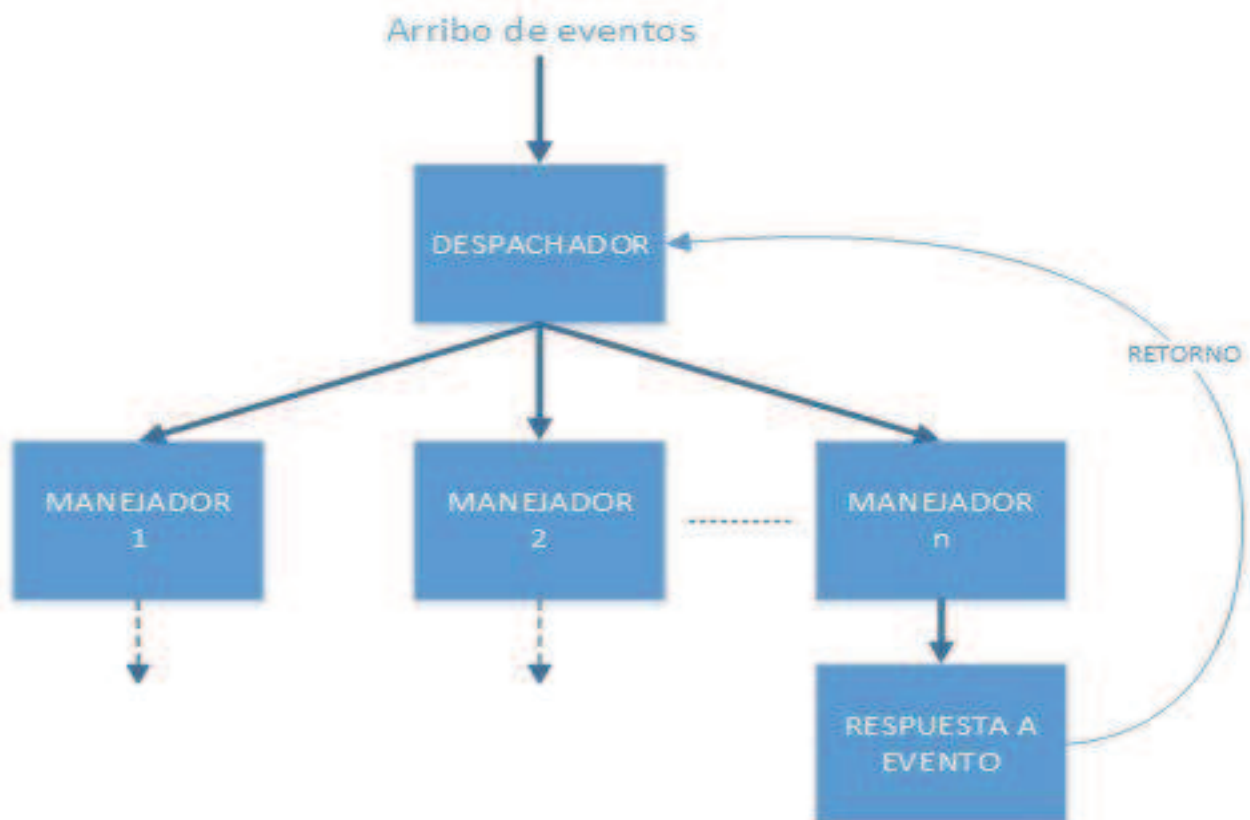


Figura 11. Representación del paradigma dirigido por eventos

Los elementos estructurales del manejador son: un flujo de datos llamados eventos, un despachador de eventos y un conjunto de manejadores. La función del despachador es tomar cada uno de los eventos que van llegando a él, analizar cada evento para determinar su tipo y finalmente enviar cada uno de estos a los respectivos manejadores quienes realizan alguna función de acuerdo al tipo de evento que estos puedan procesar.

En la Figura 12 se muestra una fracción en pseudo código que representa a un despachador. Este incluye las características antes descritas que son: un bucle continuo, una operación para salir, la determinación del tipo de eventos así como la selección apropiada de un manejador para su procesamiento y finalmente el tratamiento de eventos que no cuentan con algún manejador para su tratamiento.

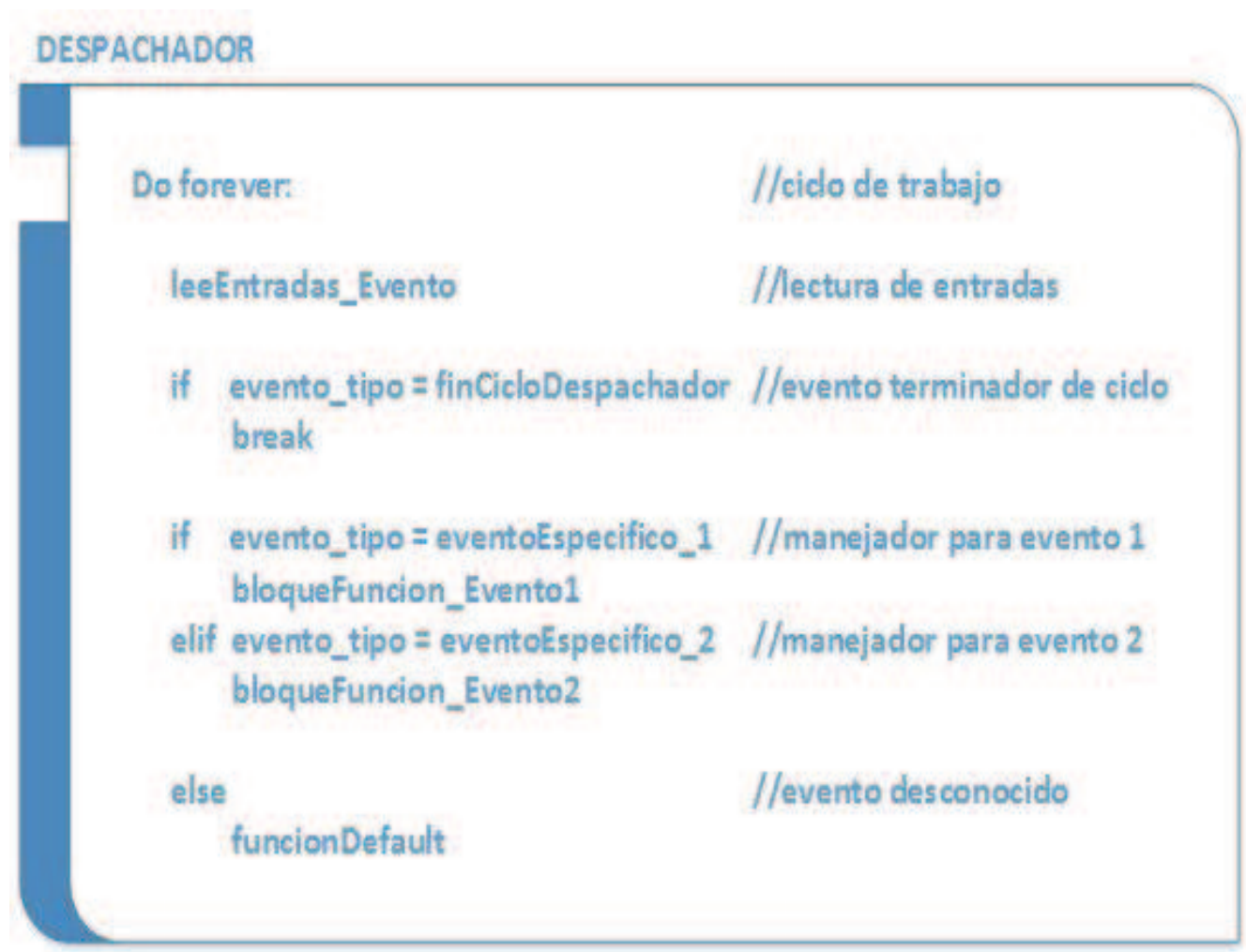


Figura 12. Despachador de eventos

2.4.2 Bloques funcionales

La existencia de bloques funcionales, más o menos complejos, como una herramienta intermedia en el paso de haber establecido el concepto del modelo a un lenguaje codificado aumenta la potencia de cálculo del autómata y simplifican la programación al añadir a la estructura del programa sentencias pre-programadas que son de uso general en la automatización. Puede entonces hacerse una referencia al concepto de modularidad a nivel de software con el mismo objetivo que la

modularidad en procesos: fragmentar en subfunciones, módulos, que puedan interactuar entre ellos para un fin común.

Estos bloques, que pueden ser definidos dentro de un diagrama de contactos, de flujo o como sentencias literales en una lista de instrucciones o lenguaje de alto nivel, se clasifican en dos grupos atendiendo a su forma de operación y disponibilidad en el programa:

- Bloques secuenciales básicos, de uso general incluso en autómatas de gama baja. Son los biestables, temporizadores, contadores, registros de desplazamiento y configuración de puertos.
- Bloques de expansión o funciones, que aumentan la potencia del lenguaje al permitir manipular variables numéricas y registros de datos, con instrucciones aritméticas, de comparación, transferencias, etc.

Los primeros pueden considerarse como parte de los lenguajes básicos del autómata, mientras que los segundos son extensiones de ellos. Todos estos bloques funcionales son sentencias pre programadas de aplicación general, es decir, el usuario debe adaptarlos a sus necesidades indicando las condiciones de trabajo: valores de temporización, direcciones de origen y destino de transferencia, monitoreo, etc.

En su caso más general, los bloques presentan tres tipos de variables asociadas:

- Condiciones de operación, que definen la habilitación y control del bloque.
- Operandos de función, sobre los que actúan las sentencias pre programadas en el bloque funcional.
- Salidas asociadas, cuyo estado depende de la ejecución de la función del bloque.

Los operandos de función pueden a su vez ser de dos tipos:

- Parámetros iniciales, que una vez fijados por programa o transferidos desde consola permanecen normalmente inalterados.
- Datos de operación, variables constantes contenidas en palabras de 8 o 16 bits que reflejan el estado de entradas/salidas, valores internos, resultados, etc.

Estos datos que aparecen como operandos pueden corresponder a:

- Números o caracteres ASCCII constantes definidos en el programa.
- Textos pre programados escritos en alguna unidad de memoria o dispositivo exterior.
- Variables numéricas.

La implementación más frecuente de un bloque funcional, sobre todo en los procesos más complejos, en diagramas de flujo se considera como elemento compacto que hace referencia a una subrutina controlada, Figura 13.

La integración a los diagramas de flujo para la planeación de la programación permite una forma compacta para hacer referencias a sub-funciones, o módulos, que se caracterizan por hacer una tarea en concreto capaz de ser usada en diferentes tiempos de operación. Su parecido a los macros en GRAFCET permite la estructuración intuitiva de la función, así el esfuerzo en la conceptualización del proceso facilita la conceptualización de software.

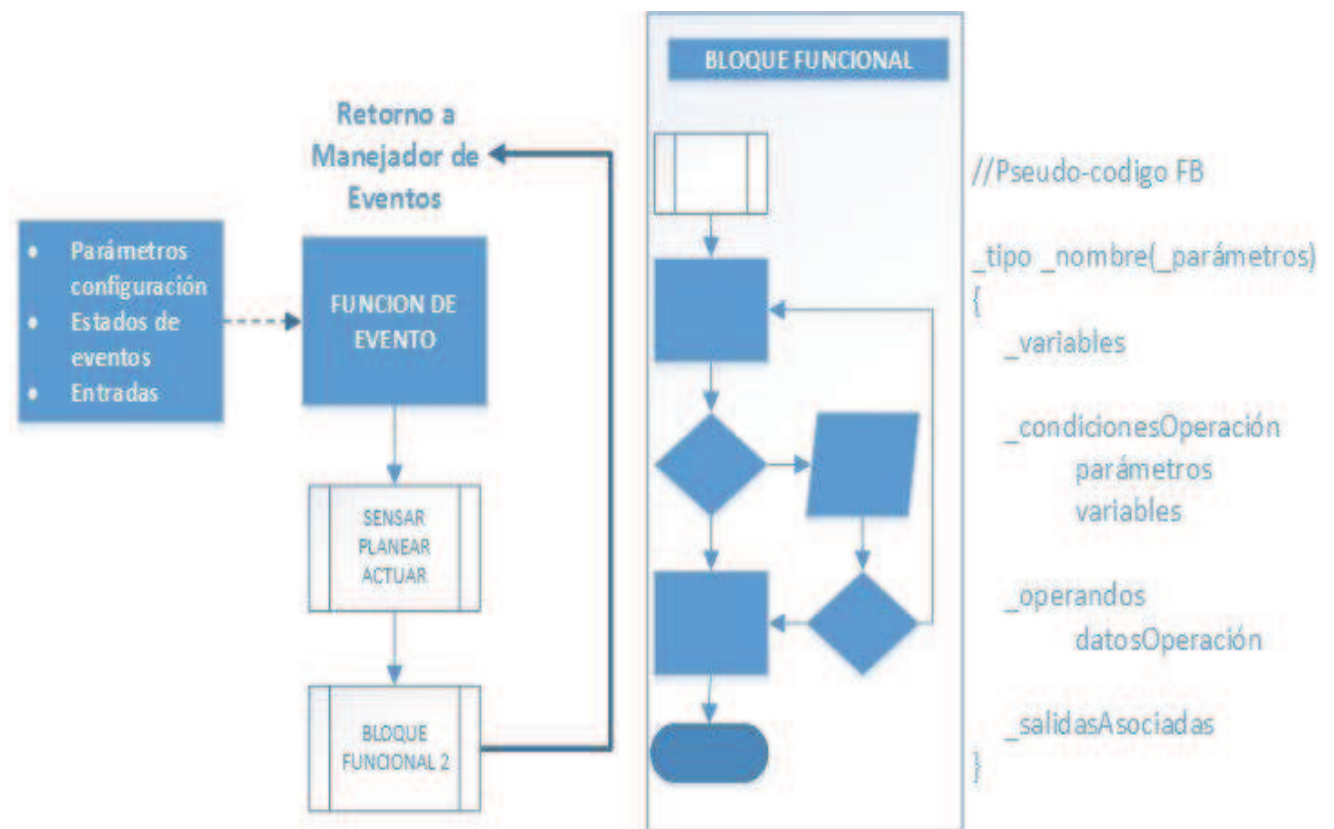


Figura 13. Bloque funcional

2.4.3 Estructura de la unidad de control

El objetivo de un sistema de control es gobernar la respuesta de una planta, sin que el operador intervenga directamente sobre los productos de salida. Dicho operador manipula únicamente las magnitudes denominadas de configuración y el sistema de control se encarga de gobernar dicha salida a través de accionamientos. En el caso más general, se puede dividir el sistema de control en los siguientes bloques [17]:

- Unidad de control.
- Accionamientos.
- Sensores.
- Interfaces.

En las plantas se encuentra habitualmente una diversidad de componentes o subsistemas de tipo mecánico, hidráulico, neumático, eléctrico o fisicoquímico. Se trata de autómatas que combinan múltiples tecnologías, haciendo necesario un lenguaje común para la coordinación e integración óptima de todas ellas en el sistema, Figura 14.

A nivel físico, la relación entre dichos subsistemas tecnológicamente diversos la realizan los sensores e interfaces. Pero a nivel de caracterizar su comportamiento, el diseñador necesita un modelo independiente que permita tratar a todos ellos con una metodología común. Con esto se busca la integración del comportamiento de los diversos actuadores y sensores en un lenguaje universal para el autómata.

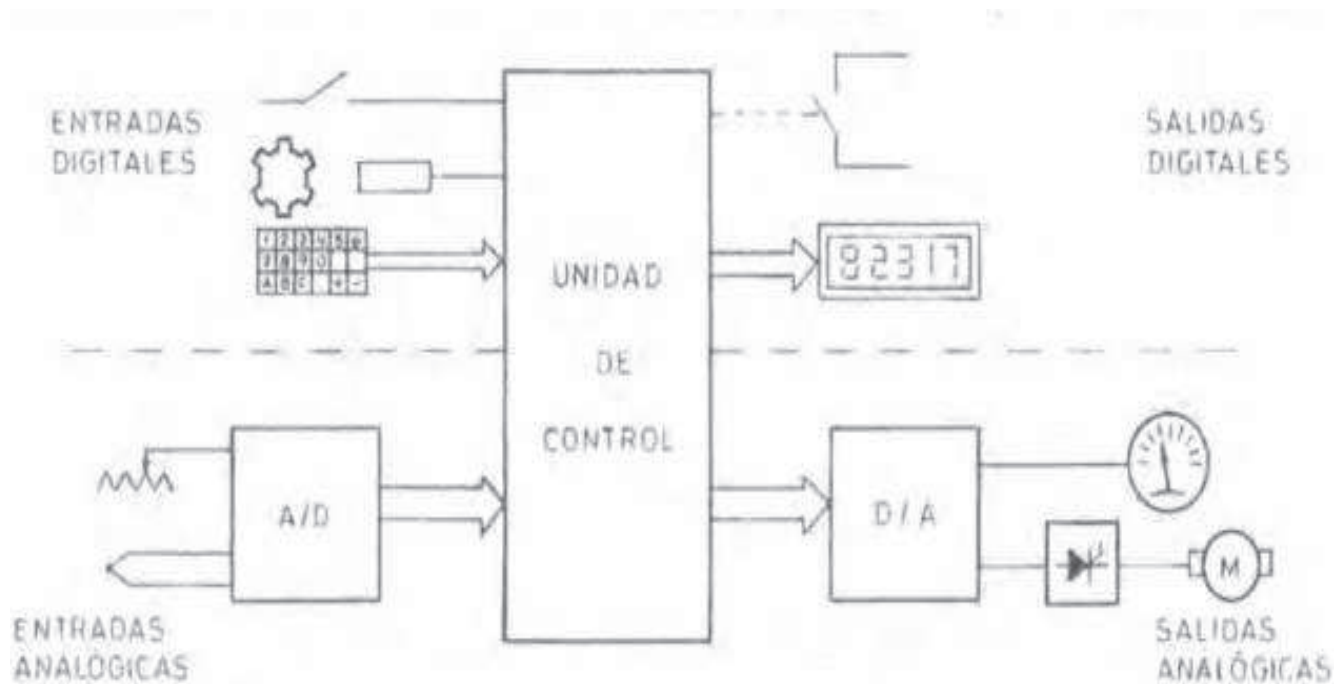


Figura 14. Componentes de una planta [26]

Se debe recordar que dentro de la unidad de control se integran todos los bloques funcionales en un lenguaje de programación el cual representa la parte operativa del proceso, pero no siempre una unidad de procesamiento podrá con la tarea del proceso al considerar multitareas, por lo que se debe considerar el cómo se define la estructura del software y del hardware.

En la modularidad del sistema, así como se acondicionan las señales físicas se debe acondicionar las señales de comunicación cuando existen diferentes módulos de programación. Al desglosar la unidad de control se determina las partes de la Figura 15: Interfaces de usuario, interfaces de máquina, de comunicación, módulos maestros y módulos esclavos.

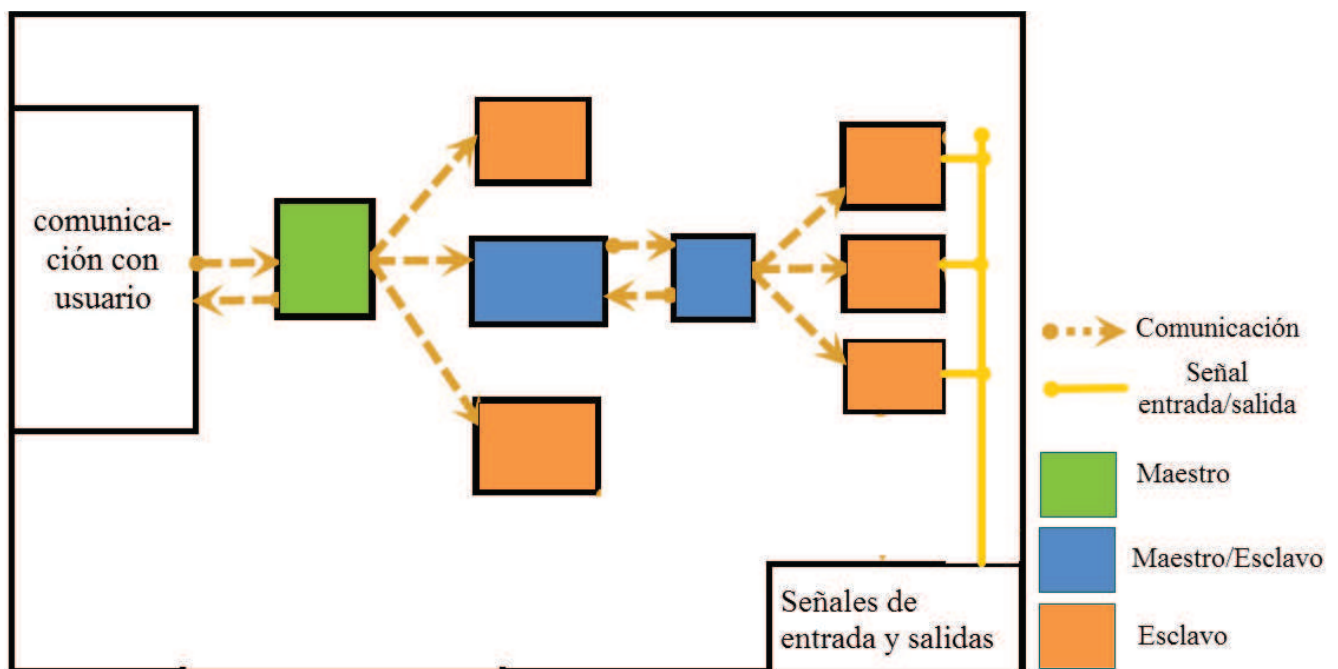


Figura 15. Estructura unidad de control

La estructura de comunicación entre los módulos que componen a la unidad de control debe tener una organización bien definida y particular para la aplicación con base en el proceso. Así, el análisis del proceso con base en componentes y su nivel jerárquico permite la identificación de los módulos cuando se plantea los bloques funcionales, además de que por ser una estructura flexible puede generarse tantos módulos maestros, maestro/esclavo y esclavo como sean necesarios, pero sin olvidar el costo de la comunicación y eficiencia de las tareas.

Cabe señalar que, para el caso de la implementación con microcontroladores, una estructura jerarquizada basada en la conceptualización del proceso permite una rápida identificación de las divisiones modulares debido a que se conoce las capacidades y limitaciones del hardware, pero a su vez se conoce la potencialidad de trabajar bajo módulos en cuanto a comunicación de tareas que va desde la entrada de usuario hasta la salida de señales considerando las entradas.

2.4.4 Modelo BITBUS de comunicación

Existen múltiples alternativas para la interconexión eléctrica de los dispositivos de control como enlaces analógicos, soluciones híbridas y sistemas completamente digitales. La topología básica de la red consiste en un simple bus lineal previsto para una estructura lógica del tipo maestro/esclavo, este conjunto debe considerarse a todos los efectos como un sólo bus con expansiones. Se define como **bus** a un sistema que trasfiere datos entre componentes de tal forma que organiza y da jerarquía para integrar los componentes en un único proceso [24].

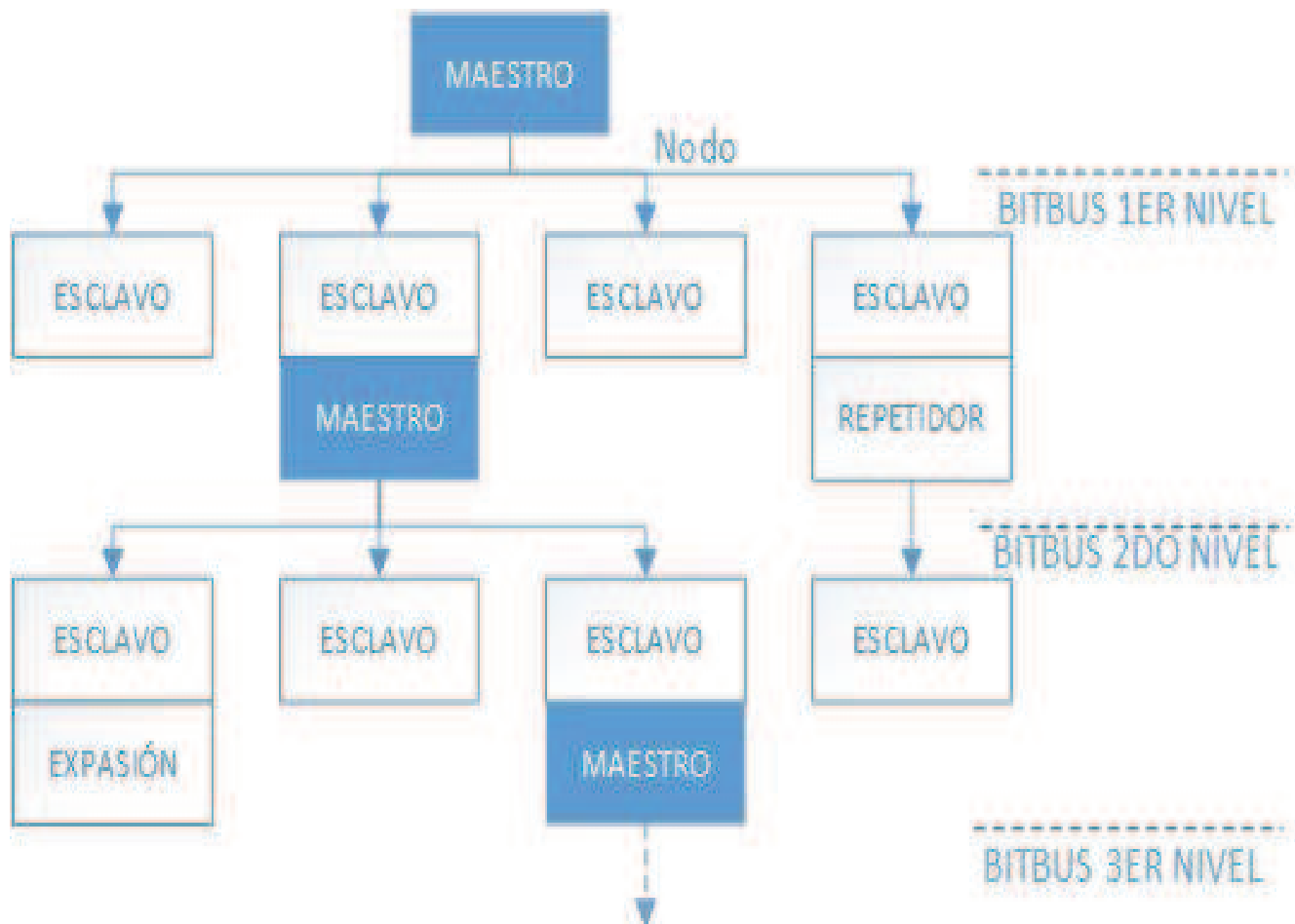


Figura 16. Arquitectura BITBUS

Obsérvese en la Figura 16 que en la estructura en árbol, donde los repetidores simplemente proporcionan una expansión del bus y regeneran la señal de datos, cada red maestro/esclavo debe ser considerada como un bus independiente que maneja direcciones propias y admite que los buses de distintos niveles operen a distintas velocidades. Esta estructura representa una distribución común para la arquitectura **BITBUS**.

El elemento básico de la arquitectura BITBUS es el nodo. Un nodo es cualquier dispositivo que contenga al menos una interfaz con el bus. Dentro de un nodo BITBUS se encuentra cuatro tipos de bloques básicos:

- Bloque maestro. Este bloque permite controlar el acceso a un bus de cualquier nivel mediante un simple mecanismo de pregunta/respuesta con los esclavos de dicho nivel. En general existe un sólo maestro en cada nivel de bus.
- Bloque esclavo. Cualquier bloque controlado por un maestro, por tanto sin capacidad de transmitir por iniciativa propia, es un bloque esclavo. Su función suele ser enlazar con un bloque de expansión de E/S.
- Bloque repetidor. Un bloque repetidor es un bloque utilizado simplemente para regenerar la señal de datos del bus, no del reloj en modo síncrono o asíncrono. Los bloques repetidores disponen de una interfaz que permite conmutar el sentido de la transmisión.
- Bloque de expansión de E/S. Este bloque constituye la interfaz con el proceso y puede consistir en simples entradas o salidas analógicas o digitales o dispositivos más complejos como bancos de memoria, discos, etc.

Un nodo de BITBUS puede contener uno o varios de los bloques citados anteriormente. Así, por ejemplo, el nodo maestro de nivel 1, Figura 16, puede contener expansión de E/S o no y un nodo esclavo de nivel uno puede contener un bloque maestro para gestionar una red de nivel dos.

Para la sincronización de bits existen dos modos posibles que condicionan en cierto modo la estructura física de la red y la forma de codificación.

- a) Modo síncrono: en este modo se transmiten los datos a través de un par trenzado y el reloj mediante otro par adicional; se precisan, por tanto, siempre dos pares trenzados, no se admiten repetidores y la estructura del bus es puramente lineal con un esquema de conexión en paralelo. Como por ejemplo la comunicación I2C.
- b) Modo auto reloj: en este modo cada nodo genera su propio reloj sincronizando con la línea de datos. Como por ejemplo la comunicación RS232.

BITBUS se basa, como se ha dicho, en una estructura lógica del tipo maestro/esclavo; por tanto, no admite el dialogo directo entre dos esclavos sin pasar por el maestro. La comunicación se establece siempre por iniciativa del maestro y se basa en pares de mensajes del tipo pregunta/respuesta o del tipo orden/reconocimiento. Así pues, tanto el acceso al medio como al enlace lógico están enteramente bajo el control del maestro.

CAP. 3 DISEÑO CONCEPTUAL

Para el diseño conceptual se considera parte de la metodología orientada al producto para determinar las especificaciones y el concepto a desarrollar de la unidad de control. Es importante mencionar que el desarrollo se basa en un trabajo previo del equipo del proyecto, donde se analizó el proceso y se desarrolló una estructura mecánica a la cual se adapta el presente trabajo. El enfoque de diseño modular es continuo y se hace mención al sistema secuencial original durante el desarrollo del diseño con fin de comparar el sistema modular, establecer condiciones y especificaciones de diseño.

3.1 Estado de proyecto

El proyecto “**Lodos**” tiene como objetivo el diseño y la implementación de un sistema automatizado para prueba de polímeros que realiza la empresa SNF FLOERGER DE MÉXICO y que permita simular el mezclado en planta, Apéndice A. De forma general el proceso de la prueba de jarras está definido en tres partes: la preparación del floculante, la prueba de floculante con efluente y el análisis de pruebas, Figura 17.

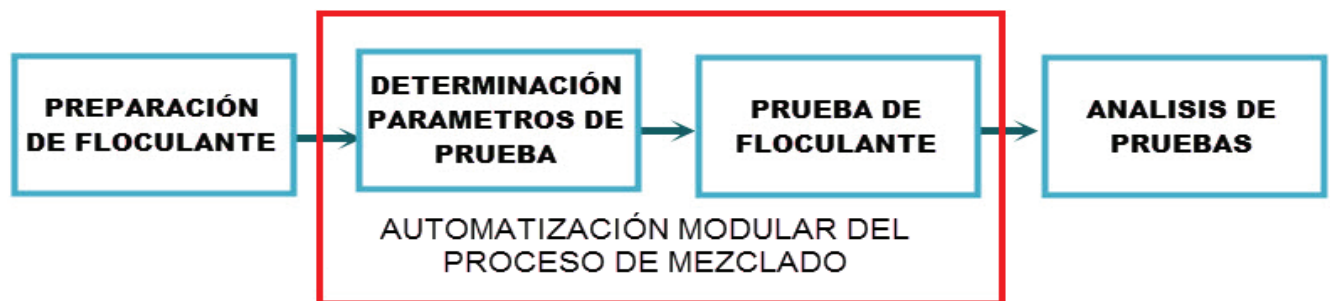


Figura 17. Ciclo del proceso de prueba de jarras

De dicha prueba, es de interés para la empresa automatizar la prueba de sus floculantes para tener un control sobre los parámetros como son: tiempos, velocidades, la cantidad de sustancia, etc., para mejorar la eficiencia, la precisión y el tiempo de sus pruebas. Este híbrido entre la prueba de laboratorio y la de planta permite asegurar la eficacia del floculante, pero estos alcances quedan fuera del presente trabajo, así se enfoca en el mezclado para las pruebas.

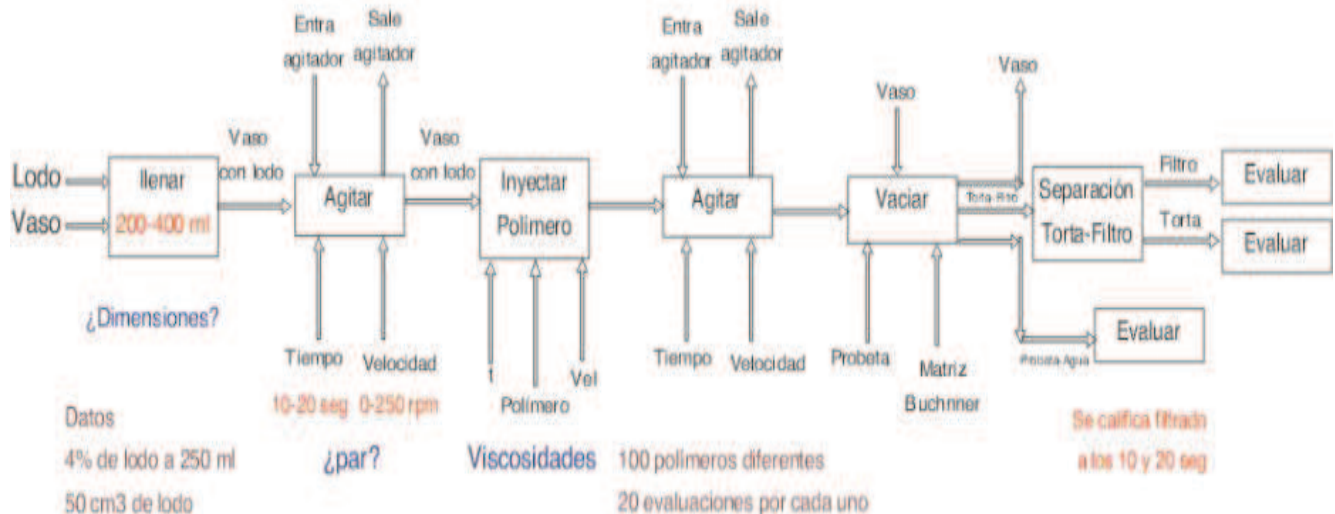


Figura 18. Algoritmo del proceso de prueba de jarras, Apéndice A

Durante la planeación del proyecto, trabajo previo, se realizó un análisis sobre el proceso de laboratorio y las formas de automatizarlo, se realizaron visitas a planta y se realizaron pruebas en ambiente controlado. Del análisis se estableció un algoritmo del proceso de la prueba de jarras, Figura 18, donde se observa la distribución de las tareas del proceso y se muestran los parámetros de las mismas, Apéndices A y B. En la Figura 19 se muestra el borrador planteado para la integración de los sistemas del proyecto, se tomaron en cuenta las tareas comunes de tal forma que se pudieran agrupar en sistemas independientes, estos sistemas propuestos fueron planteados como:

- Sistema de llenado: llena un contenedor con una muestra de lodo.
- Sistema de agitado: simula la turbulencia para hacer la mezcla.
- Sistema voltea: la mezcla tiene que ser transportada del contenedor donde se agita a un filtro para su análisis.
- Sistema de lavado: los residuos de las mezclas deben ser eliminadas de cada contenedor para tener muestras controladas.
- Sistema sube: posiciona verticalmente a la batidora para el mezclado.
- Sistema de inyección: inyecta el polímero con diferentes concentraciones.
- Sistema carrusel: posiciona contenedores, integra a los demás sistemas.

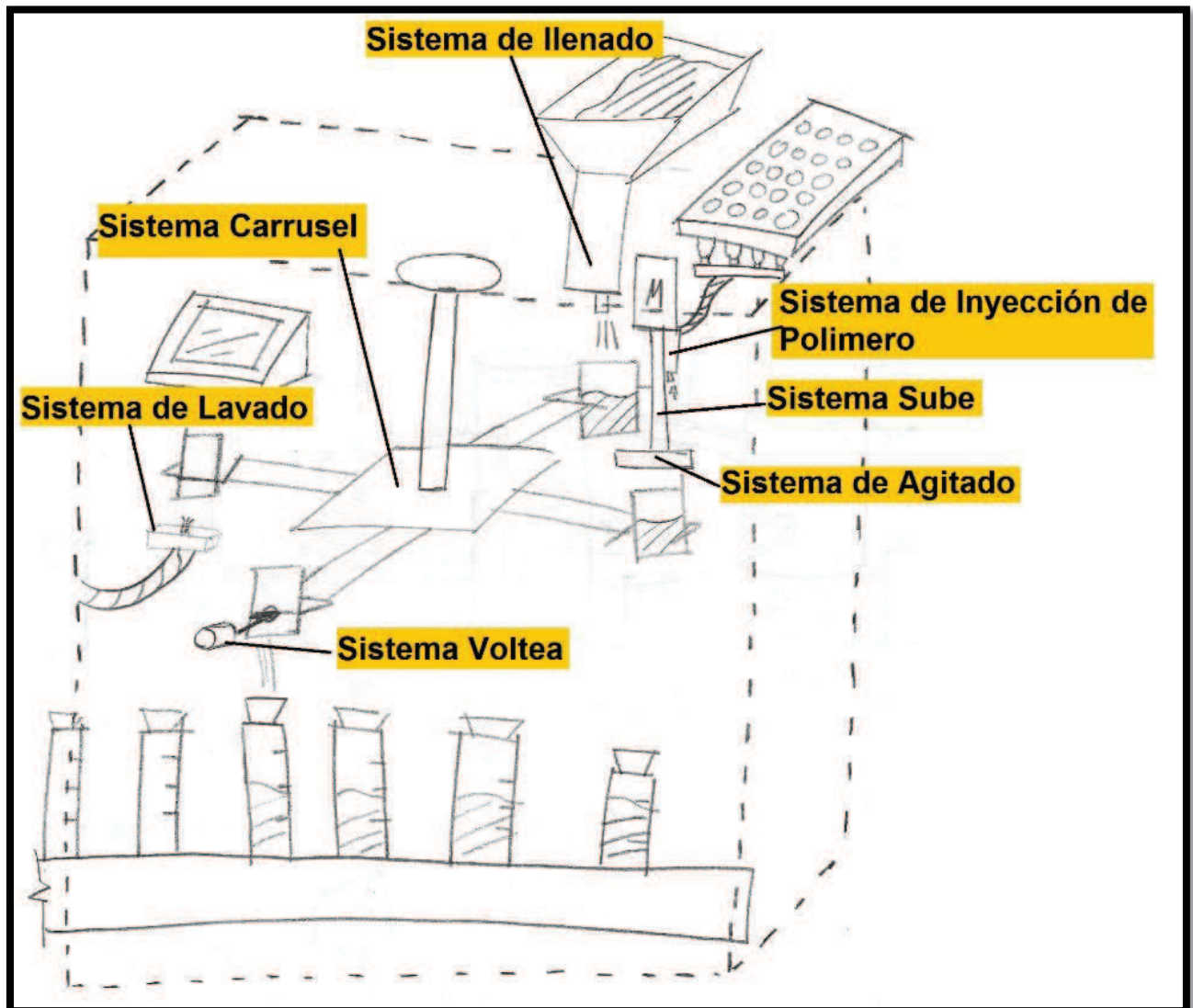


Figura 19. Distribución de sistema mecánico propuesto, Apéndice A

Cabe mencionar que el último sistema no forma una tarea del proceso, si no es el soporte de los demás, es decir, se considera como uno debido a que hace referencia al usuario durante todo el proceso, pues la estructura posiciona el contenedor hacia los demás para su interacción.

La planeación del proyecto está dispuesto para trabajarse en cinco etapas flexibles. Estas etapas fueron consideradas por los tiempos de manufactura, de implementación, de costos y además se considera que el desarrollo del proyecto representa un trabajo sobre una plataforma relativamente nueva en el mercado; de esta forma se determinaron las siguientes etapas:

- Etapa 1: posicionamiento.
- Etapa 2: agitado.
- Etapa 3: dosificación.
- Etapa 4: lavado.
- Etapa 5: recolección y análisis de muestras.

Cada una de las etapas anteriores involucra uno o más de los sistemas que fueron analizados previamente, de tal forma que durante su desarrollo se permita cambios para su adaptación en el tiempo del desarrollo. Por otra parte, se tiene la configuración mecánica del sistema carrusel y sistema voltea, Figura 20. De este punto y fecha parte el desarrollo de esta tesis, sobre el diseño y la implementación electrónica por módulos para la automatización del proceso en las etapas 1,2 y 3.

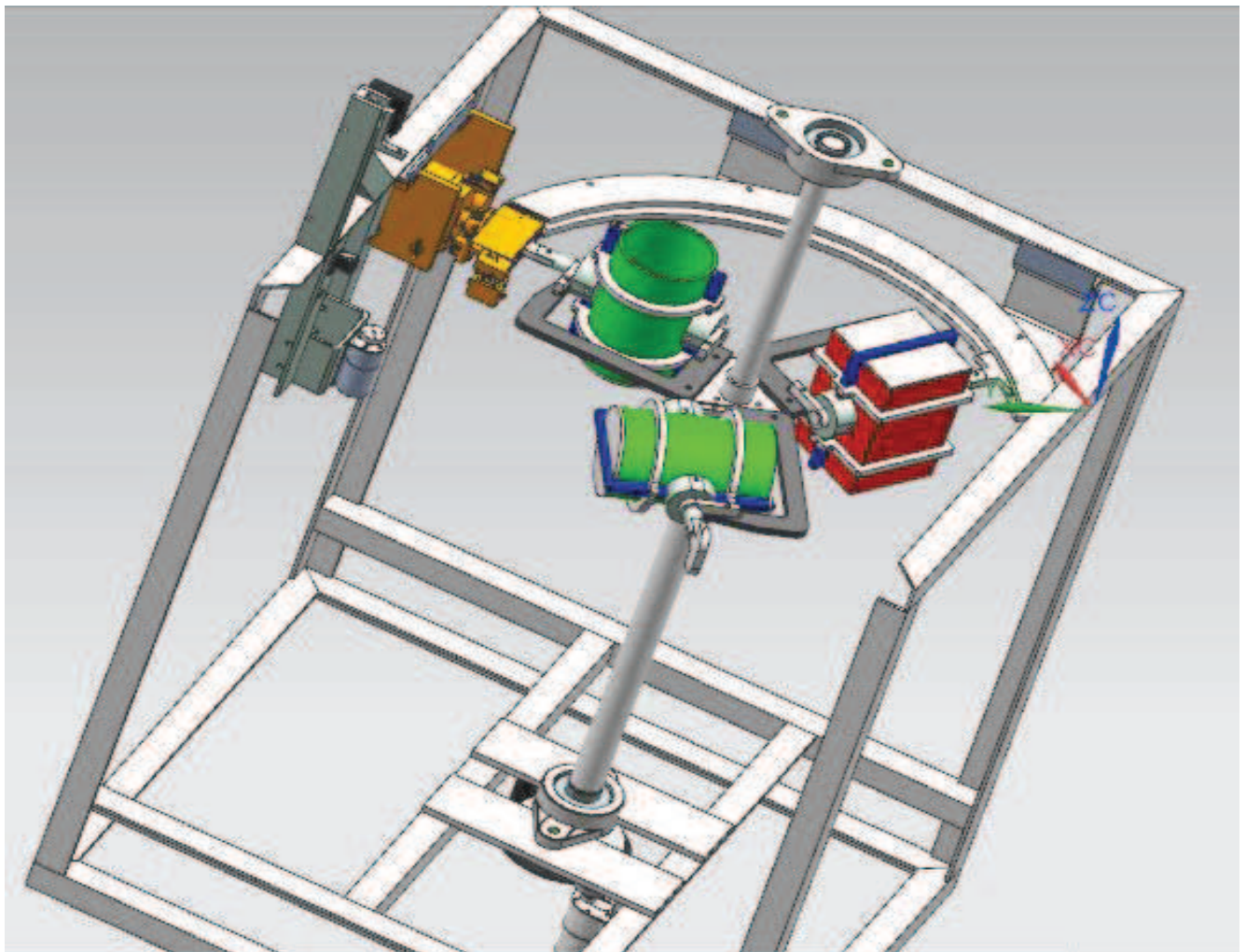


Figura 20. Etapa 1 del proyecto, sistema mecánico, CAD proyecto LODOS, estructura carrusel, diciembre 2013

3.1.1 Planeación de trabajo

Como parte de la integración al desarrollo del proyecto, se debe considerar la planeación en el grado de ejecución y participación, para poder adaptarse a la estructura de organización, asignar prioridades, darle continuidad a trabajos y dar resultados. Para poder determinar un resumen de trabajo, se va a analizar para el desarrollo de la planeación los siguientes puntos:

- Metas.
- Organización.
- Tiempo.
- Costos.
- Actividad en proyecto.

El **objetivo general** de este trabajo sobre el proyecto LODOS es la de automatizar el proceso de mezclado, de las etapas 1, 2 y 3, mediante el concepto de módulos, tanto a nivel software como hardware y a nivel de proceso, de la unidad de control. Se fundamenta a partir de que las **metas** del proyecto de hacer un sistema móvil, reducir tiempos para optimizar las pruebas y llevar registro de éstas para su posterior análisis con el fin de obtener el mejor polímero para el tratado de efluente.

En la Figura 21 se muestra la **estructura de organización** del proyecto, la cual está dada por la dirección y los equipos de trabajo de electrónica y de mecánica. Ambos equipos tienen en común la manufactura por lo que se espera hacer un avance en conjunto para reducir tiempos de pruebas e identificar cambios oportunos en el diseño, en la planeación y en la implementación.

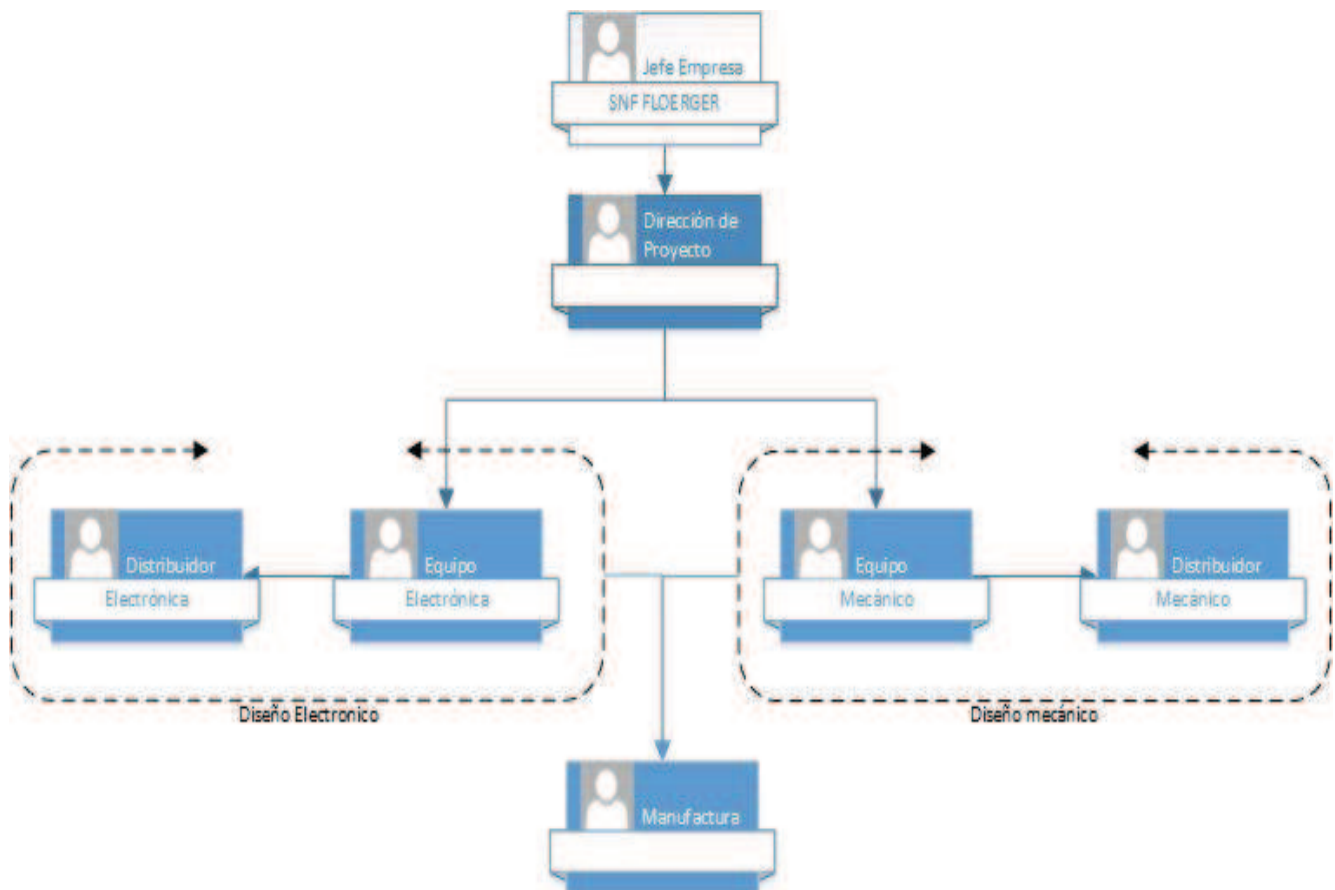


Figura 21. Organización del equipo de trabajo

Para los **costos**, se considera el material para desarrollo de los sistemas electrónicos como actuadores, sensores, elementos de acondicionamiento y microcontroladores; se toman en cuenta el **tiempo** de la implementación, la funcionalidad y su disponibilidad en el mercado para poder reducir tiempos; estos tiempos son de prioridad para el desarrollo de las etapas.

La participación en el diseño mecánico quedará limitada para la adaptación de los sistemas electrónicos y de la unidad de control a la estructura mecánica por lo que el impacto será menor. Los presupuestos disponibles serán dados al inicio de cada etapa por lo que se debe considerar que para los avances en cada una debe determinarse todos los elementos necesarios para su desarrollo y sólo considerar modificaciones menores posteriores.

Las funciones a desempeñar dentro del proyecto intervendrán directamente, para el desarrollo de la unidad de control del sistema y de los periféricos de entrada y salida, en:

- Costos: material.
- Planeación: sistemas y soluciones.
- Implementación electrónica: adaptación a la secuencia de control.
- Manufactura: adaptación de sistemas electrónicos.

Por tanto el área de trabajo estará considerada en el diseño electrónico sin participar en el diseño mecánico debido a que su desarrollo pertenece a otro equipo de trabajo. Con base en el análisis anterior, se puede estructurar una síntesis del planteamiento del proyecto, Tabla 4.

Tabla 4. Planteamiento del Proyecto Lodos	
Descripción de proyecto	• Máquina para simular mezclado de floculante y efluente en planta
Propuesta de valor	• Sistema automatizado • Control y monitoreo de datos
Metas clave	• Flexibilidad del proceso • Registro de pruebas • Robusto
Mercado primario	• Proveedores de floculantes • Empresas con plantas de tratamiento
Mercado secundario	• Laboratorios
Suposiciones	• Automático • Electrónico • Interfaz usuario máquina • Innovador
Involucrados	• Operador • Departamento de ingeniería • Plantas de tratamiento

De este cuadro se puede entender que los equipos de trabajo deben mantener una comunicación dinámica en el inicio y fin de las etapas del proyecto para permitir el paralelismo en el desarrollo. Además de que las posibles estrategias a considerar del equipo de trabajo pueden ser de intercambio de estados de desarrollo, pruebas en conjunto, desarrollo a la par en la manufactura, etc. También se

define el mercado objetivo y las suposiciones que harán los grupos de desarrollo para planear los servicios que se requieran, como manufactura o proveedores.

3.1.2 Definición de Fases de trabajo

El desarrollo de trabajo se limita para la electrónica y el control. Las principales herramientas serán de análisis de proceso basado en componentes y la estructuración de algoritmos para dar una visión general del proceso, darle modularidad y para identificar mejoras o reducción de pasos y facilitar la implementación. Mientras que la implementación será un resultado consecuente del análisis.

Tomando en cuenta la disponibilidad de material en mercado, la implementación y tiempos de desarrollo con el equipo de trabajo, se establece la necesidad de trabajar por Fases que integren las etapas, de tal forma que puedan desarrollarse al inicio de cada etapa un seguimiento de la anterior para adaptar de forma productiva los cambios que surjan de forma paralela. Se muestra en la Tabla 5 la distribución de los sistemas para las etapas del proyecto, se cambian los nombres de los sistemas asociándoles directamente la característica de la función a desarrollar.

Tabla 5. Distribución sistemas por cada etapa		
Etapas	Característica	Sistema
1	Posicionamiento de estructura general	Sistema carrusel Sistema de orientación de contenedores
2	Agitado	Sistema de agitado Sistema posicionamiento de batidora
3	Dosificación	Sistema dosificación de floculante Sistema dosificación de efluente
4	Lavado	Sistema lavado
5	Análisis de muestras	Sistema de recolección de muestras

Queda de forma conveniente estructurar el desarrollo de las etapas 1,2 y 3 en Fases que permitan distribuir los tiempos del trabajo sobre cada sistema sin alterar las entregas de cada etapa. El plan de trabajo inicia con la realización de un análisis general del proceso para determinar las etapas de sensado, planeación y actuación del proceso de producción. Los sistemas electrónicos, módulos, se plantean en cada fase. Finalmente se plantea la prueba como modelo funcional.

Las Fases de trabajo quedan definidas como:

- Arquitectura y modularidad del proceso: análisis del proceso de operación basado en componentes y definición del sistema de control basado en módulos.
- Control de estructura: integra el control del sistema carrusel, posicionamiento de contenedores y de batidora.
- Interfaz de usuario y de módulos: diseño de interfaz de usuario intuitiva y almacenamiento de historial de pruebas. Interfaz entre módulos orientada a eventos.
- Control de batidora: control de velocidad por retroalimentación de estado.
- Dosificación floculante y efluente: dosificación con bajo error de repetitividad para asegurar la calidad de las pruebas.

El plan de trabajo queda de forma lineal, Figura 22, se muestra la ventaja de fragmentar las etapas del proyecto en Fases para que pueda haber paralelismo en la integración de los sistemas de forma continua y sin retrasar el avance de las siguientes.

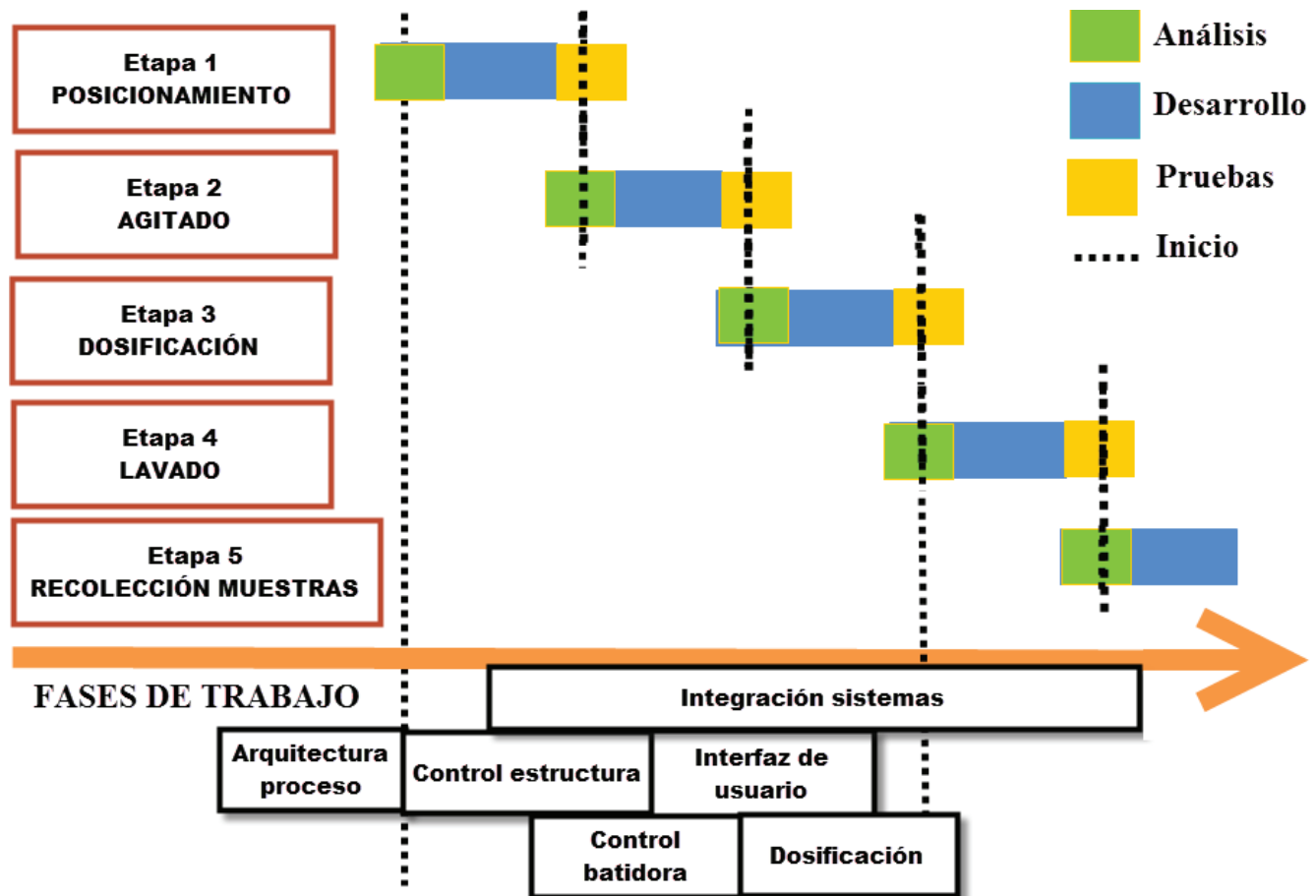


Figura 22. Fases de trabajo

Cabe mencionar nuevamente que un factor importante es el tiempo, por lo que al extender las Fases entre etapas se permite la flexibilidad de cambios al inicio y fin de cada una dándole prioridad a las tareas de los sistemas, pero también debe recordarse que el equipo mecánico mantiene cambios conforme el avance de cada etapa.

Tomando en cuenta los conceptos de proceso y de automatización, cada Fase está dispuesta para trabajar con la siguiente estructura:

- Diseño auxiliado por Grafcet: análisis de funciones del proceso y tareas
- Generación bloques funcionales: reducción de subprocessos similares en diagrama flujo
- Acondicionamientos de entrada/salida : limitación y aseguramiento de entradas, salidas
- Módulos de sistema: definición de estructura de módulo por entradas, salidas y tareas

Se resuelve de forma individual cada módulo de los sistemas para los procesos de operación manteniendo los principios de modularidad de la unidad de control, por lo que no se pretende reducir la funcionalidad de la misma si no de cada módulo. De esta forma se permite avanzar con las etapas de forma ordenada, dándole el seguimiento a la integración de los sistemas y cumpliendo los objetivos de darle flexibilidad al proceso.

3.2 Necesidades y requerimientos

El proceso de identificar las necesidades del cliente es parte integral del proceso de desarrollo del producto y está más estrechamente relacionado con la generación de conceptos, la selección de los mismos y el establecimiento de especificaciones del producto [14]. Para el caso de este trabajo, se ha analizado anteriormente las etapas, los objetivos y metas para el proyecto, por lo que se puede establecer las necesidades que se plantearon para el desarrollo de la electrónica de control, estas son cuatro:

- A. Adaptable
 - a. Sistema que permita cambiar orden del proceso.
 - b. Modificación fácil del proceso (programación).
 - c. Tareas simultáneas para reducción de tiempo.
- B. Autónomo
 - a. Detección y solución de errores.
 - b. Repetitividad.
 - c. Configurable.
- C. Dispositivo comunicación
 - a. Control en parámetros.
 - b. Registro de actividades.
 - c. Informe de eventos.
 - d. Configuración de pruebas.
- D. Mezcla controlada
 - a. Exactitud de dosificación.
 - b. Limpieza de contenedores.
 - c. Limpieza de medios de dosificación.

Estas necesidades son generales para cada etapa debido a la continuidad entre estas y que todas van sobre el desarrollo de la unidad de control. Para valorar la jerarquía de las necesidades y establecer los requerimientos que implican se analiza la importancia en el proyecto, las prioridades y se considera fechas de desarrollo.

En la Tabla 6 se establece un orden de prioridad y su importancia en el desarrollo de las necesidades; éste análisis fue discutido por los equipos de trabajo. La prioridad está dada en un a escala de 1 a 4 en orden descendente, tomando en cuenta la funcionalidad y tiempo de desarrollo debido a que por cada avance en las etapas se debe mostrar el modelo funcional al equipo de trabajo.

Tabla 6. Prioridad de las necesidades		
Necesidad	Prioridad	Importancia relativa
Mezcla controlada	4	Crítico
Adaptable	3	Indispensable
Autonomía	2	Importante
Dispositivo comunicación	1	Deseable

Sobre la importancia relativa, se considera en cada uno de los niveles:

- Crítico: No se considera al proyecto sin esta función.
- Indispensable: Debe tener al menos una de estas funcionalidades.
- Importante: Es una característica.
- Deseable: Podría considerarse el funcionamiento sin ella.

No se toman en cuenta estados de “sin importancia” o “indeseable”, debido a que el desarrollo de este trabajo es de integración a un proyecto iniciado con un trabajo definido mecánicamente, por lo que sólo se da prioridad para el diseño electrónico, es decir, a partir de los puntos ya considerados por el equipo de trabajo se define cuales son más relevantes para el desarrollo de las funciones asignadas. Agrupando las necesidades en referencia a las etapas y las Fases de trabajo se tiene la Tabla 7, dónde se aprecia la distribución de las Fases de trabajo con los sistemas y sus necesidades:

Tabla 7. Distribución de Fases de trabajo				
Etap	Sistema	Necesidad	Fase de trabajo	
1	Sistema carrusel Sistema orientación contenedores	Au D Ad	Control estructura	Interfaz
2	Sistema posicionamiento batidora Sistema de mezclado	Ad Au C	Control estructura Control batidora	
3	Sistema dosificación lodo y polímero	C Au Ad	Dosificación	
Adaptable: Ad Dispositivo comunicación: D Autónomo: Au Mezcla controlada: C				

Nótese que la Fase de interfaz es general, debido a que debe estar continuamente adaptando la comunicación tanto para usuario/máquina como entre los mismos módulos. Además, la necesidad del “Dispositivo de comunicación” está directamente relacionado con el control de estructura, por lo que éste será la parte principal que acople los sistemas, como se analizó en cap. 3.1.

3.3 Especificaciones y limitaciones

Las especificaciones del producto no indican como manejar las necesidades pero representan una base para satisfacerlas [14]. Se usarán las necesidades y requerimientos para analizar los tipos de especificaciones, con esto se busca relacionarlos con elementos que puedan dar solución. En este punto, se debe recordar que estas especificaciones no son de carácter global en el proyecto. Así, estas son consecuencia de una reiteración del análisis de proyecto en cuanto a desarrollo de la unidad de control, por tanto, sólo atiende a las especificaciones de un área de trabajo.

Los puntos a considerar son métrica y prioridad. La prioridad será dada en orden descendente, siendo 1 el de mayor prioridad, la cual está en función del tiempo de desarrollo en relación con los otros equipos de trabajo. En la Tabla 8 se muestra las especificaciones y su relación con las necesidades, su métrica y prioridad. Estas especificaciones no provocaran un análisis para la generación del concepto debido a que ya se tiene el objetivo de desarrollo de la unidad de control y

la base fue planteada al inicio del proyecto para la etapa 1, Apéndice A, pero se usarán para poder determinar los sistemas modulares. Por otra parte, se debe tomar en cuenta las especificaciones anteriores a este trabajo en forma de limitaciones de diseño. En la Tabla 9 muestran las limitaciones.

Tabla 8. Especificaciones

Necesidad	Especificaciones	Métrica	Prioridad	Observaciones
Ad, Au	Repetitividad de pruebas	Iteraciones, #	1	Programación robusta
D	Variables del proceso	Número, #	2	Planeación de módulos, programación
Ad, C	Fuente de energía	Potencial eléctrico, [V]	3	Eliminar caídas de voltaje
Ad	Costos	Dinero, [\$MX]	4	Disponibilidad mercado
Ad, C	Acondicionamiento entrada/salida	Potencial eléctrico, [V]	5	Reducción ruido
Ad, C	Cantidad de componentes	Número, #	6	Funcionalidad y optimización
Ad	Tiempo de control de proceso	Tiempo, [s]	7	Programación eficiente
Ad	Error en sensor/actuador	Porcentaje, %	8	Ruido
Ad, D, C	Velocidad de procesamiento de datos	Ciclos máquina, [MHz]	9	Capacidad microcontrolador
C	Cantidad de dosificación	Volumen, [ml]	10	Efluente y floculante
C	Tiempo mantenimiento	Tiempo, [s]	11	Limpieza de dosificadores
Ad, D, C	Dimensiones módulos	Área, [cm ²]	12	Espacio mecánico disponible

Tabla 9. Limitaciones

Limitaciones	Elemento	Implicaciones	Observaciones
Interfaz de usuario	Pantalla touch	Programación	Requiere generar funciones de interfaz y set de comandos. Comunicación serial
Unidad de control	Microcontrolador Arduino	Programación, procesamiento	Adaptación programación y consideraciones de programación
Actuadores	Motores	Etapas potencia	Programación y control con base en especificaciones de operación
Sensores	Ópticos y de carrera	Acondicionamiento	Programación y control en función de sensores
Dimensiones	Estructura mecánica	Tamaño unidad control	Restricciones de espacio
Costos	Existencia de material	Diseño e implementación electrónica	Dispositivos de uso general. Distribuidores

Debido a que se planea una implantación por módulos se considera el uso de más componentes para obtener las metas de flexibilidad, eficiencia y robustez en el proceso para su implementación, por otra parte, la unidad de control quedará reducida por los procesos funcionales. Por lo que al inicio de las Fases de trabajo se determinará el número de dispositivos electrónicos para los módulos por medio de los bloques funcionales.

Para complementar las especificaciones, en la Tabla 10 se cuantifican y se agrega el grado de restricción, para poder considerar qué componentes electrónicos pueden implementarse y cuál será su impacto en el desarrollo del trabajo; este impacto se cuantifica como un factor de escala para el despliegue de la función de calidad, con tres grados: Alto, medio y bajo, con una valor ponderado de 1.1, 1.05 y 1.01 de agregado respectivamente para considerarlo en la matriz de calidad.

Tabla 10. Grado de restricción de las especificaciones			
Especificaciones	Cuantificación	Justificación	Grado de restricción
Velocidad de procesamiento	16 [MHz]	Ya se tienen microcontroladores a 16 [MHz] para las etapas 1 y 2.	Medio
Tiempo del proceso	Estimado 1 [min] por prueba	Menor o igual al tomado en pruebas de laboratorio.	Alto
Tiempo mantenimiento	Estimado 5 [min] preventivo y 2 [hrs] correctivo	No está considerado dentro del proceso, pero es necesario para el buen funcionamiento.	Bajo
Error en sensor/actuador	Porcentaje %	La estructura cuenta con dos tipos de encóder, posición e inicio/fin de carrera,	Alto
Configuración parámetros	6 parámetros	Parámetros básicos para configuración de iteraciones	Medio
Repetitividad de pruebas	0-99 [ml] por floculante	En laboratorio se utiliza 20 [ml] por polímero, pero varía según la experiencia del operario y condiciones	Medio
Cantidad de dosificación	0-20 [ml] de floculante 200-400 [ml] de efluente	Las pruebas en laboratorio requieren variar las concentraciones de ambas sustancias de acuerdo a características particulares y experiencia	Alto
Fuente de energía	12,7,5 [V]	Se requiere fuentes de alimentación para cada etapa de potencia, de acondicionamiento y de procesamiento para asegurar funcionamiento	Alto
Acondicionamiento entrada/salida	1/0 lógicos	Se encuentran establecidos los dispositivos de entrada y salida. Todo el procesamiento no deberá tener lecturas analógicas	Alto
Costos	Depende por sistema	Por cada etapa se dará un estimado de costo y un presupuesto	Alto
Número componentes	3 grupos	Los módulos de cada sistema deben tener grupo de acondicionamiento, un dispositivo de planeación y un grupo para acondicionamiento de la actuación	Bajo
Dimensiones módulos	---	La estructura mecánica no define el espacio para la unidad de control.	Medio

3.4 Despliegue de matriz de calidad

Después de determinar las necesidades, los requerimientos y las especificaciones con sus respectivas prioridades y limitaciones, discutidas por todo el equipo de trabajo, se recurre a la matriz de calidad de la metodología QFD para evaluar la organización del diseño basado en la jerarquía de las especificaciones, Figura 23.

De forma general la matriz de calidad permite organizar el orden de la prioridad de las especificaciones por lo que tiene impacto directamente en las necesidades del cliente de tal forma que para el desarrollo de la unidad de control deberá trabajarse en el planteamiento de la solución que satisfaga las necesidades con base en las especificaciones; esto implica que antes de la implementación debe hacerse un esfuerzo mayor en el análisis del proceso para considerar todas las posibles variantes de cada módulo.

El análisis puntual de la matriz de calidad, sobre el diseño orientado a especificaciones, permite hacer una integración a la planeación del desarrollo del diseño, agrupando las especificaciones con las Fases de trabajo. Por tanto de la matriz de calidad sólo se va a tomar en cuenta los siguientes puntos para organizar las especificaciones para el diseño:

- El orden final de la prioridad de las especificaciones en relación a las necesidades.
- Agrupación de las especificaciones dominantes y dependencia de las de menor prioridad.
- Distribución de prioridad en Fases de trabajo.

En la Tabla 11 se muestra las especificaciones con su prioridad y la necesidad de mayor peso.

Tabla 11. Especificaciones ponderadas por la matriz de calidad		
Especificación	Prioridad	Necesidad de mayor peso
Velocidad de procesamiento	1	Todas
Variables del proceso	2	Autónomo
Costos	3	Mezcla controlada
Acondicionamiento E/S	4	Mezcla controlada
Error en sensor/actuador	5	Mezcla controlada
Tiempo de mantenimiento	6	Mezcla controlada
Fuente de energía	7	Mezcla controlada, Adaptable
Tiempo de control de proceso	8	Mezcla controlada, Adaptable
Cantidad de componentes	9	Adaptable
Cantidad de dosificación	10	Mezcla controlada
Repetitividad de pruebas	11	Autónomo
Dimensiones de módulos	12	Adaptable

Se reduce el análisis a estos puntos debido a que el diseño de la unidad de control será para una solución del sistema mecánico del proceso, con especificaciones vistas como limitaciones, por lo que no se espera tener variantes en los sistemas durante el desarrollo de las etapas del proyecto en cuanto a especificaciones o a modelos de funcionamiento

Escala Req/Esp			DESPLIEGUE MATRIZ DE CALIDAD QFD																
IMPORTANCIA	+	9	Extremo	Prioridad de Necesidades	ESPECIFICACIONES	Métrica	Iteraciones, #	Número, #	Potencial eléctrico, v	Dinero, \$MX	Potencial eléctrico , v	Número, #	Tiempo, s	Porcentaje, %	Ciclos maquina, MHz	Volumen, ml	Tiempo, s	Área, cm^2	
		8	Fuerte																
		7	Moderado																
		6	Ligero																
		5	neutro																
		4	Ligero																
		3	Moderado																
		2	Fuerte																
		1	Extremo																
		0	no depende																
Factor escala restricción																			
1,1	A	Alto																	
1,05	M	Medio																	
1,01	B	Bajo																	
Prioridad necesidad																			
4		Crítico																	
3		Indispensable																	
2		Importante																	
1		Deseable																	
Prioridad Especificacion					1	2	3	4	5	6	7	8	9	10	11	12			
NECESIDADES	REQUERIMIENTOS CLIENTE				M	M	A	A	A	B	A	A	M	A	B	M			
Mezcla controlada	Exactitud de dosificación		4		0	6	9	9	9	1	0	9	9	9	5	0			
	Limpieza de contenedores		4		0	5	4	7	1	0	1	0	6	0	9	0			
	Limpieza medio dosificación		4		0	1	4	7	7	7	9	9	6	0	9	3			
Adaptable	Modulo flexibe		3		1	9	0	7	3	6	0	0	9	0	0	4			
	Modificación de proceso		3		0	9	0	0	3	0	1	0	9	0	0	2			
	Tareas simultaneas		3		0	9	4	6	0	9	7	3	9	0	1	1			
	Alimentación 127 Vrms		3		0	0	9	6	3	3	0	0	0	0	0	3			
Autónomo	Detección/solución error		2		6	2	0	0	9	4	4	9	7	5	5	0			
	Repetitivo		2		9	9	0	0	9	0	6	9	9	0	0	0			
	Configurable por usuario		2		3	9	0	0	0	0	0	0	9	0	0	0			
Dispositivo comunicación	Control parametros		1		2	9	0	0	9	1	0	7	9	0	0	0			
	Registro actividades		1		1	9	0	0	1	0	7	4	9	0	4	0			
	Informe eventos		1		2	4	0	0	7	0	1	0	9	1	1	0			
Relacion Especificación/Necesidad					44	191	107	149	148	95	92	128	242	47	110	42			
Media promedio					16	71	40	55	55	35	34	47	90	17	41	16			
Consideración de restricción					16	74	44	61	60	36	37	52	94	19	41	16			
Prioridad necesidad/especificación					11	2	7	3	4	9	8	5	1	10	6	12			
Cambio en orden de prioridad					↓		↓	↑	↑	↓	↓	↑	↑		↑				

Figura 23. Matriz de calidad

Para agrupar las especificaciones con base en la relación entre ellas por su prioridad se tomará en cuenta las preguntas base para desarrollo de soluciones: cómo, con qué y con cuánto; por otra parte, el factor del tiempo va considerado desde la planeación por lo que ya es una prioridad general del proyecto. Estas tres preguntas permiten organizar las especificaciones por grupos jerárquicos sin quitar su prioridad resultante de la matriz de calidad, Tabla 12.

Tabla 12. Especificaciones agrupadas por jerarquía		
Preguntas	Dominante	Consecuente
Cómo	Velocidad de procesamiento	Repetitividad de pruebas
		Variables de proceso
		Tiempo mantenimiento
		Tiempo control actuadores
Con qué	Acondicionamiento e/s	Error sensor/actuador
		Cantidad de dosificación
Con cuánto	Costos	Cantidad de componentes
		Fuente de energía
		Dimensiones módulos

La distribución de las especificaciones muestra un orden coherente y jerárquico considerando que se trata del desarrollo de diseño electrónico de una unidad de control, sin tomar en cuenta el diseño mecánico. La Tabla 12 permite establecer un orden de las especificaciones para su consideración en el diseño.

3.5 Definición del tipo de sistema modular

El proceso de desarrollo del concepto implica una distinción entre necesidades del cliente y especificaciones del producto, esta distinción es sutil pero importante. Las necesidades son independientes de cualquier producto, por otra parte, las especificaciones dependen del concepto que se seleccione, sobre lo técnico y económicamente factible y de lo que se ofrezca en el mercado, en estrecha relación las necesidades del cliente. [14].

Hasta ahora se ha analizado las especificaciones para el desarrollo de los módulos de los sistemas y los puntos clave para el desarrollo y cumplimiento de las necesidades del cliente; ahora, con base en las especificaciones y el previo análisis del proyecto se conceptualiza el tipo sistema modular de la unidad de control UC en los siguientes puntos:

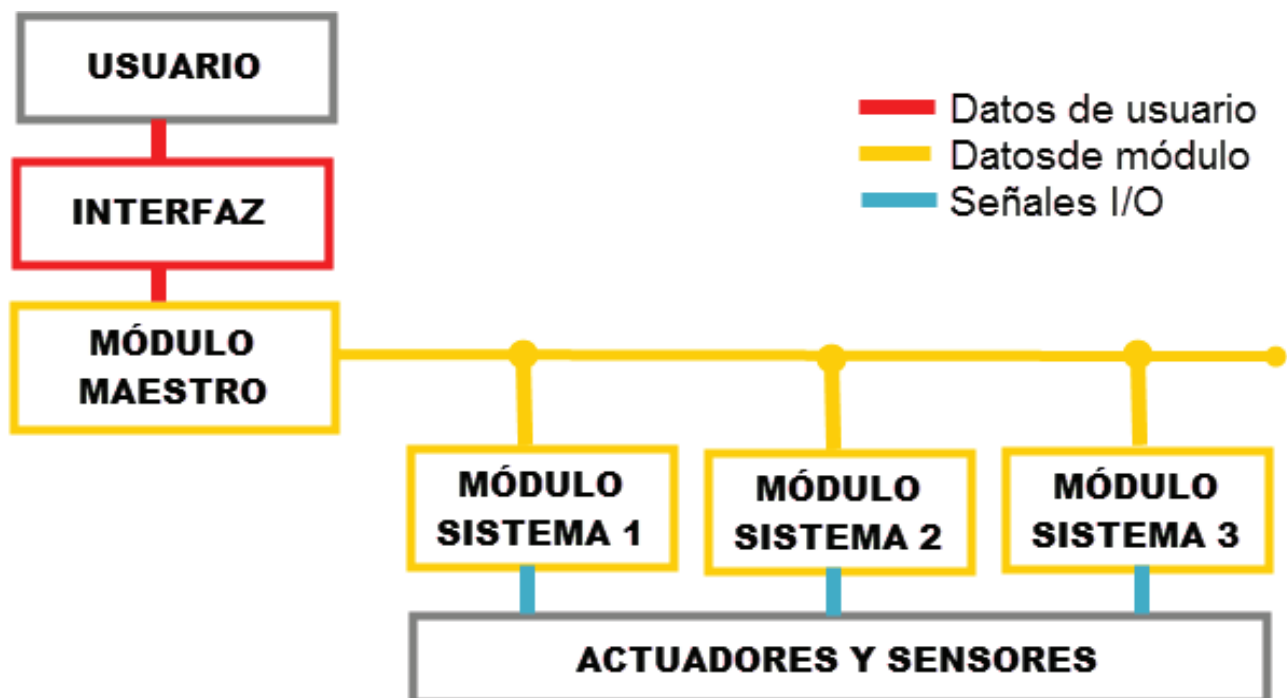
- Se requiere una UC modular para los sistemas mecánicos e interfaz de usuario para su gestión y control en la realización de la prueba de jarra a nivel planta; la planeación del desarrollo de los módulos se basa en el tiempo de desarrollo de las etapas del proyecto mediante Fases de trabajo.
- Las limitaciones en cuanto a la implementación generan especificaciones complementarias a nivel de concepto para la solución general, estas limitaciones ya fueron consideradas en las especificaciones y se mencionan en el cap. 3.3.
- El concepto inicial está propuesto para que existan en la unidad de control una interfaz de usuario y un control de sistemas, estos se describen en la Tabla 13.

Tabla 13. Concepto inicial de la unidad de control

Concepto	Objetivos
Interfaz de usuario	Parámetros de proceso
	Muestra de estado de sistemas
	Prueba de sistemas
	Reporte de eventos
	Configuración de sistema
Control de sistemas	Proceso secuencial
	Permitir cambios en el proceso
	Detección de errores
	Mantenimiento de sistema
	Robusto

De las especificaciones y limitaciones originales se generará un modelo del concepto de los módulos de la UC basado en las especificaciones para el presente trabajo y en jerarquía de componentes del sistema mecánico, el diseño de este modelo se desarrolla mediante el análisis del proceso por componentes y de la herramienta Grafcet para las funciones de operación de cada módulo, en una estructura tipo Bitbus para cambiar el concepto secuencial a multitareas y en un modelo deliberativo con base en la necesidad de robustez tanto en el sensado, en la planeación y en la actuación. El concepto implica una estructura de organización de comunicación en paralelo en cuanto a los módulos del sistema mecánico, mientras que para la interfaz de usuario implica una organización serial uno a uno.

La interfaz de usuario junto con el control de sistemas genera una UC especializada tanto en entrada/salida de señales como en flujo de datos, por tanto el uso del concepto de modularidad para la UC permite dar flexibilidad a la estructura de organización de los módulos además permite al proceso flexibilidad de operación y de realizar multitareas basado en un modelo tipo BITBUS de comunicación.

*Figura 24. Concepto de la estructura de comunicación*

Se puede plantear la estructura organizacional de comunicación como una combinación serial-paralelo para cumplir con los objetivos del concepto inicial, sus limitaciones y las especificaciones planteadas, debido a que la pantalla sólo permite comunicación punto a punto y para realizar multitarea se requiere comunicación en paralelo. Por tanto, la estructura de comunicación de la UC queda en dos grupos: usuario/interfaz e interfaz/sistemas. Este concepto de comunicación se ilustra en la Figura 24.

La estructura de la comunicación debe plantearse de forma generalizada, debido a que permite que los módulos no dependen de un caso particular de comunicación para tener mayor control en cómo se envían los datos o cómo se gestionan las tareas; por otra parte, el módulo maestro puede tener las mismas características para poder comunicarse con la interfaz.

Una de las limitaciones de la comunicación paralela en forma maestro/esclavo es que limita la línea de comunicación debido a que se comunica un elemento a la vez. Puede plantearse una estructura multimaestros pero para este tipo de sistemas basado en un proceso discreto y por el uso de módulos, la línea de comunicación no requiere transferir datos propios de cada tarea por lo que puede agregarse una línea auxiliar a la comunicación para indicar el estado de la ejecución de las tareas de los módulos. Una línea auxiliar de estados permite una independencia en la comunicación para planificar las tareas de cada módulo e identificar la detección de fallos, por lo que puede considerarse dentro de la línea de comunicación.

Se muestra en la Figura 25 la organización de las líneas de datos y señales dentro de la UC, basados en el modelo Bitbus de comunicación. Esta disposición permite establecer la necesidad de jerarquía y de organización entre los módulos, así la parte operativa de la UC está sobre la interfaz mientras que la parte de ejecución está en los módulos de los sistemas.

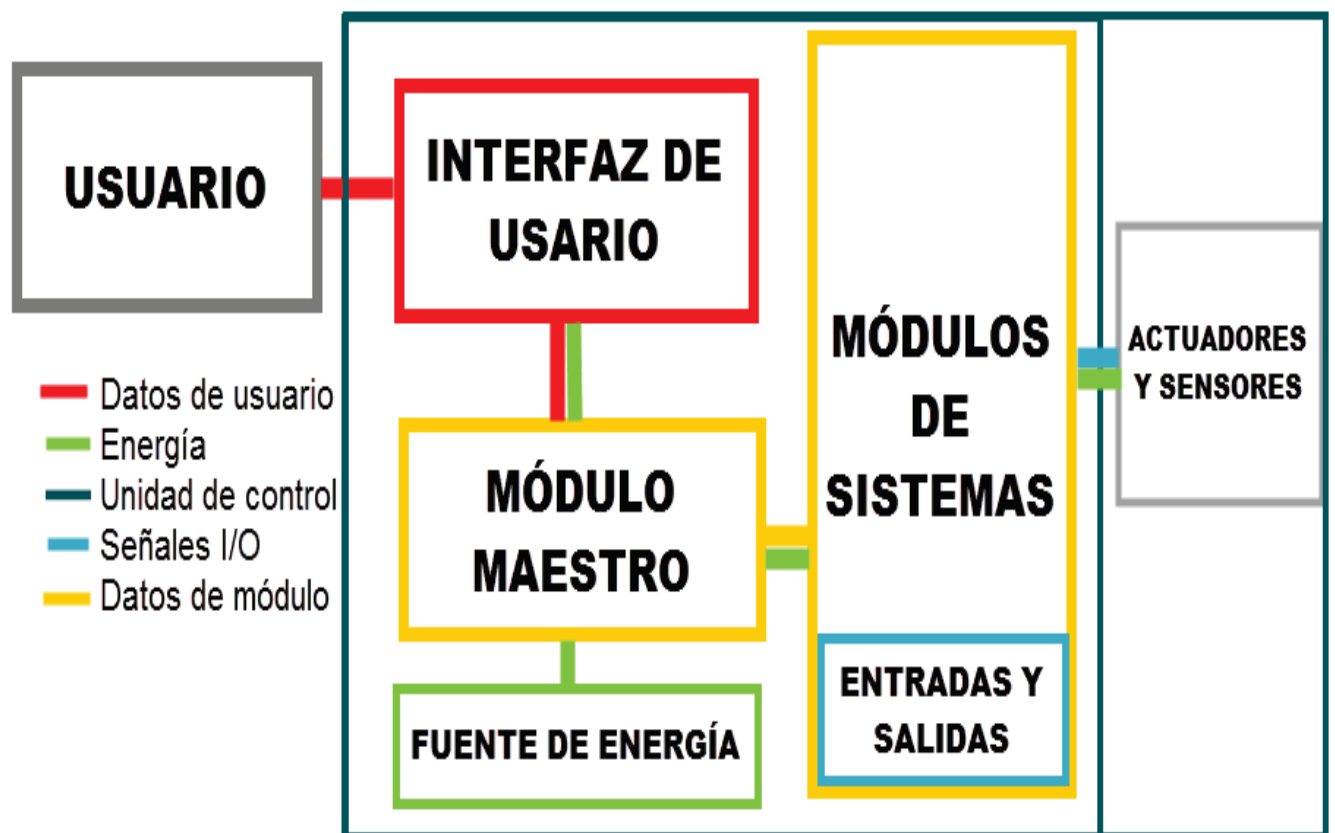


Figura 25. Concepto de estructura de organización de la unidad de control

Dentro de la comunicación, la estructura de los módulos debe permitir una comunicación y organización general, por lo que se requiere una base genérica de cada módulo, que permita la flexibilidad tanto de implementación y ejecución, como de operación, de mantenimiento y de comunicación. Esta flexibilidad permite que no se particularice el diseño de los módulos y pueda concentrarse los esfuerzos en el proceso y no en la comunicación.

El módulo, tanto para proceso como de interfaz, debe tener las siguientes características generales para la implementación:

- Acondicionamiento entradas/salidas.
- Operaciones de comunicación generalizada.
- Información de estados de error.
- Indicadores de funcionamiento.

Estos elementos permiten al módulo maestro identificar errores y estados para mostrarlos al operador así como dar organización a la forma de comunicación.

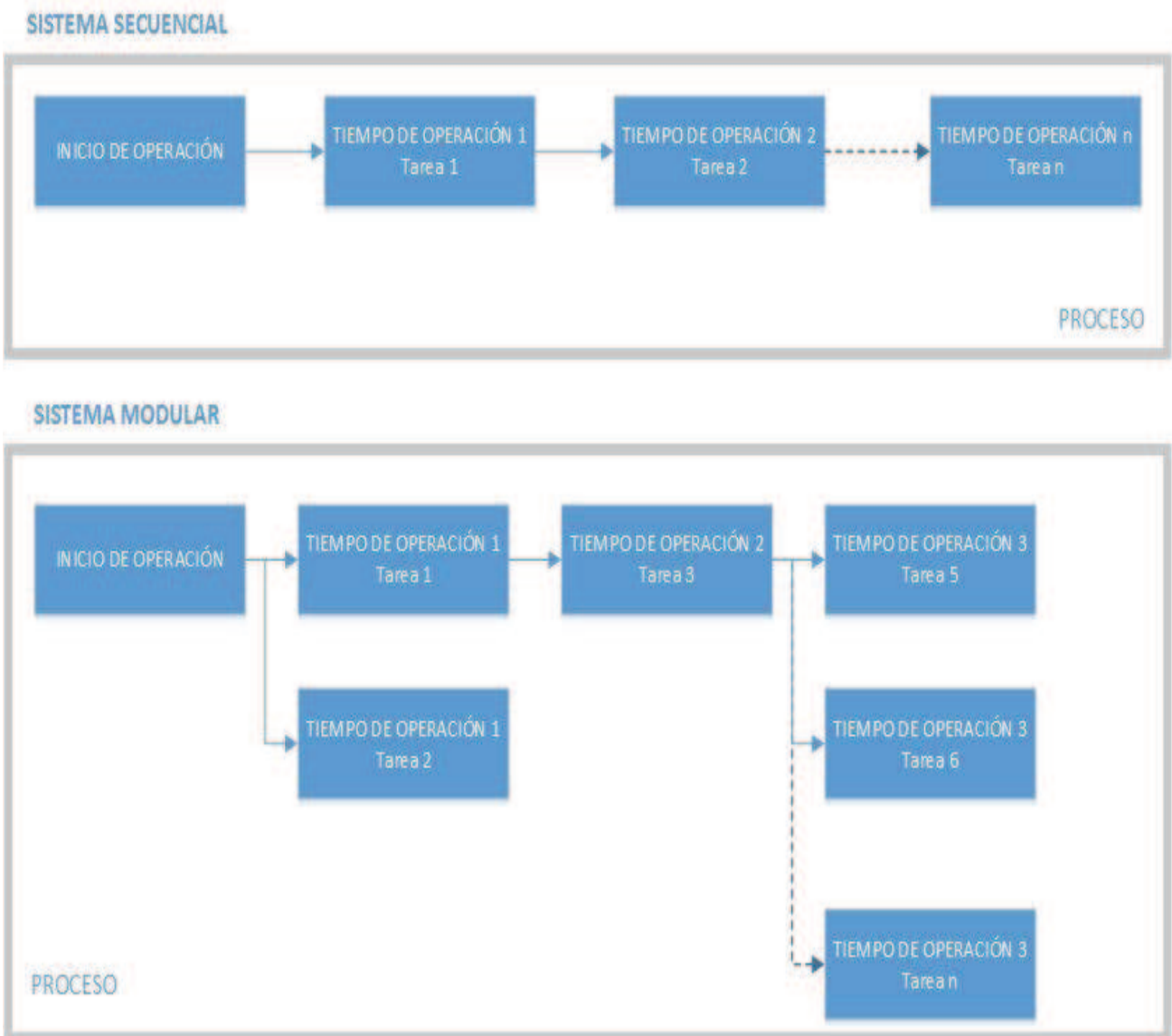


Figura 26. Ventaja de unidad de control modular

En cuanto al diseño de los módulos, para las tareas del sistema mecánico, la base será el análisis del proceso y la jerarquía de sus componentes donde debe considerarse la multi tarea, la interacción con el usuario y la flexibilidad de operación del proceso; mientras que para el diseño del módulo de interfaz se usará la misma base, pero considerando que no cuenta con señales de actuación. Estas consideraciones integran a los siguientes conceptos:

- Estructura deliberativa: sensado, planeación y actuación.
- Estructura Bitbus de comunicación generalizada.
- Estados de Error/Tarea auxiliares en comunicación.
- Planeación orientada a eventos.
- Gestión de comunicación y de proceso bajo un módulo maestro.
- Jerarquía basada en componentes.

Finalmente, la integración de estos conceptos visto desde el proceso, permite una ventaja ante un sistema secuencial al dividir a la unidad de control en módulos; la multitarea se ve reflejada por la jerarquía de operación sobre cada módulo reduciendo los tiempos de operación sin importar cuanto tiempo le tome ejecutar una tarea a cada módulo; aunque en cuanto costos representa un incremento por cada módulo, Figura 28.

CAP. 4 DISEÑO DE MÓDULOS

El diseño de la unidad de control modular para automatización de procesos se basa en el análisis del proceso visto como dos partes: la jerarquía de funciones de operación basado en componentes (por el sistema físico) y por las funciones de operación de planeación y configuración de una unidad de control (por el tipo de autómatas). Los módulos representan una estructura de jerarquías tanto para sistemas físicos como de operaciones de un proceso y el diseño se desarrolla a partir de una estructura genérica.

Se sigue la planeación de etapas del proyecto por Fases de trabajo, Tabla 7. Aunque en la primera etapa del proyecto se planteó un sistema secuencial, se determina que para las siguientes etapas y Fases de trabajo se debe implementar un sistema modular para cumplir con las especificaciones del proyecto. Además, es importante mencionar que el escrito del presente trabajo se inicia al término de las Fases planteadas por lo que se omite la estructura secuencial y sólo se hace su mención para fines de comparación, establecimiento de condiciones y justificación del sistema modular.

Cada módulo es definido en su Fase de trabajo a partir del análisis de las funciones de operación del proceso y de las operaciones de su sistema, esta distinción se debe a que la herramienta Grafcet está definida en un grupo de autómatas programables específicos, PLC, por lo que los diagramas quedan acotados para los grupos de operación del proceso. El seguimiento del diseño descompone a la unidad de control en tres tipos de operaciones:

- Operaciones de comunicación: encargadas de atender la solicitud de operaciones.
- Operaciones de módulo: encargadas de manejar las solicitudes y administrar el módulo.
- Operaciones de proceso: funciones del proceso a implementar en la unidad de control.

De estas operaciones, para la implementación de las operaciones del proceso en un sistema basado en microcontrolador se siguen dos tipos de metodologías:

- Metodología de sistemas de control basada en componentes: la que determina las funciones de operación de un proceso a automatizar, considerando jerarquía de componentes por condiciones físicas y lógicas del sistema.
- Metodología de diseño basado en el producto: para la implementación de las funciones de operación de proceso a módulo.

La herramienta Grafcet auxilia la metodología de diseño por componentes al establecer las la evolución de las funciones de operación de forma gráfica, manteniendo la relación física-lógica del proceso y la unidad de control.

Se vuelve extenso por las definiciones de condiciones y el establecimiento de la relación de la parte física con la parte lógica del proceso en un mismo gráfico de operaciones, pero por otra parte la herramienta establece la relación física y lógica de manera simultánea, lo que permite identificar estados físicos y lógicos, condiciones y tiempos de operación, jerarquía de componentes y funciones, flujo del proceso y datos, y le da sentido modular por la jerarquía de operaciones. Por esta razón se usa la metodología de Grafcet con jerarquía de componentes enfocada a proceso para complementar la metodología de Ulrich enfocada a producto.

Antes de establecer los módulos, su estructura de comunicación y bloques de funciones, el desarrollo del diseño basado en Grafcet inicia con un análisis completo del proceso dividido en tres partes:

1. Definiciones del proceso y el sistema automatizado
 - Mecánica del proceso: establecimiento de sistemas y componentes electromecánicos para la automatización del proceso, diseño mecánico.
 - Arquitectura del proceso: estudio de la estructura física de solución de automatización.
 - Operaciones del proceso: estudio del proceso a automatizar.
2. Establecimiento de las funciones de operación del sistema
 - Jerarquía de componentes: condiciones y estados del sistema a automatizar.
 - Estructura de operaciones: tiempos de operaciones a partir de estados y condiciones del sistema a automatizar.
3. Establecimiento del algoritmo de las operaciones del proceso
 - Cartas de función: funciones de operación ejecutiva del proceso del sistema a automatizar.
 - Grafcet del proceso: funciones de procesos de operación en tiempos de operación.

Para la primera parte, la mecánica del proceso no es resuelta para este trabajo debido a que su desarrollo está a cargo del de otro grupo de trabajo del proyecto, como se mencionó en cap. 3.1.1 y solo representa condiciones y especificaciones para el análisis del proceso, de tal forma que la arquitectura y operaciones se basan en la mecánica y se desarrolla su análisis de componentes para asociarlo a los procesos de operación.

Finalmente, es importante aclarar que para el desarrollo de los módulos se reserva la descripción de la programación en el microcontrolador, por la cantidad excesiva de líneas de código, por representar una solución particular y por las condiciones de privacidad del proyecto, de esta forma el desarrollo queda estructurado de forma general a partir de la unión de métodos de diseño enfocados en el proceso, sus componentes y la estructura general del sistema de control modular. Las pruebas fueron reducidas para presentar las características principales desarrolladas para la validación del modelo funcional de la unidad de control modular. Estas condiciones son mencionadas en la realización de las pruebas para no desviar el enfoque del diseño general.

4.1 Jerarquía de operaciones de proceso

Para las primeras etapas del proyecto las consideraciones y el planteamiento realizado, Apéndice A y B, describen la distribución del sistema mecánico como un modelo funcional en relación al proceso de mezclado en la prueba de laboratorio, Figura 27. La distribución mecánica debe ser analizada antes de plantear los procesos de la unidad de control debido a que representa una restricción en cuanto al funcionamiento. Para este análisis se va a relacionar el proceso con el sistema mecánico para agrupar las funciones de cada sistema.



Figura 27. Ciclo de operación en laboratorio y en sistema mecánico

Esta disposición en tres etapas permitió establecer un sistema mecánico con una distribución circular que permite adaptar los subsistemas mecánicos de forma flexible, pero para la secuencia del proceso representa una limitación de las operaciones en tiempos de operación.

Los módulos deben de distribuirse conforme a los procesos de laboratorio y adaptarse a las restricciones que presenta el sistema mecánico para poder maximizar el número de tareas por tiempo de operación. Esta integración debe permitir la flexibilidad de adaptación de módulos para sistemas posteriores sin afectar las multitareas.

4.1.1 Arquitectura del proceso

El establecimiento de la arquitectura modular de un proceso, es aquella en la que cada elemento del **modelo funcional**⁸ esta implementado por un conjunto de componentes que desarrollan operaciones del proceso y en el que hay pocas interacciones bien definidas entre estos módulos. Esta clasificación delimita el funcionamiento y las operaciones dentro de un proceso, además esta arquitectura permite que un cambio de diseño que se haga a un componente no modifique, sin recurrir a cambios, al funcionamiento de otros componentes.

El proceso a automatizar para el presente trabajo consiste en un híbrido entre una parte del proceso de mezclado de la prueba de jarras en laboratorio con una parte del proceso de mezclado y filtrado en planta, Apéndice A, éste híbrido permite tener mejores resultados en selección del floculante al simular, en menor escala, la interacción de más de un polímero candidato en el proceso primario de tratamiento de efluentes, con lo que el análisis de las muestras resultantes incluirá el comportamiento del mezclado con su separación en planta y no en el de laboratorio.

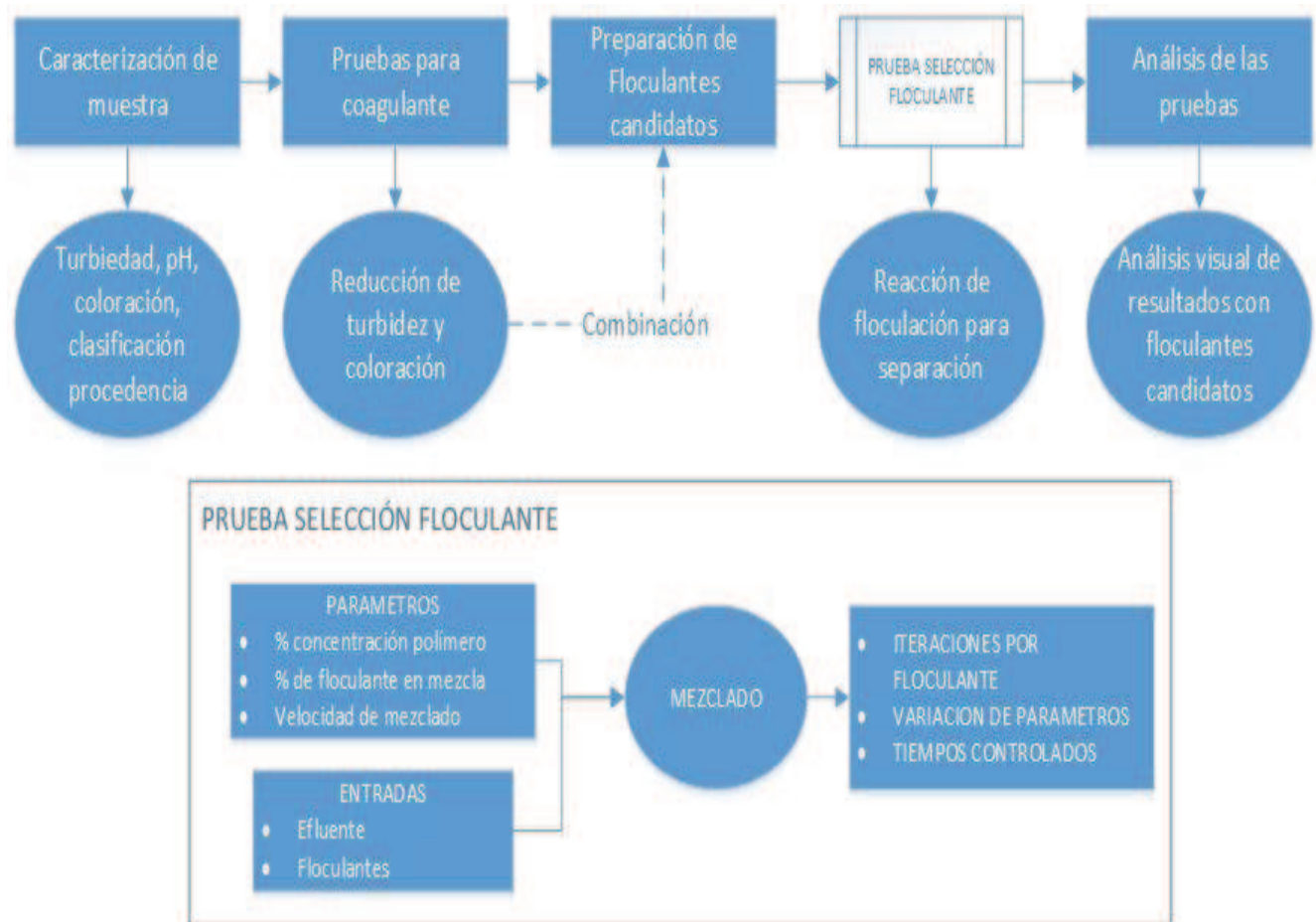


Figura 28. Prueba de Jarras

⁸ *Modelo funcional*: es aquel que integra las operaciones y transformaciones individuales que contribuyen al rendimiento general del modelo; estos elementos se describen en forma esquemática antes de ser reducidos a tecnologías específicas.

De forma general la prueba de jarras consiste en mezclar floculantes candidatos con un mismo efluente de forma iterativa, donde la importancia está en el análisis de los efluentes decantados y en el control de los parámetros de las pruebas, Figura 28. Por otro lado, una prioridad del proyecto es la adaptación de las pruebas con floculantes al proceso de mezclado en planta para obtener resultados más precisos. Por tanto, es necesario mencionar el proceso de mezclado en planta para entender su adaptación con el proceso de laboratorio.

Se muestra en la Figura 29 parte del proceso primario para el tratamiento de residuos industriales, donde el proceso de interés es el uso de floculante como agente preparador y separador. Del diagrama se observa que existen dos procesos de mezclado, uno para preparar a las aguas residuales y otro que permite la floculación. El floculante que se usa es resultado de la prueba de jarras, siendo el proceso de mezclado la etapa más importante por lo que agregar un proceso de simulación del mezclado en planta mejora los resultados.

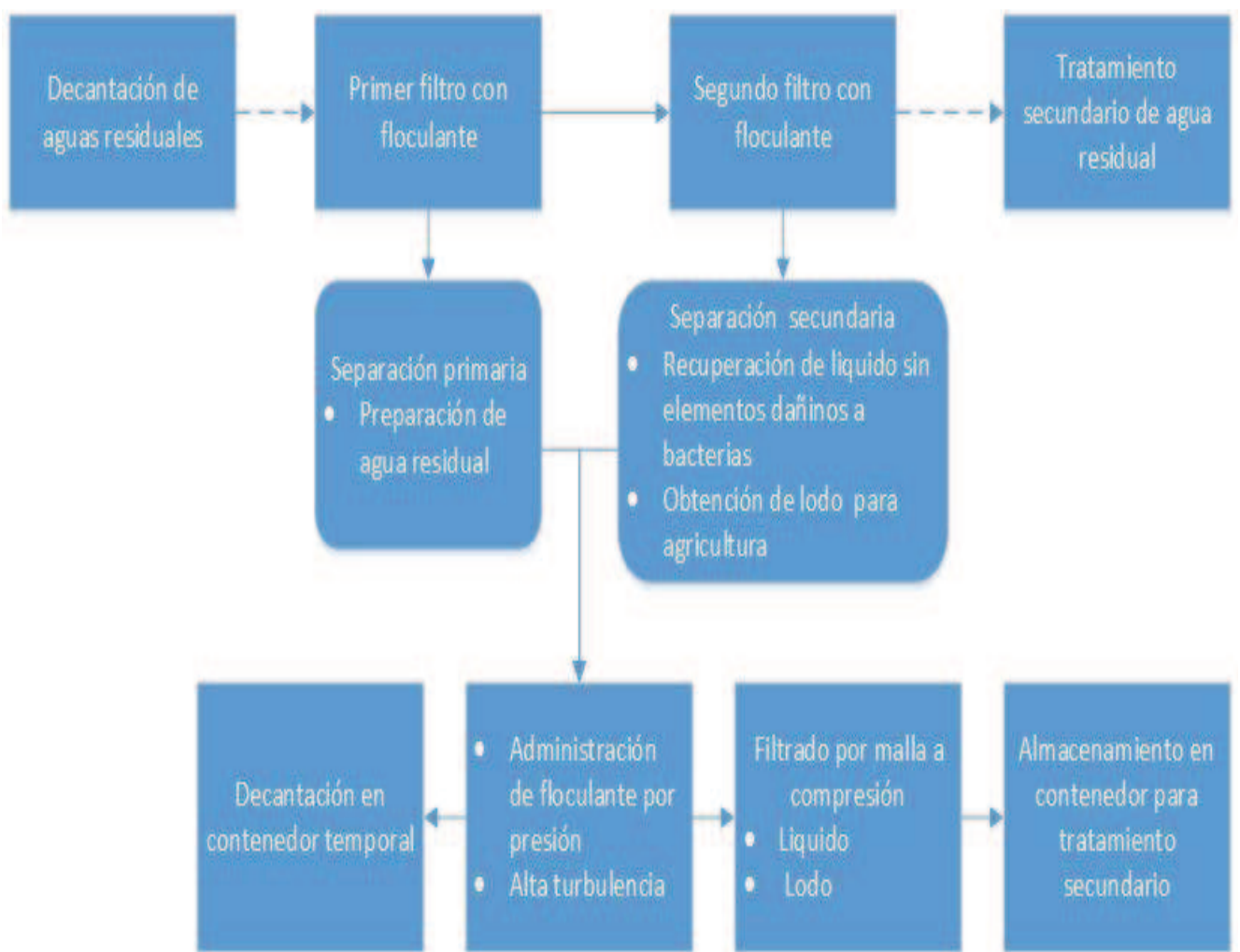


Figura 29. Floculante en proceso primario de tratamiento de efluentes

Aunque el proceso de mezclado es sencillo tanto en planta como en laboratorio, la importancia radica en el análisis de la mezcla en función de sus parámetros, de ahí que se puede anticipar la necesidad de un sistema de control modular para todo el proceso, que permita tener el control de todas las variables, que reduzca los tiempos de iteración y que dé al usuario datos de cada iteración para almacenar y comparar resultados.

Definidos los procesos de forma independiente, Figura 28 y 29, se establece la unión de la parte de preparación y de mezclado en prueba de jarras con el análisis y el mezclado en planta, Figura 30; este diagrama representa, de forma general, el desarrollo completo del proceso en proyecto donde el desarrollo de este trabajo pertenece al “mezclado de floculante-efluente”.

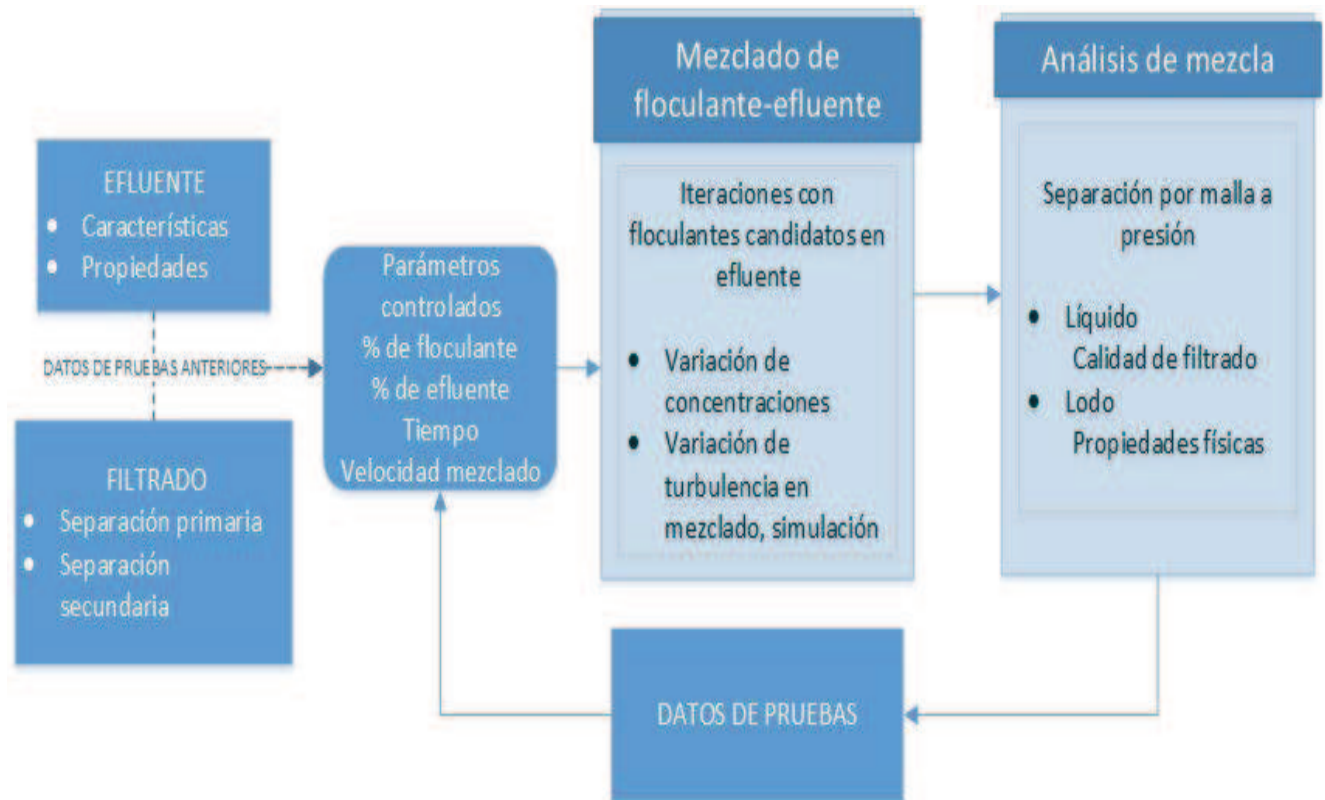


Figura 30. Proceso completo del proyecto LODOS

Para el desarrollo del análisis de operación se definirá primero su arquitectura. Se debe definir los elementos y los pasos que componen al proceso de mezclado en laboratorio para posteriormente asociarlos al sistema mecánico del proyecto. La estructura para el proceso se basa en cuatro grupos: de herramientas, preparación, mezclado y recolección. De forma independiente se define al grupo de entradas y salidas debido a que no representa acciones en el proceso.

Entradas y salidas:

- **Efluente:** Muestra de líquido residual.
- **Floculante preparado:** Mezcla preparada de polímero con coagulante, previo a la prueba.
- **Líquido recuperado:** Tras pasar la mezcla por el embudo con filtro.

Herramientas para el proceso de mezclado

- **Contenedor primario:** Recipiente donde se lleva a cabo la mezcla.
- **Contenedor secundario:** Recipiente donde se almacena el líquido recuperado.
- **Dosificador floculante:** Elemento para administrar la cantidad deseada en el mezclado.
- **Embudo con filtro:** Recipiente para separar el sólido “lodo” del líquido recuperado.

Preparación del proceso de mezclado

- **Selección de floculantes candidatos:** Con base en el tipo de muestra de efluente.
- **Selección de volumen del floculante:** Con base en histórico de pruebas de cada floculante.
- **Selección de volumen de efluente:** De acuerdo con los floculantes candidatos, se selecciona un volumen fijo de efluente.
- **Selección de velocidad y tiempo de mezclado:** Para todas las pruebas se determina la velocidad y tiempo de mezclado para disolver el floculante sin romperlo. Se busca simular el mezclado de la planta.
- **Selección de número de pruebas:** Para poder obtener un resultado concreto se realiza más de una prueba por floculante para descartar falsos positivos y negativos.

Mezclado de efluente con floculante

- **Dosificación efluente:** Suministro en contenedor primario.
- **Dosificación floculante:** Suministro en contenedor primario con efluente antes de mezclar.
- **Mezclado:** Con la velocidad y tiempo determinado se mezcla el efluente con el floculante, debe considerarse que algunos polímeros pueden romperse por la disposición del mezclador.

Recolección de muestra

- **Separación solido-líquido:** Terminado de revolver, la mezcla se dispone para vaciarle en el embudo con filtro donde el “lodo” se formará y el líquido recuperado será puesto en el contenedor secundario.
- **Lavado de contenedor:** Para nueva iteración debe limpiarse el contenedor primario, mientras que el contenedor secundario se limpiará luego de analizar el líquido recuperado.

Por otra parte, antes de establecer las definiciones de la arquitectura, para el sistema mecánico el planteamiento puede verse en los Apéndices A y B, por lo que sólo se desarrolla un resumen de la distribución de los sistemas para el desarrollo de este trabajo.

Como se ha mencionado anteriormente, el proyecto tiene desarrollado parte de la primera etapa en cuanto a la mecánica, Figura 31, esta hace referencia a los sistemas de posicionamiento de los contenedores y no es una versión final. La estructura mecánica está dispuesta para integrar a los demás sistemas de forma circular, se encuentra compuesta por las siguientes partes:

- Giro de contenedor,** después del lavado debe orientarse para la dosificación de lodo.
- Giro de contenedor,** para que la mezcla pase al sistema de recolección de muestra.
- Lavado de contenedor,** en esta posición estará el sistema de lavado.
- Posicionamiento batidora,** para el sistema de mezclado y dosificación polímero.
- Posicionamiento carrusel,** para las tres Fases del proceso: llenado, mezclado y lavado.

Los elementos que componen de forma general al sistema mecánico se clasifican en tres grupos: herramientas, estructura general y estructuras específicas. Estas partes surgieron por la adaptación de las acciones del operario en el proceso de laboratorio hacia la mecanización, Apéndice A, y mantienen las mismas entradas y salidas del proceso. Las definiciones para cada grupo son:

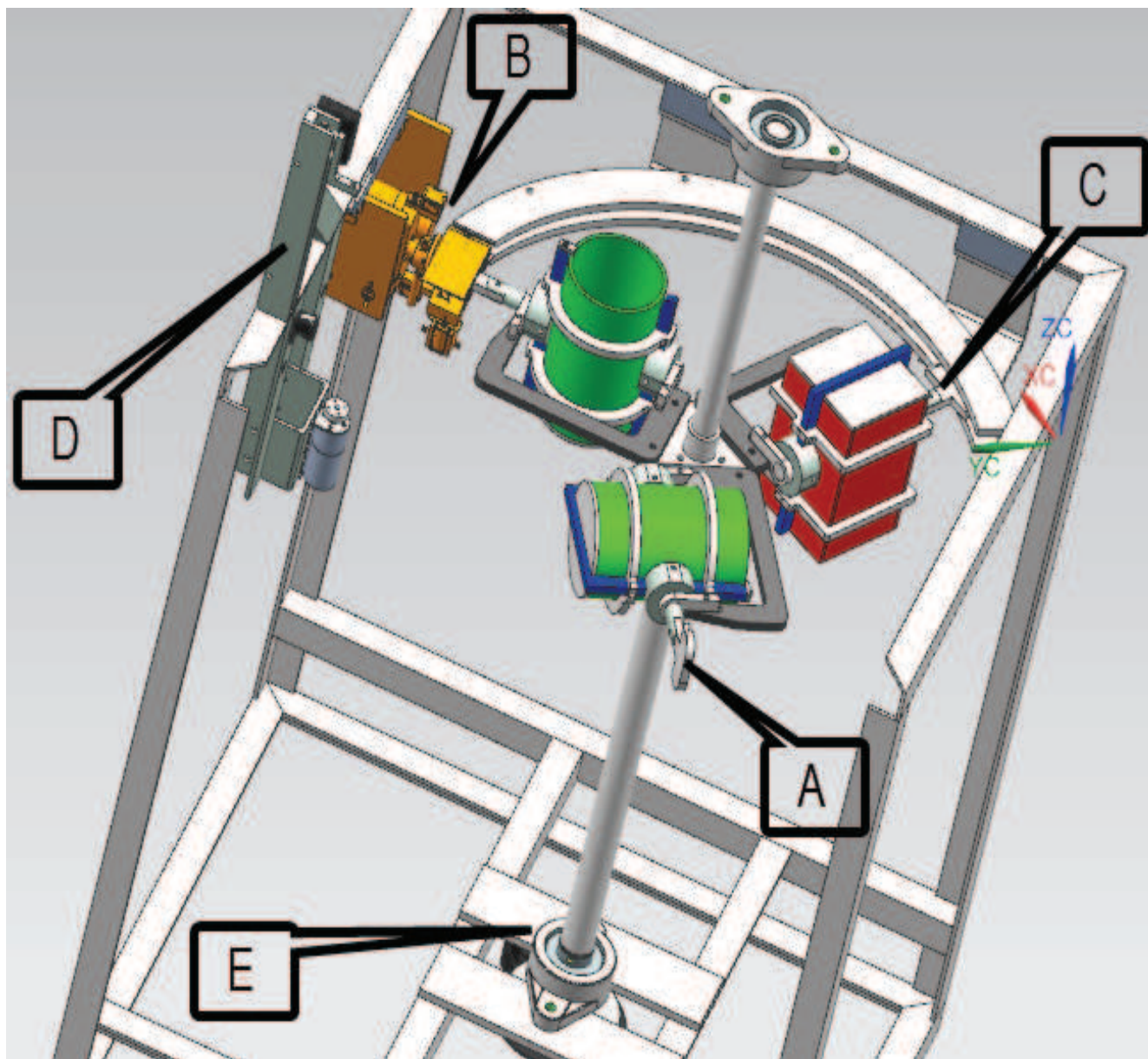


Figura 31. Estructura mecánica, primera etapa, Apéndice A

Herramientas de sistema mecánico

- **Contenedor:** Recipiente general para la mezcla.
- **Bomba:** Actuador para dosificar efluente o floculantes o agua.
- **Receptor de muestra⁹:** Recipiente preparado para filtrar el sólido “lodo” y almacenar el líquido recuperado.

Estructura general

- **Posicionamiento base:** Estructura base para llevar contenedores a diferentes posiciones.
- **Posicionamiento auxiliar:** Sistema mecánico auxiliar para mover una herramienta.
- **Orientación:** Sistema mecánico auxiliar en una posición para orientar contenedores.

⁹ Para la etapa de análisis de muestra, no implementado para este trabajo.

Estructuras específicas

- **Dosificación:** Sistema compuesto por al menos un motor y un sensor para administrar alguna sustancia.
- **Batidora:** sistema compuesto de un motor y un sensor que regula la velocidad de mezclado.

Las definiciones de los grupos de elementos que componen a la estructura mecánica y la de grupos del proceso permiten establecer la relación entre el sistema mecánico y el proceso de mezclado en la distribución de los sistemas en función de las entradas, las cuales determinan el funcionamiento de cada sistema, Tabla 14.

Tabla 14. Relación de las entradas entre el proceso y el sistema mecánico		
Sistemas mecánicos	Entradas	Procesos
Dosificación lodo	Cantidad de lodo	Dosificación efluente
Dosificación polímero	Cantidad de floculante	Dosificación floculante
Mezclado	Revoluciones de mezclado Tiempo de mezclado	Mezclado con batidora
Carrusel	Movimiento por iteración	Posicionamiento en tres pasos de contenedores
Orientación contenedor	Contenedor en posición	Orientación de contenedor y recolección de mezcla
Posicionamiento batidora	Posición de batidora	Movimiento de batidora en contenedor

De las Figuras 30 y 31 se muestra cómo se distribuye en forma circular a los sistemas de la estructura mecánica y su relación al ciclo de operación, esta distribución en tres posiciones junto con la relación de entradas permite definir la relación de las posiciones con el proceso. En la Tabla 15 se puede apreciar que la mayor carga de trabajo se encuentra en la posición 1.

Tabla 15. Relación de las posiciones en estructura mecánica con el proceso	
Posición	Proceso
Posición 1	Recolección muestra
	Posicionamiento de batidora
	Mezclado
	Dosificación polímero
	Orientación contenedor
Posición 2	Lavado
Posición 3	Orientación de contenedor
	Dosificación de lodo

A partir de la Tabla 15, se puede representar un diagrama que relaciona directamente las posiciones del sistema mecánico con el proceso, Figura 32. En este diagrama se muestran los sistemas, las posiciones y el ciclo del proceso de forma simplificada, sin detallar los componentes de cada sistema, pero a partir de la Tabla 14 se puede definir un esquemático por planos de cada posición, Figura 33, para incluir de forma gráfica los sistemas distribuidos por posición.

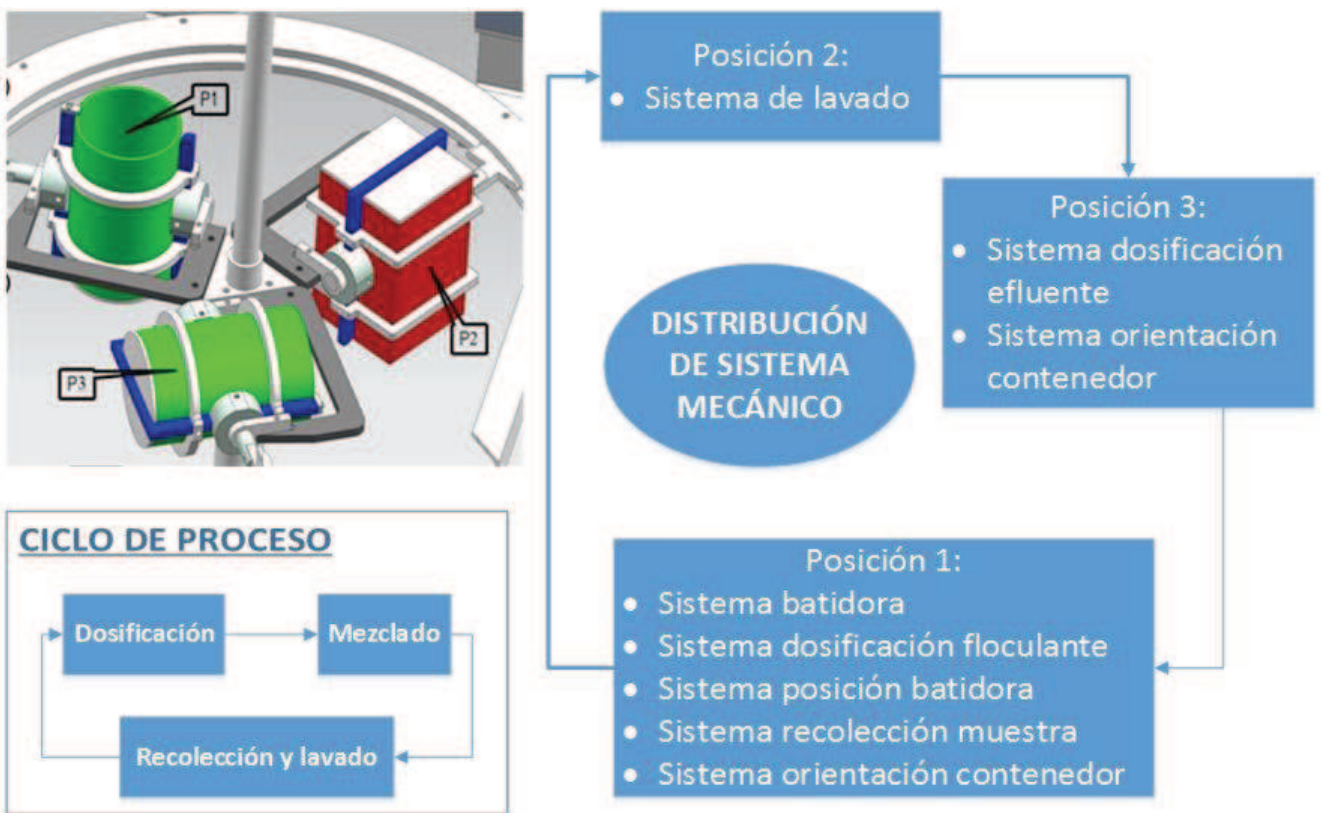


Figura 32. Distribución de posiciones y del ciclo de operación

Finalmente, la arquitectura del proceso de mezclado con base en el sistema mecánico queda definida en la Tabla 16, donde la distribución de los procesos y los grupos de elementos que conforman a la estructura quedan implícitos dentro de los sistemas y sus posiciones, Figura 33.

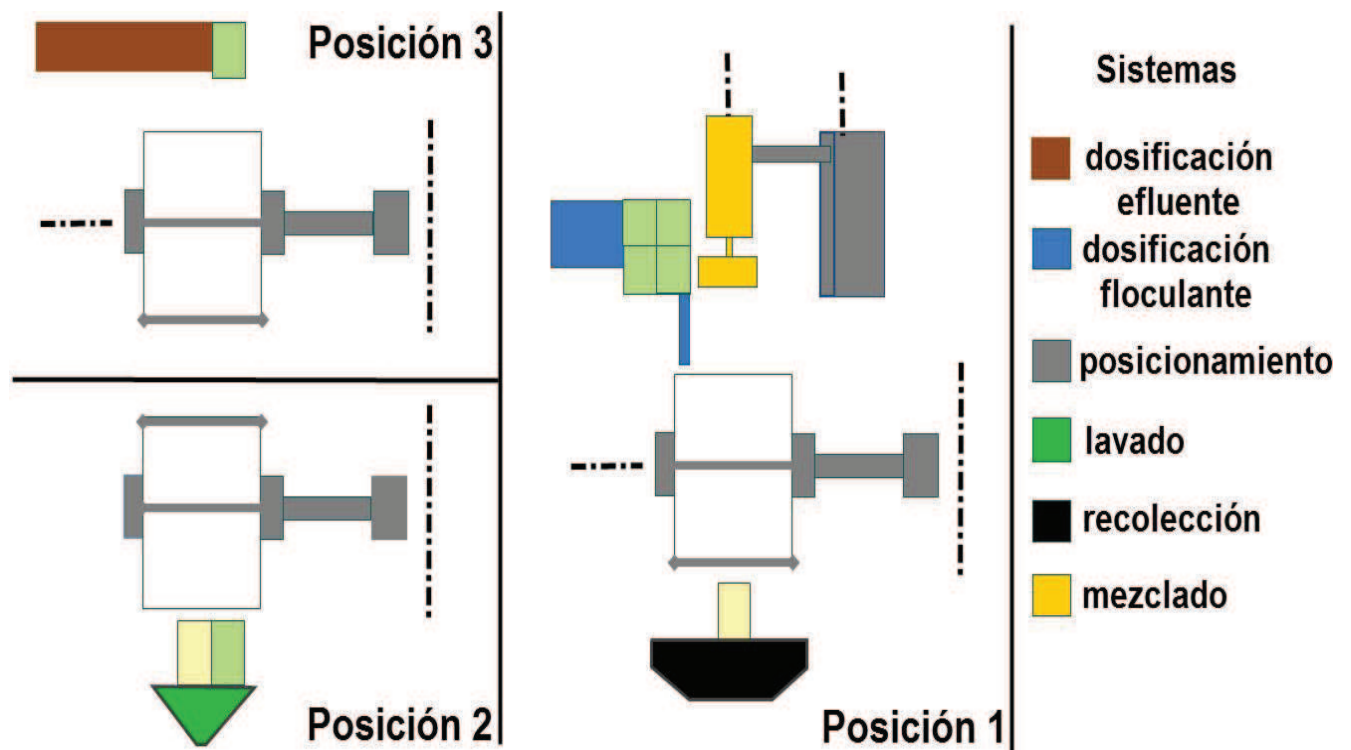


Figura 33. Distribución de sistemas por posición

Tabla 16. Arquitectura del proceso

Grupos del proceso	Referencia de Laboratorio	Relación con sistema mecánico	Posición en estructura	Etapas proyecto
Herramientas	Contenedor primario	Contenedores	En posición 1, 2 y 3 Sistema carrusel	1
	Dosificador floculante	Bomba de efluente	En posición 1 Sistema dosificación efluente	4
	Contenedor secundario	Receptor de muestra	En posición 2 Sistema de análisis	5
	Embudo con filtro			
Preparación	Selección de parámetros de prueba	Configuración de los parámetros de cada sistema	En posición 1, 2 y 3 Todos los sistemas	1-5
Mezclado	Dosificación efluente	Dosificación Posicionamiento base	En posición 3 Sistema dosificación efluente	4
	Dosificación floculante	Dosificación Posicionamiento base	En posición 1 Sistema dosificación floculante	4
	Mezclado	Batidora Posicionamiento auxiliar	En posición 1 Sistema carrusel y posición batidora	2, 3
Recolección De muestra	Separación sólida/líquido	Orientación	En posición 1 Sistema orientación contenedor y análisis	1
	Lavado contenedor	Posicionamiento base	En posición 2 Sistema carrusel	1

Para el desarrollo de los módulos de la unidad de control el planteamiento y sus funciones dependerán de la distribución de los componentes, arquitectura, para establecer tiempos de operación, para aprovechar la distribución mecánica y para establecer las condiciones de operación del proceso.

4.1.2 Operaciones del proceso

Para establecer el orden de jerarquía del proceso basado en componentes se va a desglosar el proceso de operación de mezclado mediante definición de grupos de operación, tomando en cuenta los sistemas de la estructura mecánica y su arquitectura. Para ello se debe considerar todas las posibles tareas, los estados y las etapas en operación, tanto para una ejecución como en espera, éstos se representan en tres grupos de operación con base en su funcionalidad: parada y puesta en marcha, proceso de funcionamiento y proceso de falla.

En la Figura 34 se muestra la estructura general de operaciones de un proceso. Cada uno de estos grupos debe ser tratado independientemente para definir estados, tareas y condiciones de su operación, pero deben analizarse en grupo para establecer límites y condiciones del proceso.



Figura 34. Grupos de operación de un proceso

El desglose del proceso se centra en conocer sus funciones de operación para asociarlos con componentes de los sistemas y para establecer las funciones que desarrollará la unidad de control; además representa condiciones para desarrollar los módulos con base en la estructura jerárquica del proceso. Desde el punto de vista del proceso y la estructura mecánica, Tabla 16, los grupos de operación de la etapa de mezclado de floculante-efluente son los siguientes:

Parada y puesta en marcha

El proceso de mezclado tiene un estado de no habilitación donde se realizan preparativos como mantenimiento a las herramientas o al espacio de trabajo, así como verificar el funcionamiento, Apéndice B. También se plantean los parámetros de operación para las pruebas. Los puntos de parada y preparación son los siguientes:

- **Parada en estado inicial:** Todas herramientas deben estar orientadas y en posición de inicio sin estar habilitadas. Contenedores y dosificadores deben estar limpios.
- **Parada solicitada al final del ciclo:** Se deshabilitan los dosificadores y los contenedores son puestos a limpiar.
- **Preparación para la puesta en marcha después de un defecto:** Se deshabilitan las herramientas y se verifica el estado de posición y orientación al momento del defecto para reposicionar al estado inicial, si es necesario se solicita ayuda del operador.
- **Puesta del sistema en estado inicial:** las herramientas deben estar limpias; las posiciones y orientaciones no deben obstruirse y deben quedar dispuestos para arranque inmediato de nueva iteración sin tener ninguna herramienta habilitada.

Procedimientos de funcionamiento

Los procesos de funcionamiento están centrados en un ciclo iterativo de posicionamiento de la herramienta que siguen un orden secuencial llamados marchas. Los procedimientos de funcionamiento son los siguientes:

- **Producción normal:** La secuencia de producción inicia con la dosificación de efluente en un contenedor; posteriormente se dosifica el floculante a la vez que se bate para uniformizar la muestra, luego se mezcla con los parámetros deseados para simular la turbulencia de la planta; finalmente se vacía la mezcla para su recolección y análisis.

- **Marcha de preparación:** Se verifica que todas las herramientas estén en posición y orientación correspondiente, en caso contrario se verifica con orden el movimiento de las herramientas. Se realiza un movimiento general de las posiciones.
- **Marcha de cierre:** Las posiciones y orientaciones deben estar como en el estado inicial. Se procede a limpiar los contenedores y finalmente hacer una marcha de preparación para verificar funcionamiento.
- **Marcha de verificación sin orden:** El operador prueba cada uno de los sistemas de forma manual para verificar el funcionamiento y los estados.
- **Marcha de test:** Se pone en marcha un ciclo de interacción simple para probar el funcionamiento de todos los subprocesos. Las herramientas quedan de vuelta en su posición original y no afecta a los estados internos.

Procedimientos de falla

Las causas de fallo en el proceso pueden darse por fallas físicas, de alguna herramienta, durante su posicionamiento u orientación. Estas fallas deben de detener toda la prueba y reiniciarla para evitar tomar muestras contaminadas. Los procedimientos considerados en caso de falla son los siguientes:

- **Parada de emergencia:** Se deshabilitan todas las herramientas, los estados se guardan y se notifica al operador el tipo de fallo. Se pone en marcha la verificación para poner el sistema en estado inicial.
- **Diagnóstico y/o tratamiento de defectos:** Durante el proceso se guardan los estados de cada subproceso para tener referencia del momento de fallo. Esto debe auxiliar al operador para identificar el fallo de forma visual y en caso necesario ayudar al sistema a elegir el tipo de solución.

4.1.3 Jerarquía de componentes

Para describir al proceso en operaciones por niveles jerárquicos, con base en sus componentes y en las condiciones físicas del sistema mecánico, se va a utilizar un modelo genérico de organización modular para la funcionalidad del sistema mecánico **SM**, lo que permitirá posteriormente definir de forma modular a la unidad de control **UC**. Se trata de reducir todas las posibles operaciones de los sistemas en grupos funcionales con base en la jerarquía de funciones de componentes básicos funcionales **FCB**.

Al relacionar la arquitectura y el proceso de operación en función de los FBC se establece la relación de condiciones de estados de operación, que se representan por niveles jerárquicos de funciones físicas esenciales del sistema, que será la base para el análisis de la modularidad del proceso y puente para pasar de funciones asociadas al SM a funciones asociadas a la UC, es decir se identifican los bloques funcionales de todo el proceso junto con sus condiciones de operación.

En la Tabla 16 y en las Figuras 32 y 33 se resume los sistemas físicos que componen el proyecto para la automatización del proceso, Figura 32; son ocho sistemas distribuidos de forma circular, de los cuales para el presente trabajo se desarrollan seis. Se establece un esquema general de los componentes funcionales de cada sistema a partir del modelo desarrollado en el Apéndice B, en donde se describió los componentes de la estructura mecánica, Figura 35.

De la distribución de las Fases de trabajo de la Tabla 7, se definen los sistemas en tres grupos de acuerdo con la planeación del proyecto, se agruparon por el funcionamiento y el orden de las etapas

del proyecto. A partir de la Figura 33 y 35 se desarrollan los esquemas de cada sistema en orden de Fases para poder identificar los componentes y las funciones en forma jerárquica, y su estructura de operación será analizada en la modularidad del proceso.

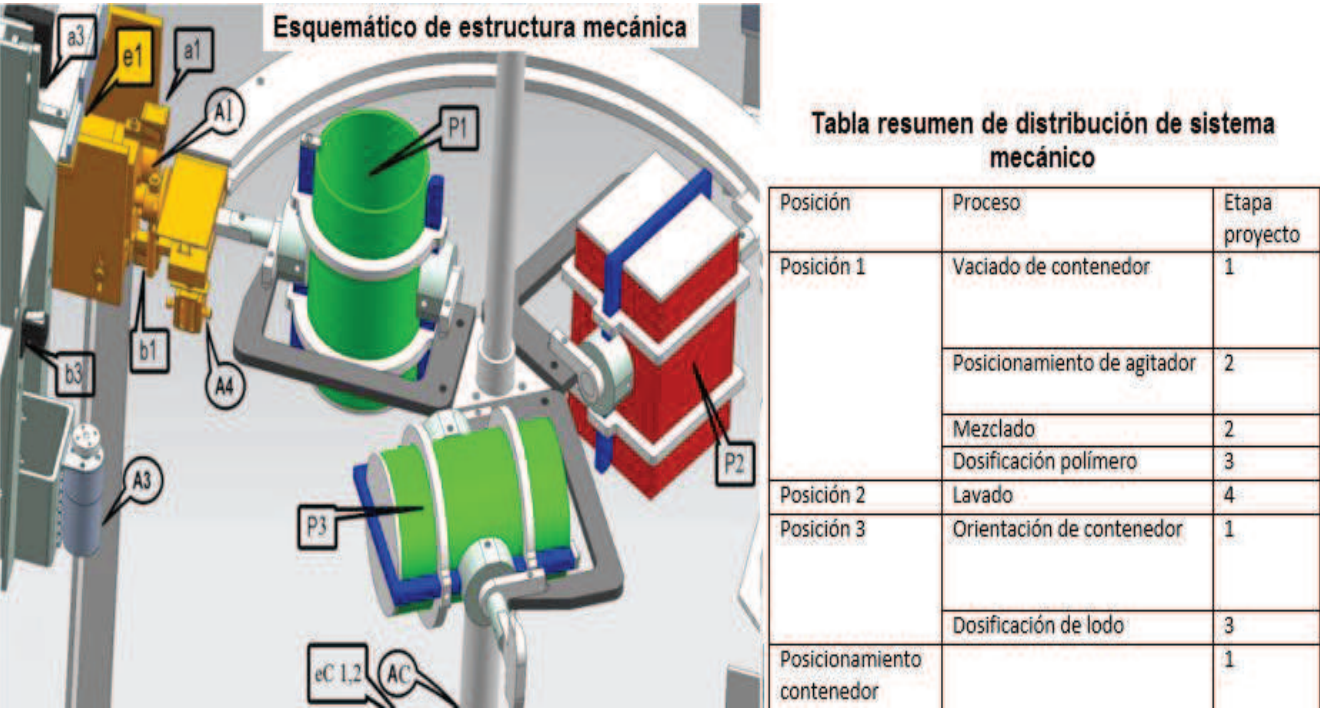


Figura 35. Esquema general de componentes y su distribución, Apéndice B

Fase 1, control estructura

La Fase uno integra tres sistemas que tienen en común el cambio de estado de movimiento sea de forma circular, lineal o de orientación; se compone de mecanismos, de motores y de sensores de posición. En las Figuras 36, 37 y 38 se identifican en un esquema plano la distribución de cada sistema y la de sus componentes funcionales.

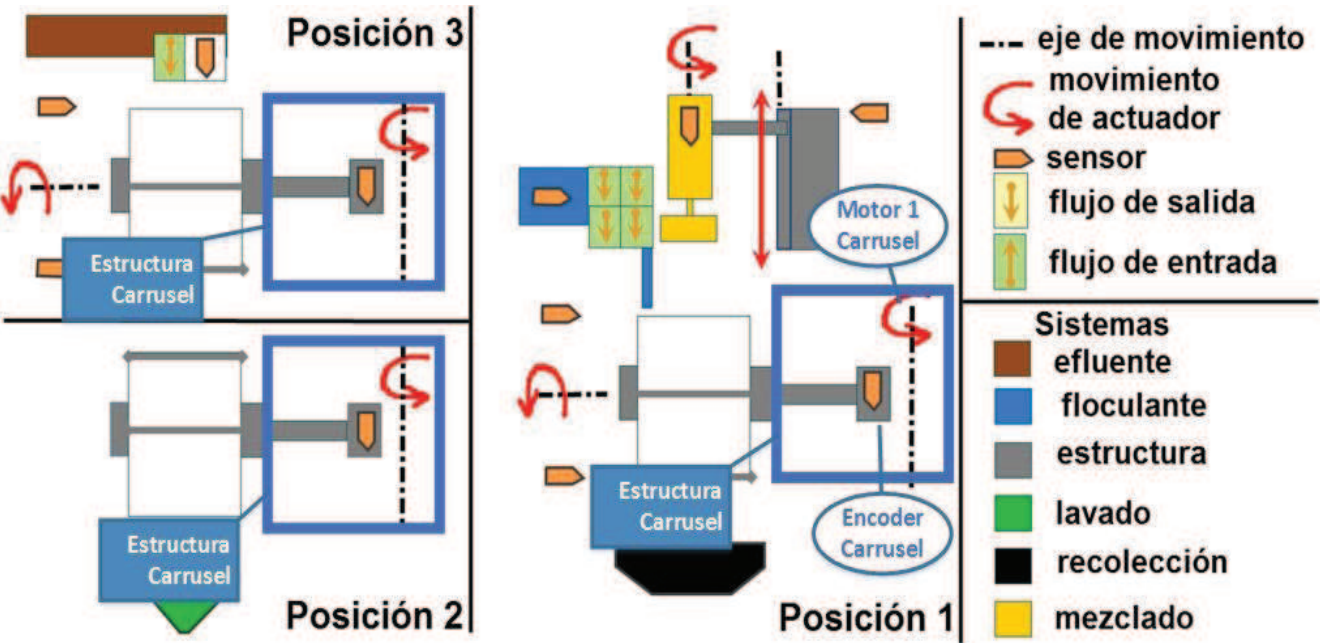


Figura 36. Esquema sistema carrusel

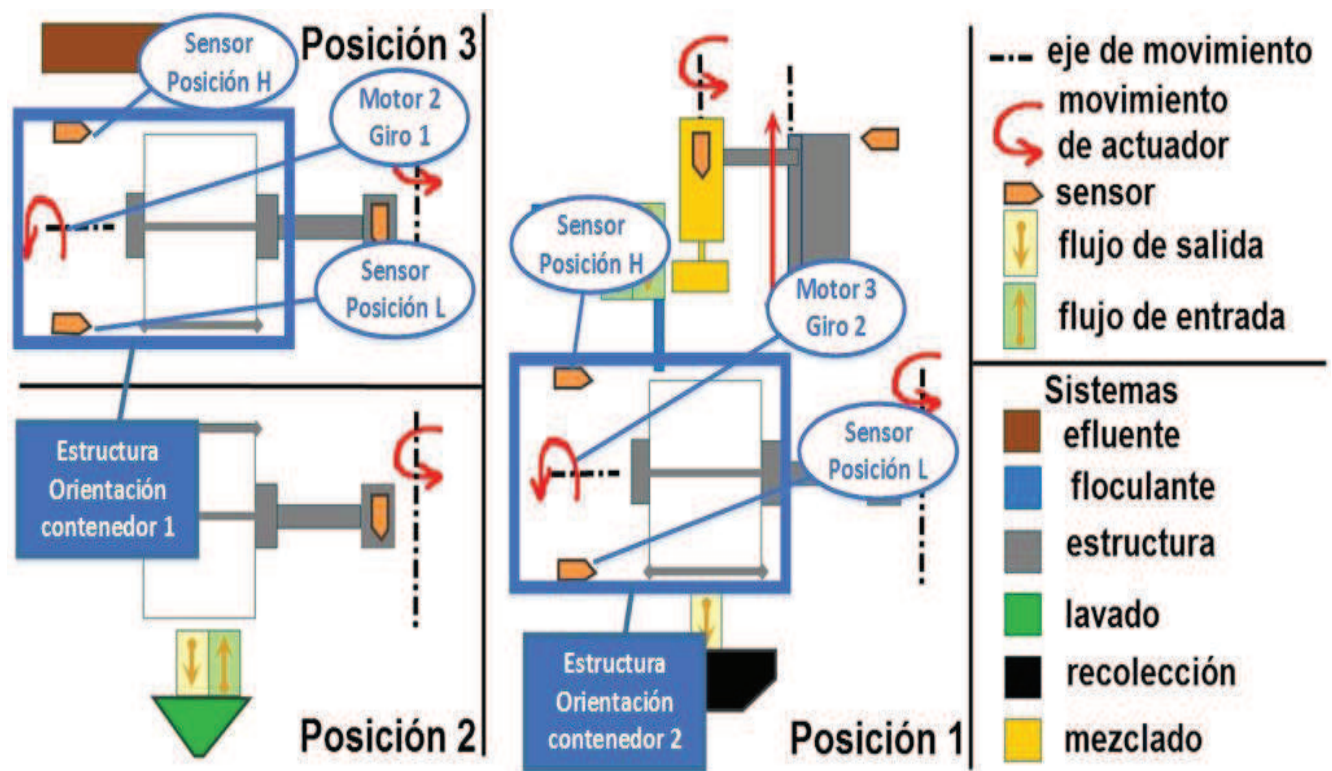


Figura 37. Esquema sistema orientación de contenedores

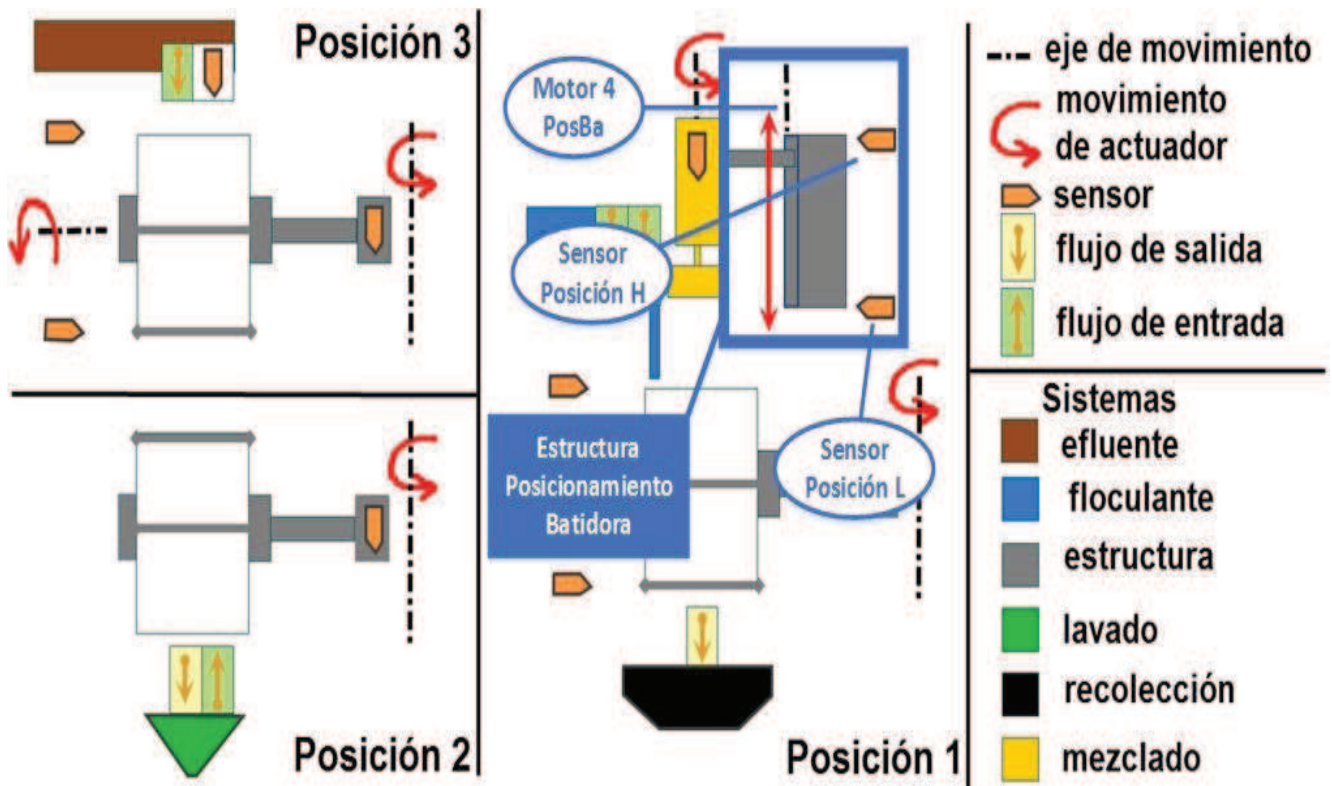


Figura 38. Esquema sistema posicionamiento batidora

Los esquemas muestran los FBC para el funcionamiento de la estructura; los componentes tienen en común el movimiento de sus actuadores, sea para posicionar u orientar, esto permite asociarlos por sus entradas y salidas a un mismo tipo, pues el elemento que relaciona estos sistemas es su

mecanismo que cambia la actuación a un estado de posicionamiento. Para el sistema de orientación, se divide en dos FBC por las condiciones de operación, Apéndice B, al no efectuarse el cambio de orientación con las mismas condiciones de operación. Los FCB se describen en la Tabla 17.

Tabla 17. FCB de la Fase 1, estructura

FASE	CONTROL ESTRUCTURA			
SISTEMA	Carrusel	Orientación contenedores		Posicionamiento batidora
FBC	Posicionamiento contenedores	Orientación C1	Orientación C2	Posicionamiento batidora
ETIQUETA	1A	1B	1C	1D
TIPO	Movimiento Circular	Movimiento circular	Movimiento circular	Movimiento Lineal
COMPONENTES	- Motor M1 - Sensores - Mecanismo de posición circular	- Motor M2 - Sensores - Mecanismo de orientación	- Motor M3 - Sensores - Mecanismo de orientación	- Motor M4 - Sensores - Mecanismo de posición lineal
ENTRADAS	Estado encóder carrusel	Estado sensor posición H y L	Estado sensor posición H y L	Estado sensor posición H y L
SALIDAS	Cambio posición	Cambio orientación	Cambio orientación	Cambio posición

Fase 2, control batidora

La Fase dos está centrada en el controlador de la batidora, su mecanismo es sencillo y el desarrollo se excluye de los demás sistemas al requerir la especialización de su unidad de control por uso de retroalimentación de estados para asegurar las revoluciones, pues de las especificaciones, Tabla 11, forma parte de los parámetros del proceso. Se identifica su esquema con componentes, Figura 39.

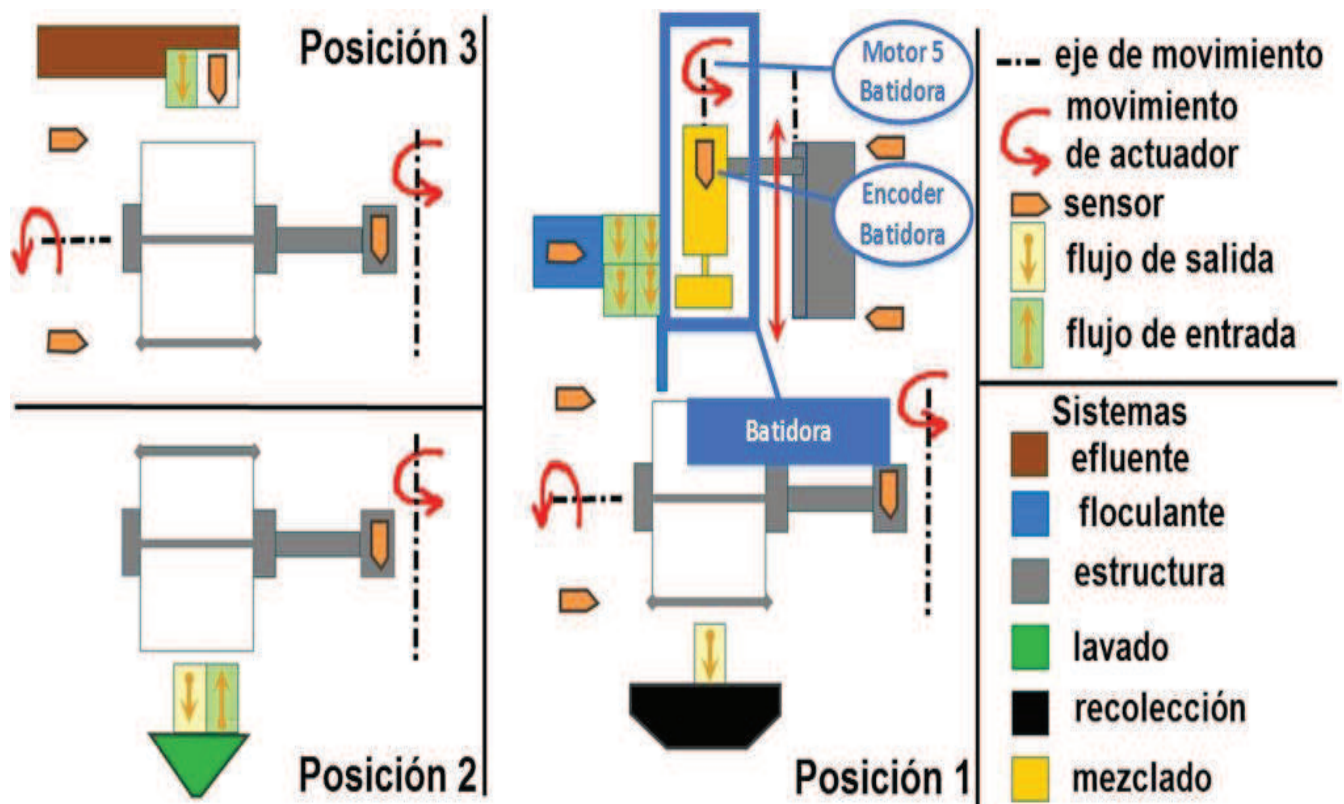


Figura 39. Esquema sistema de mezclado

Los componentes del FBC 2A son los mínimos para el concepto de módulo, el actuador junto con el mecanismo de engranes y el sensor permiten obtener una señal uniforme para procesar la velocidad. Además en el esquema se muestra la relación funcional que tiene 2A el sistema de posicionamiento 1D, esto es, físicamente solo hay una entrada para el FBC del sistema pero sus parámetros de configuración y operación serán determinados por los sistemas de la estructura. En la Tabla 18 se describe el FCB del sistema mezclado.

Tabla 18. FCB de la Fase 2, Batidora	
FASE	CONTROL BATIDORA
SISTEMA	Mezclado
FBC	Velocidad batidora
ETIQUETA	2A
TIPO	Velocidad
COMPONENTES	• Motor M5 • Sensor • Mecanismo de engranes
ENTRADAS	Estado encóder batidora
SALIDAS	Revoluciones

Fase 3, dosificación

La Fase tres está formada por dos sistemas de dosificación, uno para efluente y otro para floculantes; se componen de vías, de sensores y de actuadores para fluido; la precisión en cada sistema es diferente en función de la prioridad de la especificación, Tabla 11, y los sistemas forman parte de los parámetros del proceso. En la Figura 40 y 41 se muestra el esquema de los sistemas.

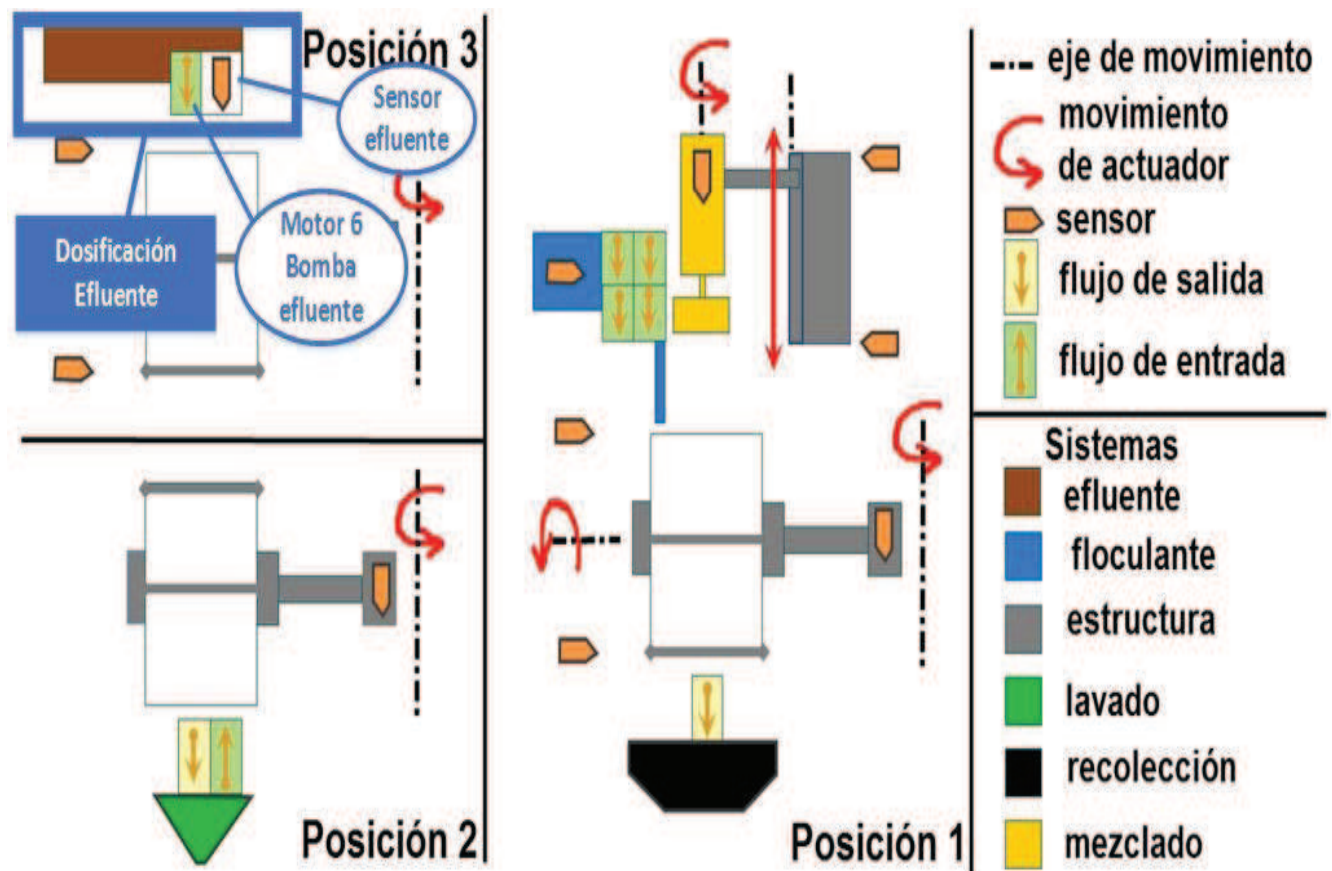


Figura 40. Esquema sistema dosificación efluente

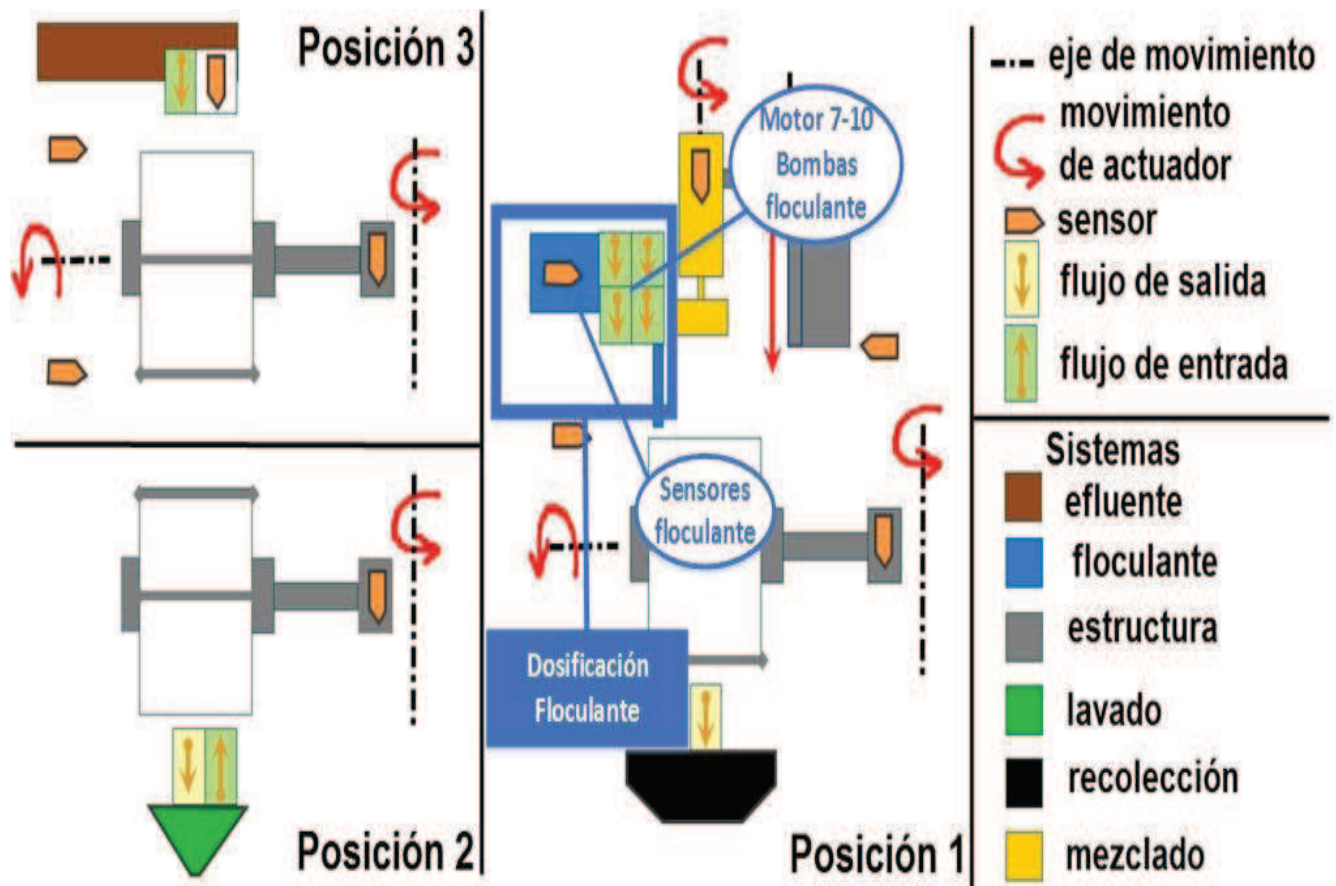


Figura 41. Esquema sistema dosificación floculante

Los sistemas de dosificación se encuentran en diferentes posiciones, tienen en común los mismos componentes y su diferencia está en la forma de medir la dosificación. Aun cuando el sistema de floculante tiene más actuadores el principio de funcionamiento es el mismo que del efluente, y a su vez el mismo que del sistema mezclado. Los FBC se encuentran descritos en la Tabla 19.

Tabla 19. FBC de la Fase 3, dosificación		
FASE	DOSIFICACIÓN	
SISTEMA	Dosificación efluente	Dosificación floculante
FBC	Lodo	Polímero
ETIQUETA	3A	3B
TIPO	Dosificación	Dosificación
COMPONENTES	- Motor M6 - Sensor - Recipiente - Vía de dosificación	- Motor M7 a M10 - Sensores - Recipiente - Vía de dosificación
ENTRADAS	Estado sensor de volumen	Estado sensores de volumen
SALIDAS	Volumen líquido	Volumen líquido

Los FBC fueron descritos en grupos por cada fase, cada uno fue descrito con sus entradas y salidas de forma independiente para identificar sus componentes; para poder establecer la dependencia funcional se identifica los datos internos entre los FBC y sus parámetros de configuración, sea por estados o por configuración. Esta dependencia y parámetros se definen a partir del Apéndice B, los grupos del proceso 4.1.1, estado del proyecto 3.1 y de la arquitectura, Tabla 16.

Las dependencias se describen como las condiciones de los estados de un FBC que requiere otro para realizar su función, sin ser todos los estados. Los parámetros por estados son indiferentes a los estados de dependencia, pues solo afectan en función del FBC, siendo entonces que los estados de dependencia permiten iniciar la acción al FBC. Para cada Fase se describe de forma general las dependencias y los parámetros de sus FBC en los siguientes puntos en orden de FBC dependiente, condiciones por estados y condiciones por iteraciones:

- **Fase uno**, al ser un grupo de sistemas de posicionamiento los FBC dependen de 1A para realizar sus operaciones, pues viene del sistema carrusel que es el soporte para todos los demás sistemas. Los datos internos son los estados de las posiciones de la Fase, en caso de 1D requiere el estado de operación de 2A pero no de forma dependiente. Las dependencias de los FBC son:
 - 1A no depende de otro FBC; estado posición 1B, 1C, 1D; iteraciones.
 - 1B depende de 1A; no requiere estados de condición; sin parámetros.
 - 1C depende de 1A; estado posición 1D; sin parámetros.
 - 1D depende de 1A; estado mezclado 2A, posición, 1C; sin parámetros.
- **Fase dos**, el sistema mezclado es un elemento que físicamente tiene limitaciones mecánicas para su actuación en función de su posicionado y por tanto de la posición de la estructura. Los datos internos corresponden al estado del sensor para la velocidad, los estados de su soporte 1D y el estado de dosificación. La dependencia del FBC es:
 - 2A depende de 1D; estado dosificación 3B; tiempo y numero de revoluciones.
- **Fase tres**, es un grupo de dosificación de fluidos, mecánicamente tiene las mismas limitaciones que en 2A al requerir que los contenedores estén orientados, en la posición uno esa dependencia corresponde a 1D por 1C y en la posición tres por 1B. Los datos internos son los estados de volumen de fluido, los estados de orientación de contenedores y sus parámetros de configuración; además 3B requiere que sus vías de dosificación estén llenas. Las dependencias de los FBC son:
 - 3A depende de 1B; estado posición 1A; cantidad de efluente.
 - 3B depende de 1D; estado vías en 3B; tipo y cantidad de floculante.

De manera general, un FBC que depende de estados de uno o más FCB permite heredar los estados de los que depende a otros que dependan del mismo, por ejemplo, no puede funcionar un FBC 1 que depende de un FBC 2, si para el mismo FBC 2 no se cumple la dependencia de un FBC 3; esta herencia permite obviar y reducir la necesidad de comprobar estados para el funcionamiento. De los puntos anteriores por ejemplo, 3A depende de 1B pero este a su vez de 1A.

Por otra parte, cada FBC requiere parámetros por configuración y por estados de condición para su función; cada FBC puede tener n parámetros, por lo que por dependencia toman uno o algunos que requieran para la función además de los estados auxiliares de condición. Por ejemplo, 3A requiere estados por dependencia de 1A y por condición de 1B, pero estos dos tienen todos los estados de sus FBC, por lo que 3A seleccionará los que permitan su función. La dependencia de cada FBC se describe en la Tabla 20.

Tabla 20. Dependencias de los FBC de las Fases de trabajo						
FCB	Posición	Procesos	Función asociada	Estados		Parámetros de configuración
				Dependencia	Condición	
1A	1, 2, 3	Posición contenedor	ninguna		Posición 1B, 1C, 1D	Iteraciones
1B	3	Orientación contenedor C1	1A	Posición 1A		
1C	1	Orientación contenedor C2	1A	Posición 1A	Posición 1D	
1D	1	Posición agitador	1A	Posición 1A	Mezclado 2A	
					Posición 1C	
2A	1	Mezclado	1D	Posición 1D	Dosificación 3B	Velocidad, rpm Tiempo revolución
3A	3	Dosificación efluente	1B	Posición 1B	Posición 1A	Efluente, ml
3B	1	Dosificación floculante	1D	Posición 1D	Estado vías fluido 3B	Floculante, ml
						Tipo floculante

A partir de la descripción de dependencias Tabla 20 se definen las funciones de cada FBC con base en los estados de las dependencias, los parámetros y los procesos en la posición en estructura, Apéndice A y B; el número de funciones y estados de los FBC depende de cada sistema y sus procesos.

Para la Fase 1, cada FBC tiene en común el cambio estado de posición y dependen de 1A por ser la estructura base. Cada FBC tiene una sola función con diferentes estados de condición, por lo que no se puede definir una misma función, Figura 42. La relación de funciones a partir del proceso y la posición de cada FBC es la siguiente:

- 1A, es la base del movimiento en la estructura, al mover una posición se afecta los estados de posiciones en 1, 2 y 3, por lo que es suficiente mover una posición de forma general, además por condición de operación debe moverse en dos tiempos para permitir cambios de orientación en 1C.
- 1B, 1C son FBC para un mismo sistema. Se encuentran en posiciones diferentes, tienen el mismo funcionamiento, pero requieren estados diferentes para su operación por condiciones físicas; 1B físicamente no afecta el movimiento de 1A, mientras que 1C sí al distinguir su orientación, por lo que debe regresar a posición en la mitad del cambio de 1A. esta distinción no permite unir estas funciones en una misma.
- 1D, se encuentra en la posición 1. En su estado de inicio permite el movimiento de 1A pero efectúa un movimiento lineal que depende de 1C por estar en la misma posición, donde 2A físicamente condiciona a la estructura mas no la hace dependiente de él, permite por operación la actuación de 3B, regresa a su estado inicial al terminar mezclado de 2A.

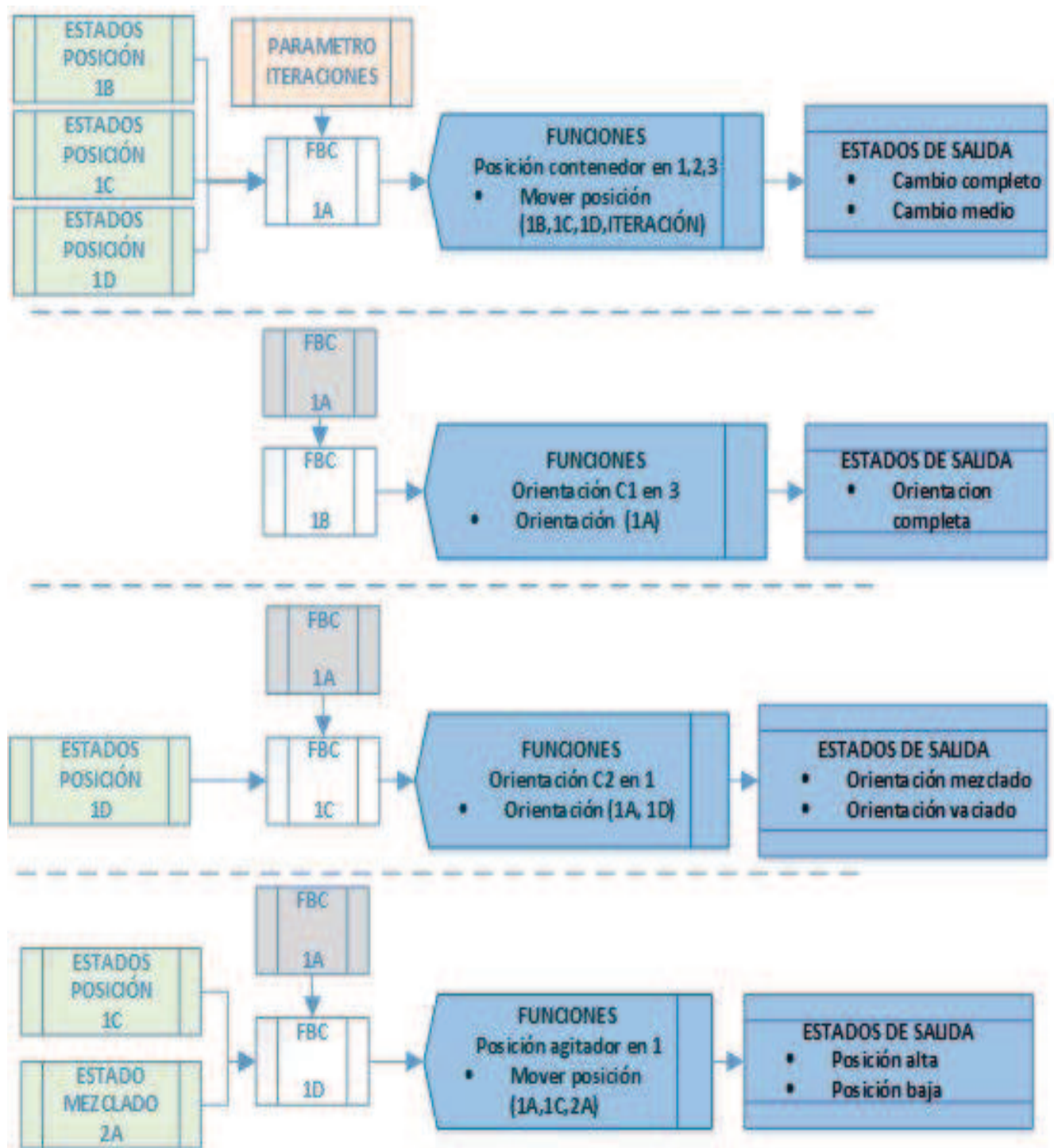


Figura 42. Funciones de FBC en Estructura

Para la Fase 2, se encarga de un solo sistema; físicamente se encuentra en la posición uno de la estructura y sobre 1D, del cual depende además de la configuración de sus parámetros, auxiliariamente toma el estado de dosificación de 3B para complementar sus funciones, Figura 43. Por condición de operación actúa de dos formas y solo genera un estado:

- Revuelve mientras baja, a partir de los estados de transición de 1D y 3B
- Bate cuando los estados de 1D y 3B estén completos, con sus parámetros

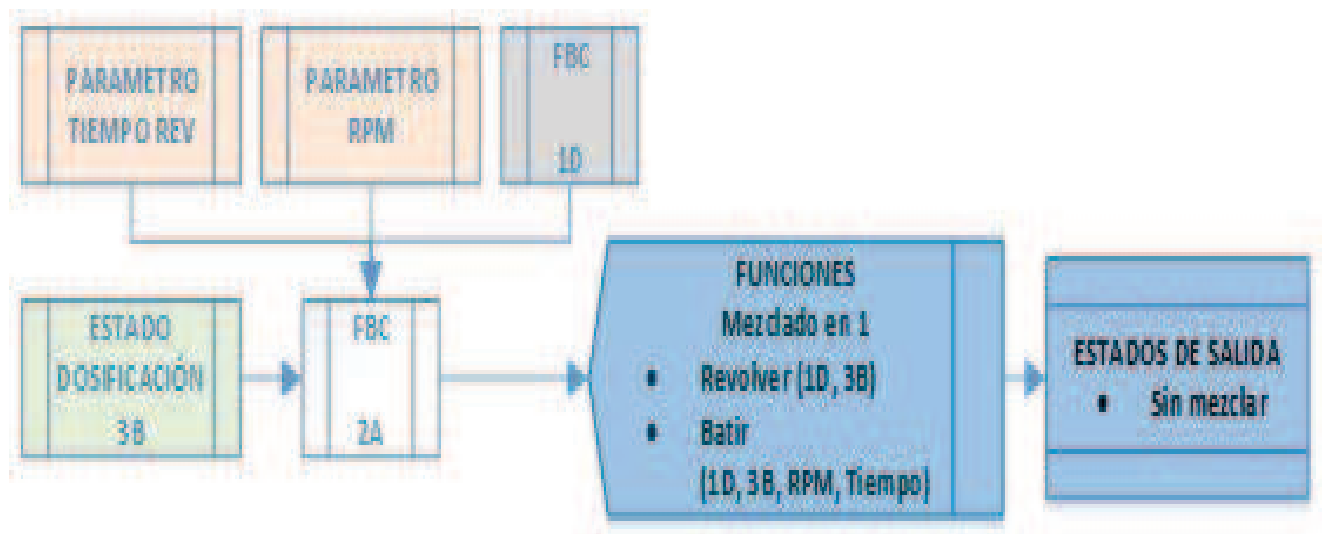


Figura 43. Funciones de FBC en Mezclado

Para la Fase 3, tiene dos FBC encargados de la dosificación, se encuentran en posiciones frentes, bajo las condiciones del sistema de orientación, ambos reciben parámetros de volumen de fluido, por lo que su actuación está limitada físicamente por los contenedores, Figura 44 Para cada FBC las funciones son:

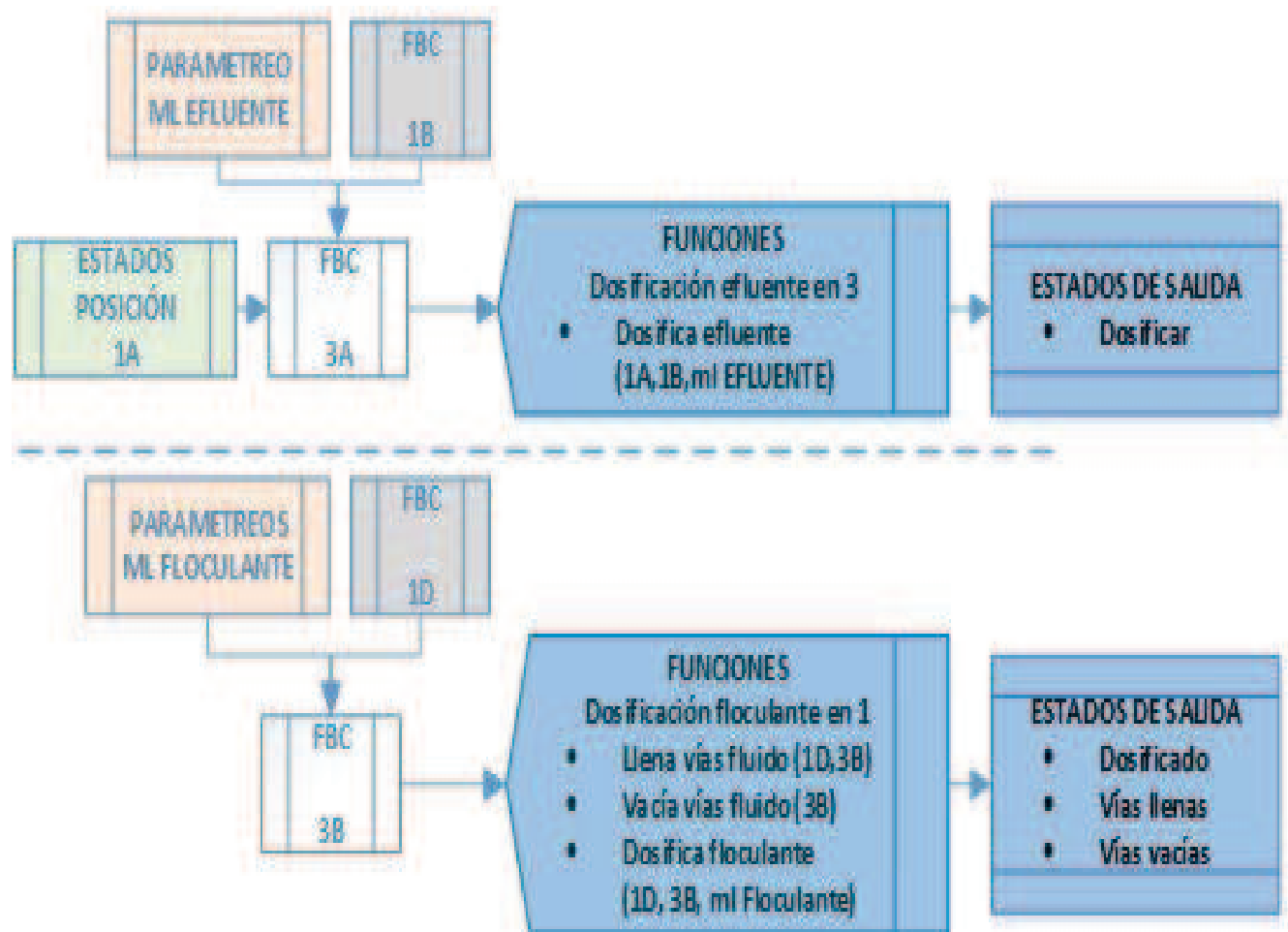


Figura 44. Funciones de FBC en Dosificación

- 3A, se encarga de dosificar el efluente en la posición 3, físicamente depende del estado de orientación de 1B y requiere condiciones de 1A para asegurar la dosificación.
- 3B, se encarga de dosificar el floculante para la posición 1, físicamente depende de estado de orientación de 1D. Su actuación puede definirse en 2 tipos de funciones, unas para acondicionar sus vías de dosificación y otra para acondicionar; las primeras no requieren una dependencia directa y la segunda depende del estado de las vías.

La categoría de los FBC se determina por la relación de todos los estados de dependencias de un sistema; permiten clasificar las funciones de acuerdo a los procesos de operación, Tabla 21. Esta clasificación depende de los estados de salidas de otros FBC, así se definen los estados de salida como todos los posibles estados de operación de un sistema, esto implica que:

- De las operaciones del proceso se establecen los estados de las dependencias en función las condiciones físicas.
- Los estados de cada función se enumeran del 1 a n.
- Los estados 0 son los de estado de transición para las funciones.
- El estado 1 representa los estados de inicio, de parada y puesta en marcha.

Tabla 21. Niveles de jerarquía de los FCB					
NIVEL	FBC	FUNCIÓN	ESTADOS ENTRADA	CONDICIONES OPERACIÓN	ESTADOS DE SALIDA
1	1A	Mover posición	Iteraciones Posición 1B, 1C, 1D	1B (en 1), 1C (en 1), 1D (en 1) Iteraciones	Posición 1A (1,2) 1 cambio completo 2 cambio medio
2	1B	Orientación	Posición 1A	1A (en 1)	Orientación 1B (1) 1 Orientación completa
	1C	Orientación	Posición 1A, 1D	1A (en 2), 1D (en 1)	Orientación 1C (1,2) 1 Orientación mezclado 2 Orientación vaciado
	1D	Mover posición	Posición 1A, 1C Mezclado 2A	1A (en 1), 1C (en 1) 2A (en 1)	Posición 1D (1,2) 1 posición alta 2 posición baja
	3B2	Llenar vías	Estado vías 3B2 Posición 1D	1D (en 1) 3B2 (en 1)	Vías B2(1,2) 1 Vías vacías 2 Vías llenas
		Vaciar vías	Estado vías 3B2	3B2 (en 2)	
3	2A	Revolver	Velocidad, Tiempo Posición 1D, 3B	1D (en 0), 3B1 (en 0)	Mezclado 2A (1) 1 sin mezclar
		Batir		1D (en 2), 3B1 (en 1)	
	3A	Dosificar efluente	Posición 1B, 1A MI	1A (en 1), 1B (en 1)	Dosificación 3A (1) 1 Dosificado
	3B1	Dosificar floculante	Posición 1D Estado vías 3B2 MI, tipo	1D (en 0) 3B2 (en 2)	Dosificación 3B1 (1) 1 Dosificado

Finalmente, en la Figura 45 se muestra los niveles de jerarquía de la Tabla 21, la clasificación en la estructura funcional del proceso a partir de los componentes permite establecer los bloques funcionales al determinar las condiciones de operación en relación al sistema físico, componentes; por lo que el análisis de la estructura de operación puede ser definición a través de los tiempos de operación en función de las jerarquías y con ello se puede establecer el puente entre las funciones de FBC y las funciones de la UC.

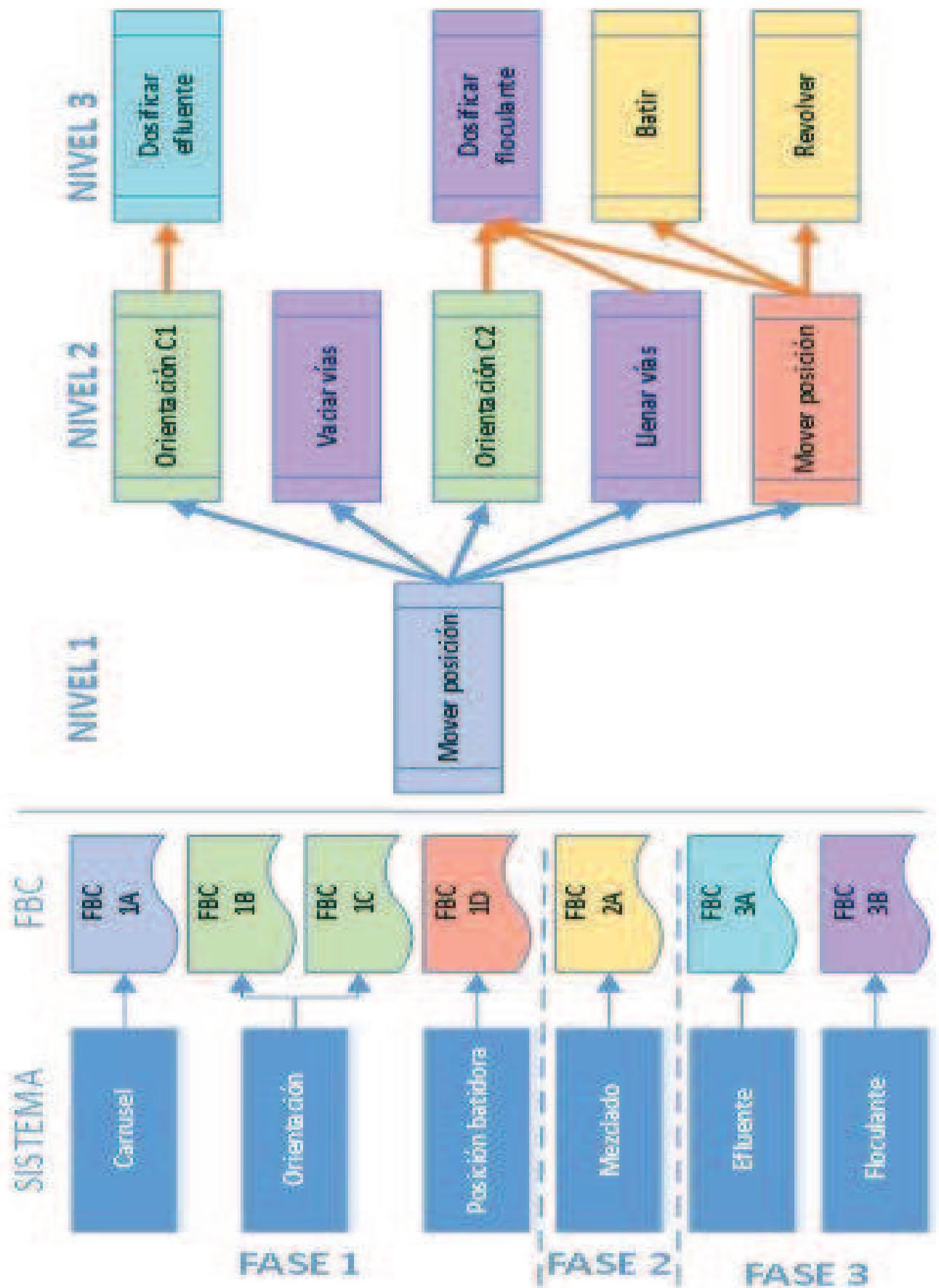


Figura 45. Jerarquía de funciones de FBC

A partir de la Figura 45 y Tabla 21 se establecen los estados de inicio del grupo de los componentes para cualquier operación, Tabla 22. La justificación de estos estados se encuentra en el Apéndice B, donde se analiza el proceso secuencial y se establece uno modular modificando las condiciones de inicio propuestos en la primera etapa del proyecto.

Tabla 22. Estados de inicio de sistemas

Parámetros iniciales	Carrusel 1A	Orientación		Posición batidora 1D	Mezclado 2A	Efluente 3A	Floculante 3B
		1B	1C				
Estados	1	1*	2*	1	1	1	1

* El estado de inicio del sistema de orientación debe mantener el contenedor en 1B boca arriba mientras que en 1C boca abajo, por condición física de sus FBC en el proceso, Apéndice B

4.1.4 Estructura de operaciones

En sentido estricto, antes de definir bloques funcionales en la unidad de control primero se analiza la estructura de las funciones de operación a partir de las operaciones de proceso de un sistema. Una operación del proceso en la unidad de control, **UC**, se define como la ejecución de una o más funciones de un sistema físico con estados de condición de entrada y salida definidos.

A diferencia de las operaciones de proceso, las operaciones de UC son a un nivel lógico programable, **LP**, pero mantiene una estrecha relación con el proceso, por lo que para definir las funciones primero se identifican en relación con los módulos en el proceso, luego se asocian a un grupo de operación en la UC y finalmente se des la estructura por tiempos de operación, que posteriormente permite representar gráficamente a todo el proceso tanto en la UC como físicamente.

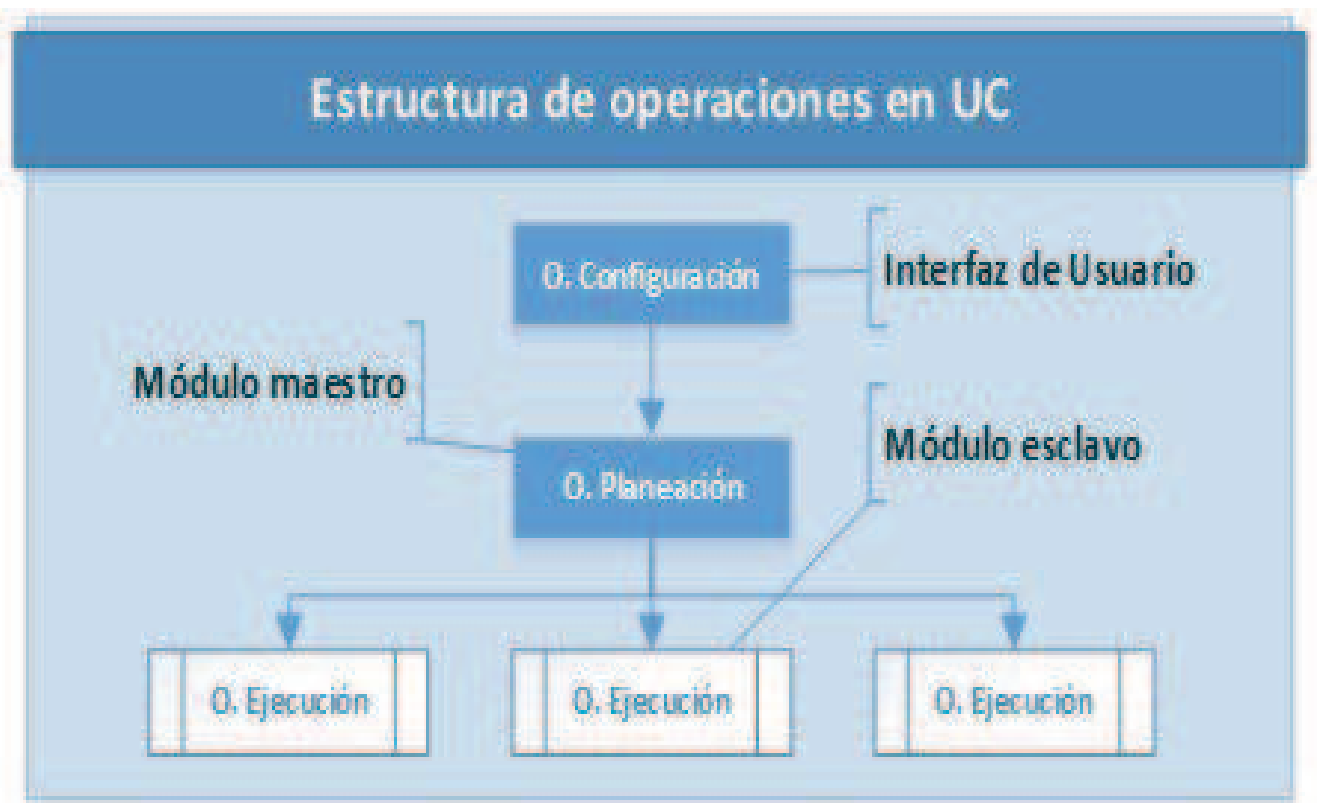


Figura 46. Estructura de operaciones en UC

Las operaciones son de tres tipos, Figura 46: para la ejecución del proceso, para su planeación y para su configuración. Las operaciones de ejecución se relacionan directamente con las funciones de FBC. Por otra parte, se requieren elementos externos independientes físicamente al sistema que realicen las otras operaciones. Hasta este punto del diseño se define la interfaz de usuario y el módulo maestro como elementos externos y dependientes al proceso, que junto con los sistemas basados en FBC forman las operaciones LP que la UC gestiona.

Las operaciones de la UC se clasifican en seis de acuerdo a los cuatro grupos de operación del proceso, Figura 36. Esta asociación se determina en función de las dependencias de los módulos que limitan los estados de inicio de tarea, los estados esperados e identifican errores, Tabla 23.

Tabla 23. Relación de los grupos de operación de la UC y del proceso					
Grupos de operación		Tareas	Estado de inicio	Estado Esperado	Errores
Proceso	UC				
Parada Puesta en marcha	Preparación Mantenimiento Configuración	Preparación Configuración Mantenimiento	Inicial Parámetros	Sin errores Sin transición	Sistemas Estados de inicio diferentes al esperado
Funcionamiento	Marcha Fin	Tiempos de operación Secuencia Reporte eventos	Transición Lógicos en tiempo de operación	Fin tiempo de operación	A partir de los estados de transición y los esperados
Falla	Fin por errores	Detener proceso Código de error Notificación a usuario	Transición	Siguiente al anterior	Identificación de errores, por estados de transición y comparación de estados esperados

Las operaciones representan a las funciones de los FBC en el proceso, estas pueden ser ejecutadas en grupos para realizar una secuencia o bien por partes para tareas de mantenimiento o configuración. Las funciones están formadas de uno o más componentes desde el punto de vista del autómata programable, cap. 2.4, que permiten definir el tipo de ejecución de funciones, estos se clasifican en:

- **Gestión del sistema.**
- **Lectura del estado de las entradas.**
- **Ejecución del programa almacenado.**
- **Estructura de datos de salida.**

De las jerarquías de componentes se determinó funciones de forma general en el proceso a partir de componentes, ahora se traducen a funciones LP. La definición de los componentes de función se establece con la asociación con grupos de operaciones del proceso, Tabla 24.

Tabla 24. Componentes de función			
Componentes de función	Grupo de operación del proceso		
	Parada y puesta en marcha	Funcionamiento	Falla
Gestión sistema	Preparar estados de inicio, tarea y error en operaciones por módulo	Cambia estados de transición y posición esperadas por tiempo de operación	Comparar estado actual y esperado

Lectura estados	Sensores y operaciones: Estado Actual	Sensores y tareas: Anterior, esperado y actual	Sensores y estado anterior
Ejecución programa	Cambios de estados por operaciones y configuración	Funciones en secuencia	Detener, identificar y avisar
Estructura datos salida	Estado actual de sensores, operaciones y datos	Estado de tareas	Estado de tarea, sensores y error

Para cada tipo de operación de la UC se establece la relación funcional del proceso con los módulos manteniendo la jerarquía de componentes y las operaciones del proceso. Se define primero las funciones de operación de ejecución por ser elementos dependientes del proceso. Las consideraciones para sus definiciones son:

- Funciones indican inicio y fin de tarea sin alterar parámetros ni estados de su operación.
- Detección de errores en paralelo a tareas de operación.
- Configuración de parámetros en parada y puesta en marcha.
- Datos de salida formados por variables de operación y estados de tarea y error.
- Líneas de comunicación independientes a tarea y error.

En la Tabla 25 se muestra los sistemas con sus componentes y funciones asociadas con base en los puntos anteriores, la etiqueta hace referencia al módulo que se desarrollará. En cuanto a componentes de función, la estructura de datos de salida representa la comunicación entre módulos y maestro mientras que gestión, lectura y ejecución son propios de cada módulo. Además, la definición de líneas de comunicación independiente a tarea y error permiten generalizar los estados de los módulos, esto por la definición del tipo de sistema modular, cap. 3.5.

Tabla 25. Funciones de operación de ejecución						
Etiqueta	Estructura			Mezclado	Lodo	Polímero
Sistemas	Carrusel	Orientación contenedor	Posición batidora	Batidora	Dosificación efluente	Dosificación floculante
FBC	1A	1B, 1C	1D	2A	3A	3B
Función	Mover posición: • Completa • Mitad Bidireccional	Orientación: • Un sentido • Bidireccional	Mover posición: • Bajar • Subir	Mezclar: • Revolver • Batir	Efluente: • Dosificar	Floculante: • Llenar vías • Vaciar vías • Dosificar
Estado función	0,1,2	0,1 - B 0,1,2 - C	0,1,2	0,1,2	0,1	0,1 - B1 0,1,2 - B2
Datos	Error, Tarea, Estados					

Por otra parte, para el módulo maestro e interfaz se definen sus funciones por los alcances de sus operaciones a partir de las de ejecución debido a que sus operaciones LP son a nivel de planeación y de configuración, y sus cartas de función representan grupos de funciones de los sistemas en tiempos de operación. Las operaciones de la interfaz y el maestro son:

- Maestro: se encarga de la planeación de las funciones a partir de las operaciones que pida la interfaz, mediante lectura de estados operación y de error en cada tiempo de operación. Traduce las órdenes de la interfaz de usuario para configurar, probar o poner en marcha iteraciones o procesos por envío de datos de módulos. Debe permitir una comunicación

paralela para integrar más elementos, sean módulos o monitores con el fin de mantener la propiedad modular del sistema.

- Interfaz: Se encuentra dividido en dos partes, la que interpreta, tradúcelas acciones de usuario y despliega información, y la que se encarga de generar reportes de estados, de errores, de iteraciones y de configuraciones. Para determinar las operaciones en pantalla se sigue el grupo de operación de ejecución de la UC. Las funciones de la interfaz permiten asociarlas a un grupo menú para el usuario, que ejecuta de forma individual cada función, pero todas las funciones se agrupan según el tipo de operación en la UC.

La interfaz de usuario está desarrollada en el Apéndice C, donde se especifica el desarrollo de los menús a partir de los módulos del sistema mecánico; estos menús tienen como objetivo ordenar operaciones en grupos para hacerle intuitivo al usuario el uso de la UC, pero las funciones se establecen y clasifican de acuerdo al grupo de operación.

Se agrega un monitor como parte del módulo maestro pero su definición y desarrollo queda limitado conceptualmente hasta este capítulo por los alcances de esta tesis. Sus funciones de operación son independientes a los estados de operación de ejecución y planeación, pues su presencia en el sistema no afecta ningún parámetro de configuración pero si permite ejecución de operaciones. En la Tabla 26 muestra las funciones de la interfaz y del maestro.

Tabla 26. Funciones de operación de planeación y configuración				
Etiqueta	Interfaz		Maestro	
Sistema	Pantalla	Reportes	Maestro	Monitor
Función	Menús: • Conexión • Configuración de parámetros • Secuencia • Monitoreo: Prueba individual de módulos - Estado sensores - Prueba estructura - Prueba Batidora • Herramientas: Lavado - Purga de vías - Purga de bombas - Reset de estados - Mensajes a operador: • Error • Notificación • Confirmación	Registro eventos: • Configuración • Conexión • Pruebas • Iteraciones: Errores - Operaciones Cada evento registrado almacena datos dividido en dos tipos de archivo: registro por día y por evento Formato visual para usuario	En parada y puesta en marcha • Envío a módulos: Parámetros - Operación - Secuencia • Envío a interfaz: Datos de estados En funcionamiento • Lectura estados de tarea y error En falla • Lectura tipo de error • Forzar paro	Registros: • Envío de datos M-E • Envío datos E-M • Cambio de estados tarea y error Lectura: • Estados módulos • Datos módulos Acciones: • Conexiones • Paro emergencia • Llamar función Operaciones en parada Ejecución funciones de módulo El uso de funciones exige retorno a estado de parada No se alteran estados ni parámetros
Datos	Parámetros	Parámetros Datos de módulos Datos de operación	Datos de módulos Parámetros	Datos de módulos Datos de operación

Las operaciones de la UC a partir de las operaciones del proceso representan las funciones de operación de cada módulo a nivel LP. La clasificación de las funciones en grupos de operaciones permite establecer tiempos de operación de trabajo, estos grupos se definen los grupos como:

- Preparación: las funciones que determinan estados de inicio.
- Marcha: una secuencia de funciones.
- Fin: funciones que permiten a los sistemas regresar a su estado de inicio al terminar marcha.
- Fin por errores: reinicio estados de inicio y des habilitación de operaciones al interrumpirse la marcha o alguna función de prueba.
- Mantenimiento: funciones de prueba de cada módulo y limpieza.
- Configuración: función de entrada parámetros y cambio de estados.

Se muestra en la Tabla 27 los grupos de operación de UC con sus módulos asociados a los tipos de operación; en las de planeación solo se muestra el maestro debido a que el monitor es independiente al proceso y permite la ejecución de cualquier operación con la única condición de no cambiar la configuración de parámetros. Por otra parte, se agregan los dos sistemas de las etapas posteriores del proyecto: lavado y recolección muestra, son de forma indicativa para referencia en siguientes etapas.

Tabla 27. Funciones de grupo de operación en UC					
Grupos de operación	Funciones de módulos en tipos de operaciones				
	Configuración		Ejecución		Planeación
Preparación	Pantalla	Monitoreo: • prueba de estructura Mensaje: • Notificación	Estructura	Orientación contenedor: • Orientación B1 • Orientación C1	Maestro
	Reportes	Registro: • Pruebas		Carrusel	
Puesta en marcha	Pantalla	Secuencia Mensaje: • Notificación • Confirmación	Estructura	Carrusel	
				Orientación contenedor: • Orientación 1B • Orientación 1C	
				Posición batidora	
	Reportes	Registro: • iteraciones	Mezclado	Batidora	
			Lodo	Dosificar	
			Polímero	Dosificar 3B2	
			Lavado		
			Recolección muestra		
Fin	Pantalla	Herramientas: • Lavado Mensaje: • Notificación	Estructura	Carrusel	
				Orientación contenedor: • Orientación 1B • Orientación 1C	
	Reportes	Registro: • Iteraciones	Lavado		

Fin por errores	Pantalla	Monitoreo: • Prueba estructura Conexión Mensaje: • Error	Estructura	Carrusel
				Orientación contenedor: • Orientación 1B • Orientación 1C
				Posición batidora
			Mezclado	Batidora
			Lodo	Dosificar
	Reportes	Registro: • Iteraciones • Pruebas • Conexión	Polímero	Dosificar 3B2
			Lavado	
			Recolección muestra	
Mantenimiento	Pantalla	Conexión Monitoreo Herramientas: • Lavado • Purga Mensaje: • Notificación • Error	Estructura	Carrusel
				Orientación contenedor: • Orientación 1B • Orientación 1C
				Posición batidora
			Mezclado	Batidora
			Lodo	Dosificar
	Reportes	Registro: • Conexión • Pruebas	Polímero	Dosificar 3B1 Vaciar vías 3B2 Llenar vías 3B2
			Lavado	
			Recolección muestra	
Configuración	Pantalla	Configuración de parámetros Herramientas: • Reset de estados Mensaje: • Notificación • Confirmación • Configuración	Estructura	
			Mezclado	
			Lodo	
			Polímero	
	Reportes	Registro: • Configuración	Lavado	
			Recolección muestra	

Para representar a los grupos de operación como líneas de tiempo de funciones, cada tiempo de operación debe cumplir los siguientes puntos:

- Estados lógicos y de transición: un estado lógico es aquel que no es de transición.
- Condiciones de inicio: las condiciones de inicio de una función son estados lógicos y en unos casos de transición.
- Inicio y fin de funciones: toda función inicia y termina sin estados de transición.
- Inicio y fin de tiempos de operación: el inicio y fin está marcado cuando todas las funciones terminan en estados lógicos.

- Funciones en serie: cuando hay dos funciones o más en secuencia, la anterior debe terminar en es estados lógicos sin transición para la siguiente.
- Funciones en paralelo: el inicio de funciones en paralelo es a partir del fin de un tiempo de operación, las condiciones de inicio pueden ser estados lógicos o de transición.
- Tiempos multitarea: grupos de funciones en serie o en paralelo.
- Ciclo de trabajo: un ciclo tiene tiempos de operación definidos.

Las funciones de los grupos de operación permiten definir tiempos de operación como un conjunto de funciones que realizan un ciclo de operación, que representa una parte de un proceso, con entradas, salidas y estados internos bien definidos. Esta definición permite planificar operaciones por tiempos, además permite generar análisis de cargas y consumos de corrientes, lo cual queda fuera de los alcances de este trabajo.

Cada tiempo de operación sigue la jerarquía de componentes y en caso de la interfaz y maestro se sigue las operaciones que permiten sus funciones. En los siguientes diagramas se representa los tiempos de operación para cada grupo de la Tabla 27, se mantiene los colores de la Figura 45 y los estados de inicio de la Tabla 22.

Figura 45. El grupo de preparación representa las operaciones para verificar los estados del carrusel por tener mayor jerarquía en módulos. Los estados no se alteran y permite identificar errores por transición.

La puesta en marcha, Figura 48, es un grupo de operación completo desde el punto de vista del autómatas por ser la parte principal de la automatización del proceso, esta se puede dividir en tres secciones:

- Configuración y planeación, Figura 53 a partir de la instrucción del usuario hasta el envío de parámetros de configuración de los módulos por parte del maestro. Se agrega un bloque de prueba carrusel debido a la verificación de errores antes de la puesta en marcha, al ser un módulo con mayor prioridad.
- Inicio, Figura 49: la primera iteración tiene estados lógicos que no permiten la ejecución de algunas operaciones, como lavado y recolección muestra.
- Iteraciones, Figura 50: a partir del último tiempo de operación del inicio se repiten los tiempos hasta completar las iteraciones.
- Figura 51. Al igual que el inicio en puesta en marcha, fin se ejecuta para llevar todos los estados a inicio, acondiciona los contenedores para una nueva iteración.
- El grupo de error, Figura 52, a diferencia de los demás grupos es llamada por evento en módulo, en cualquier otro tiempo del grupo de operación para identificar el error y llevar todo el sistema estado inicial sin importar el estado que tenga, “x”.

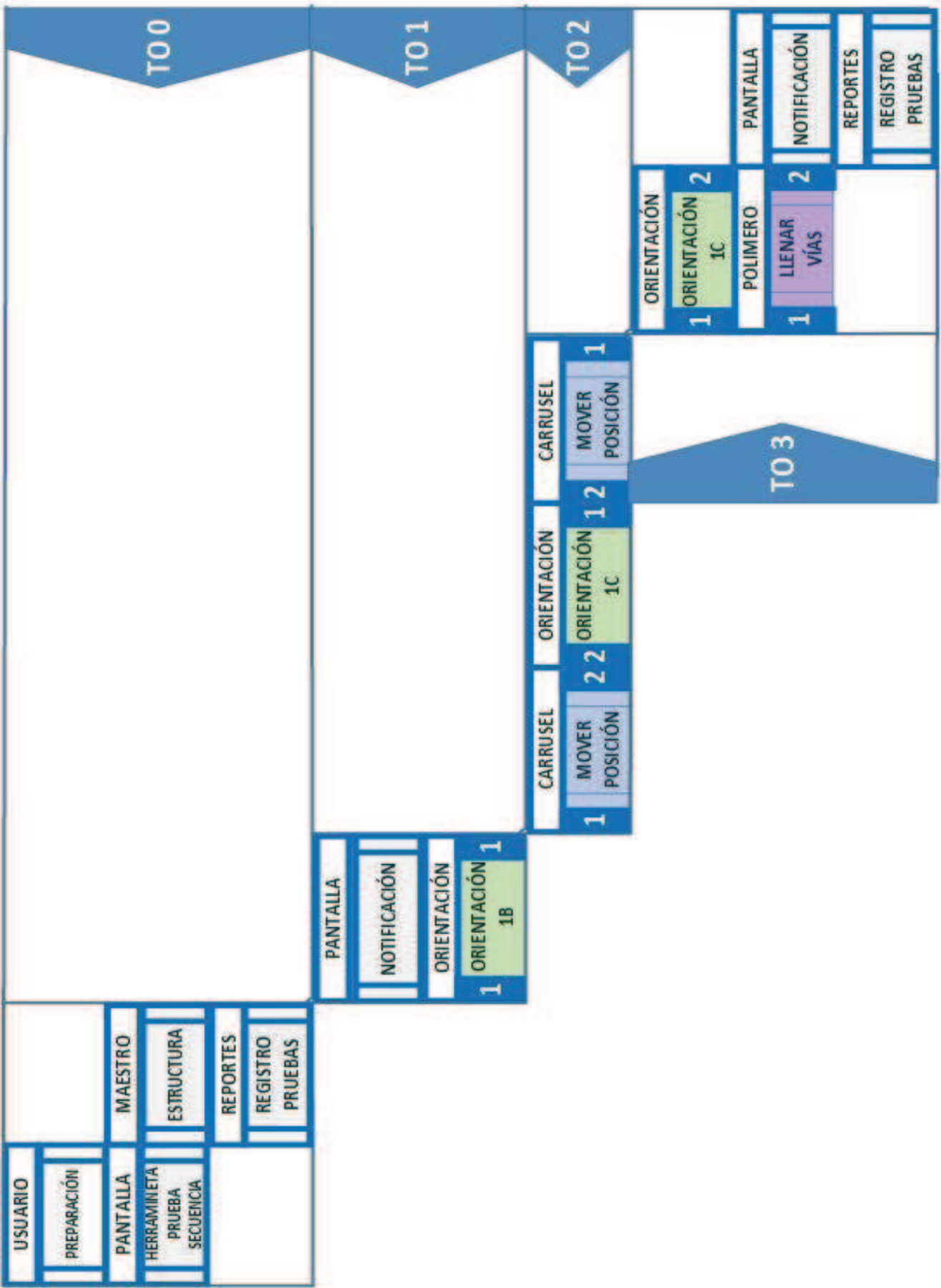


Figura 47. Tiempos de operación: Preparación

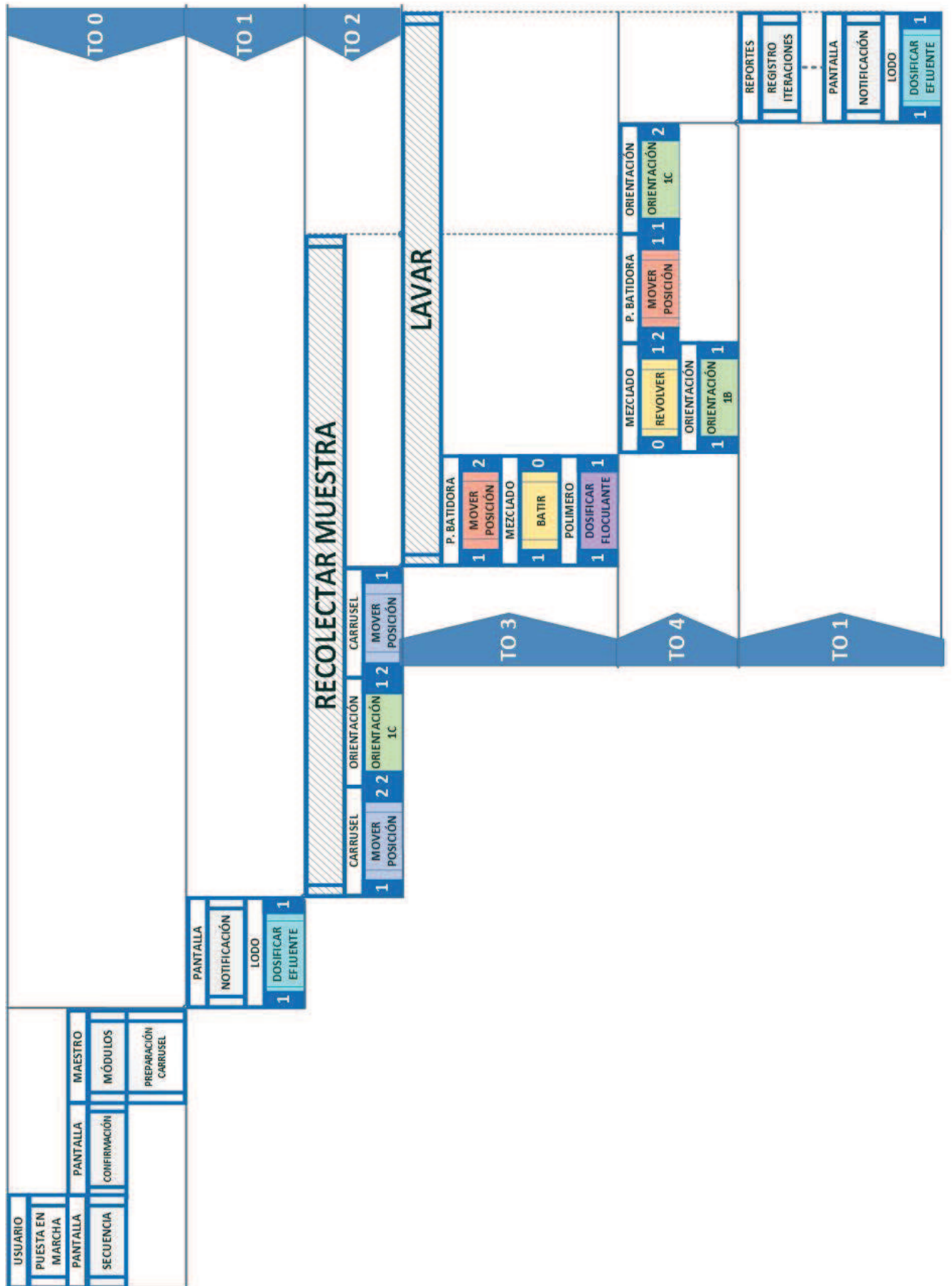


Figura 48. Tiempos de operación: Puesta en Marcha

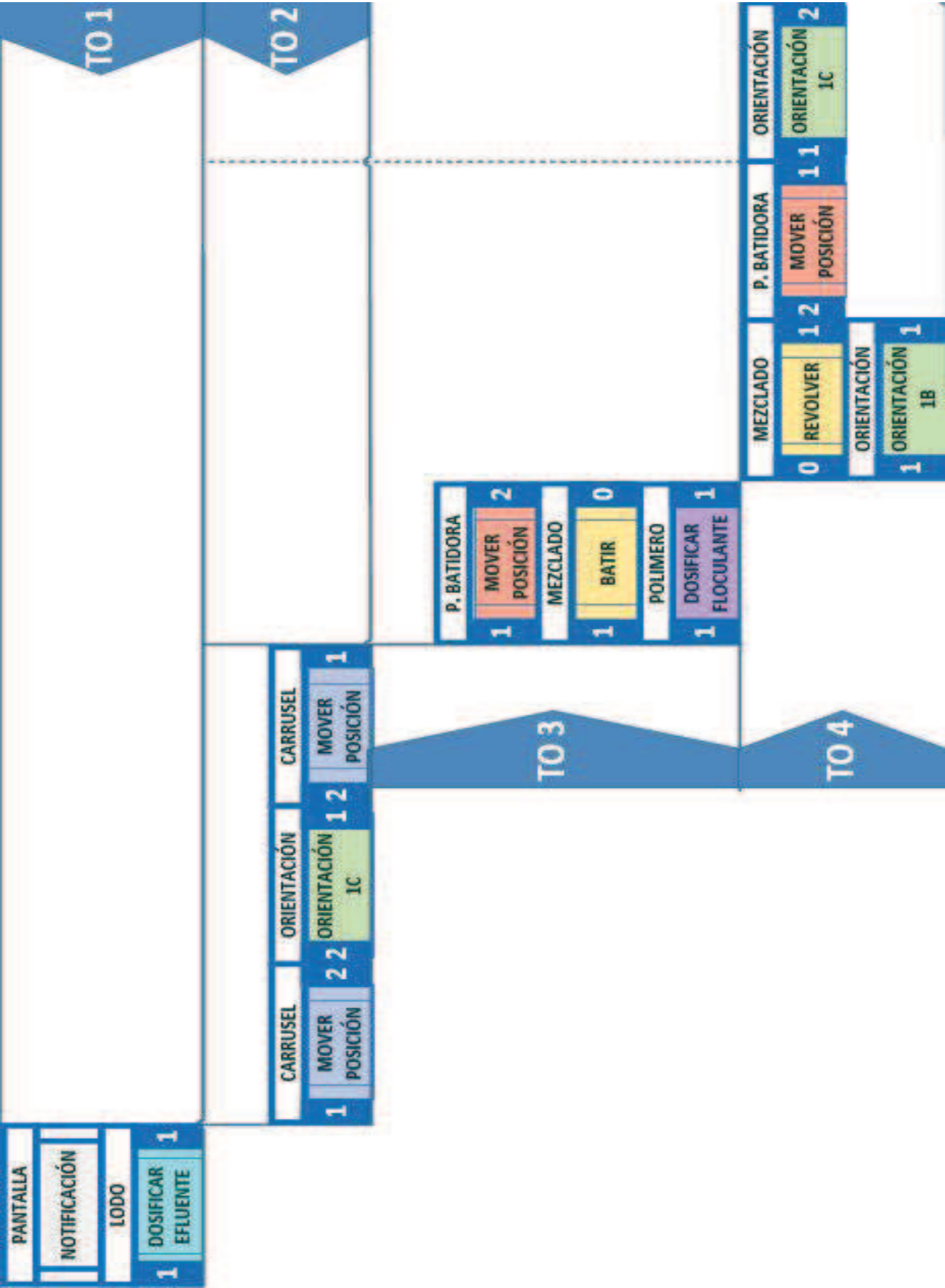


Figura 49. Tiempos de operación: Puesta en Marcha, 2

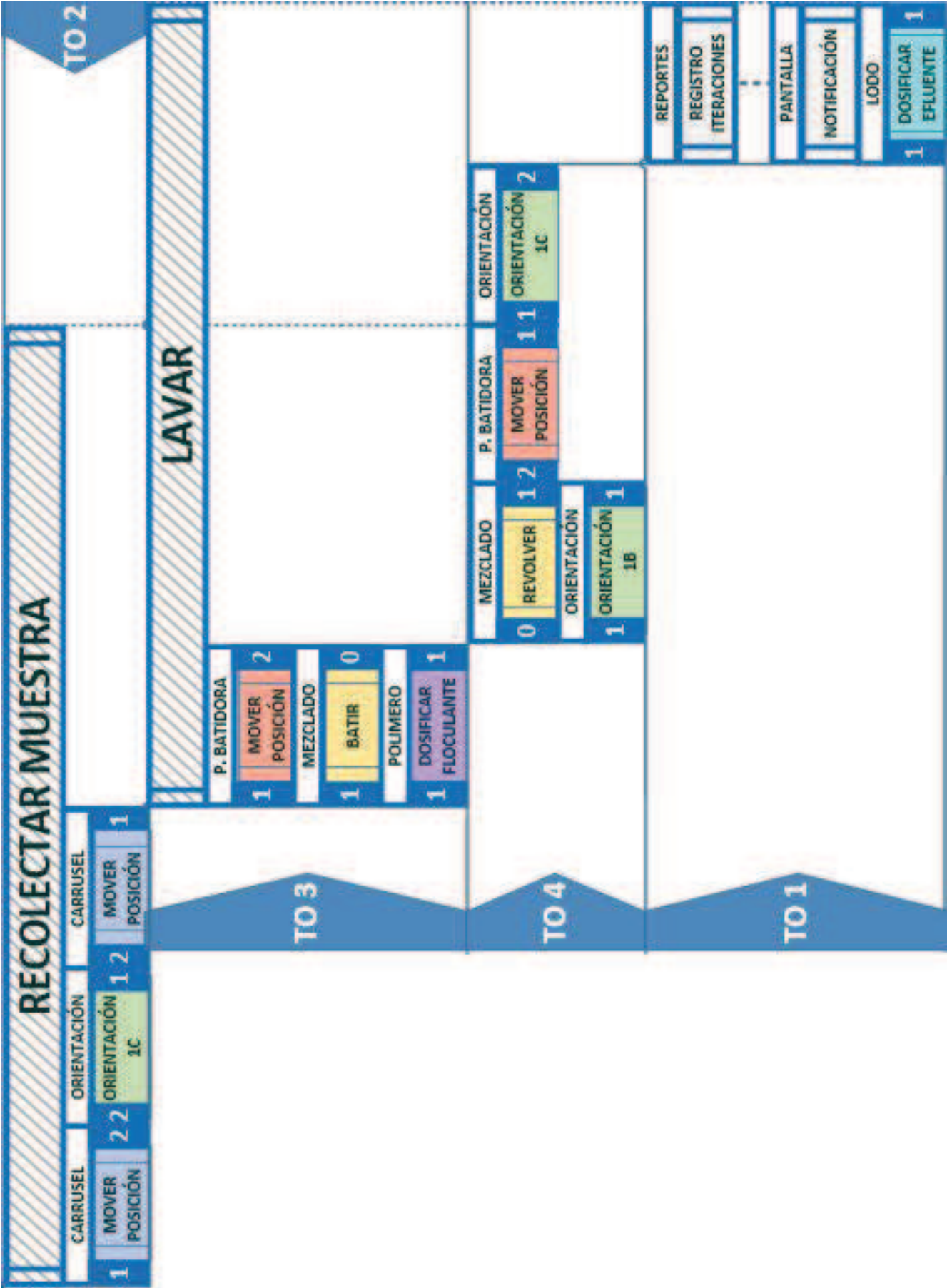


Figura 50. Tiempos de operación: Puesta en Marcha, 3

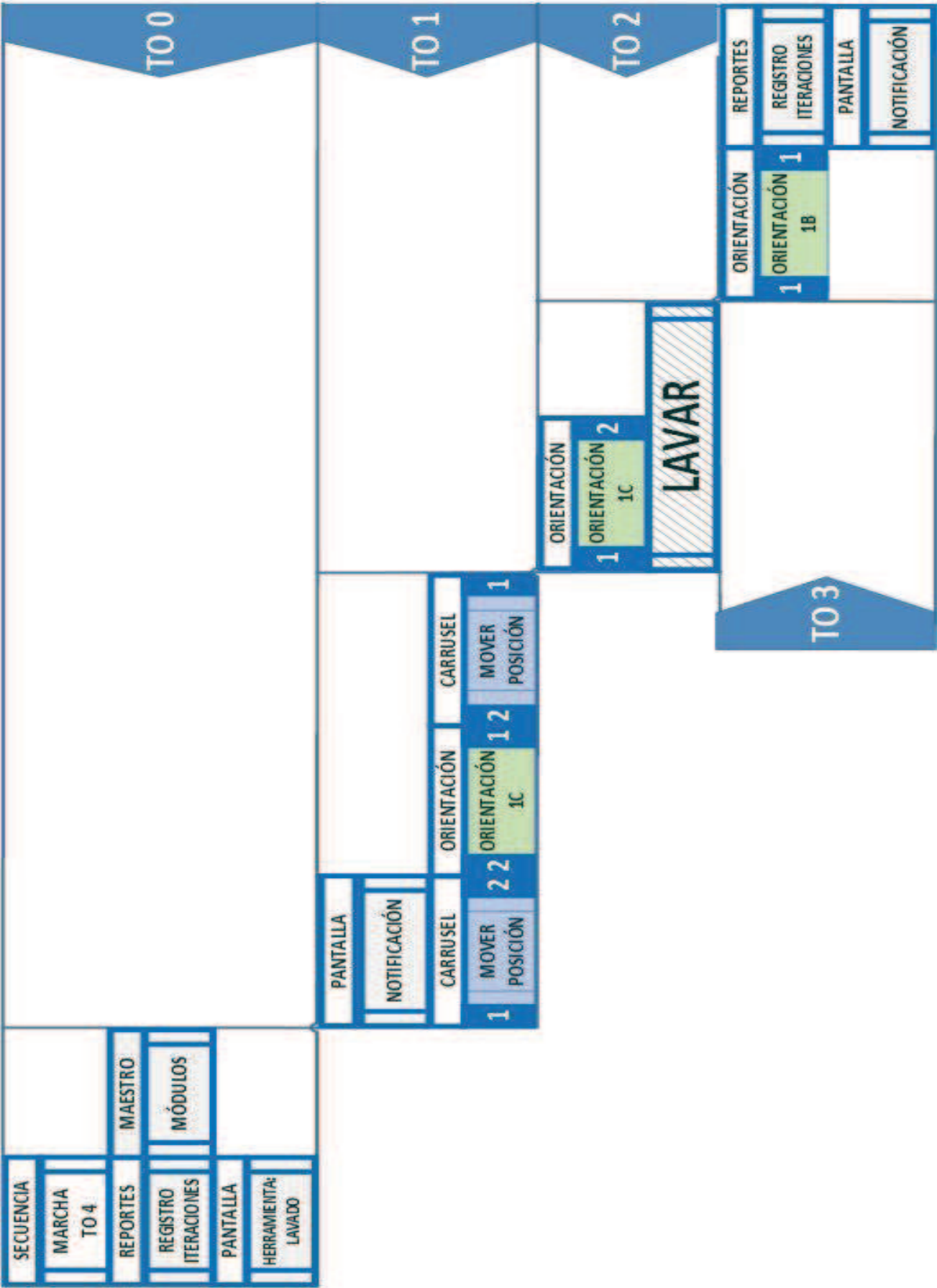


Figura 51. Tiempos de operación: Fin de marcha

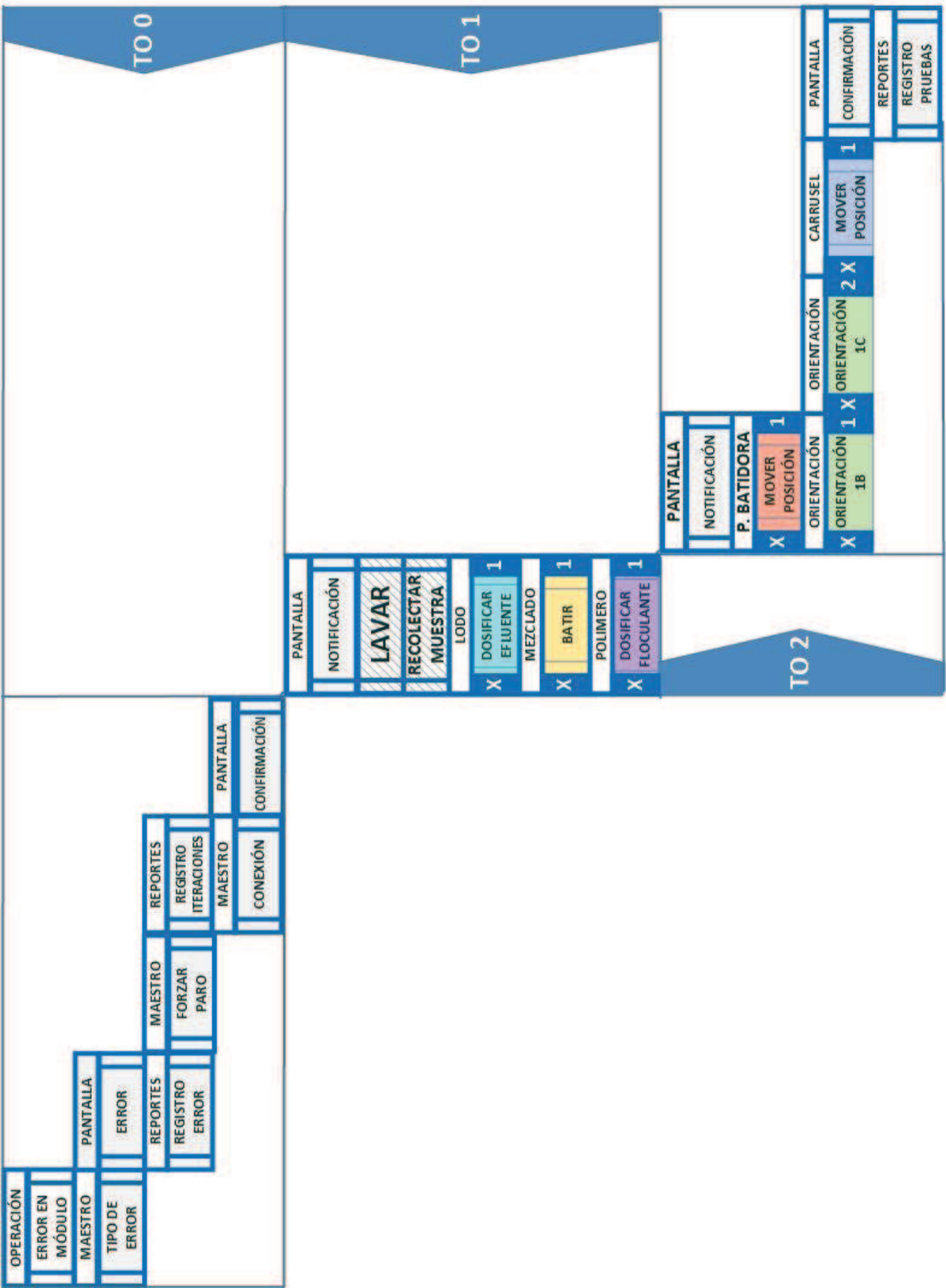


Figura 52. Tiempos de operación: Fin por error

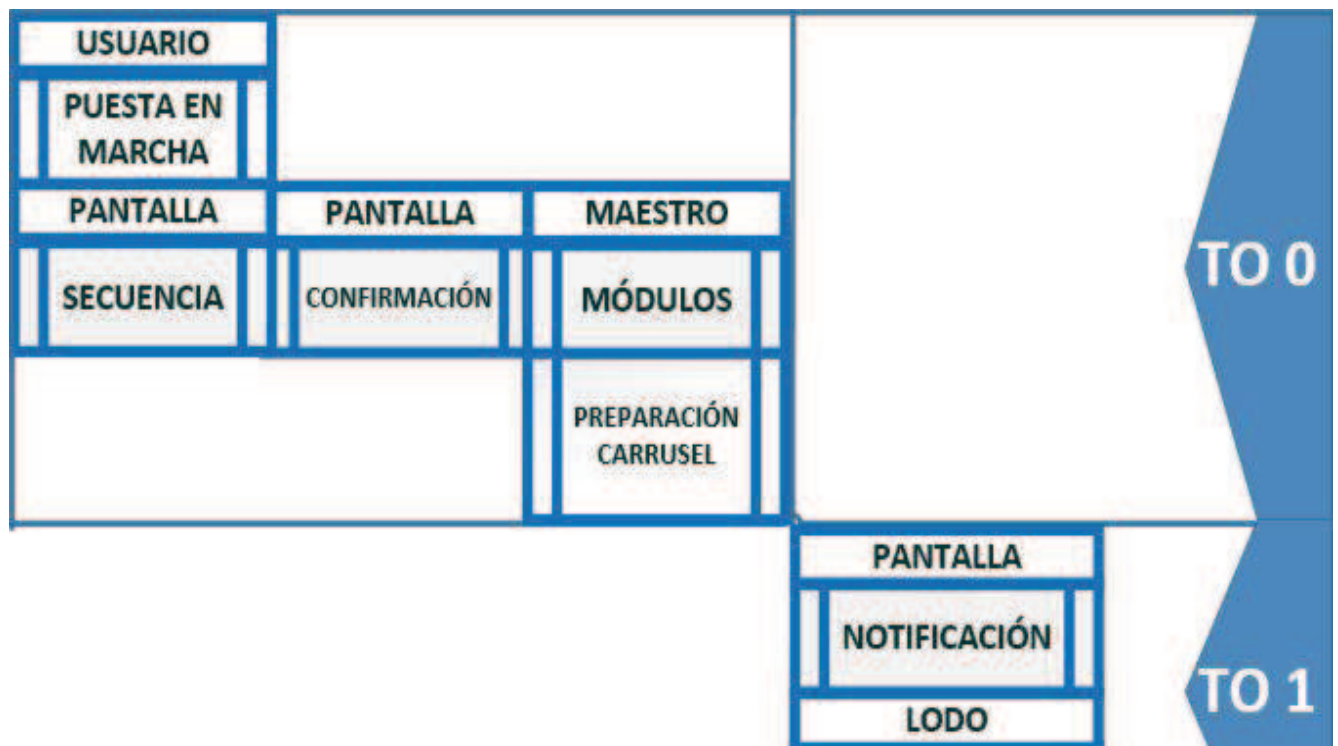


Figura 53. Tiempos de operación: Puesta en marcha, I

El grupo de mantenimiento y de configuración no son representados por tiempos de operación debido a que son grupos que llaman de forma independiente a funciones de módulo y depende del usuario la duración de operación.

Como se mencionó al inicio del capítulo 4, del Apéndice A y B se tiene un análisis del proceso secuencial propuesto para inicio de la primera etapa del proyecto, del cual se propone un proceso modular. La principal ventaja del sistema modular está en los tiempos de operación. De la Figura 18 se aprecia el proceso de puesta en marcha, a continuación, se ponen sus tiempos de operación como comparación al tiempo de operación de marcha, Figura 54.

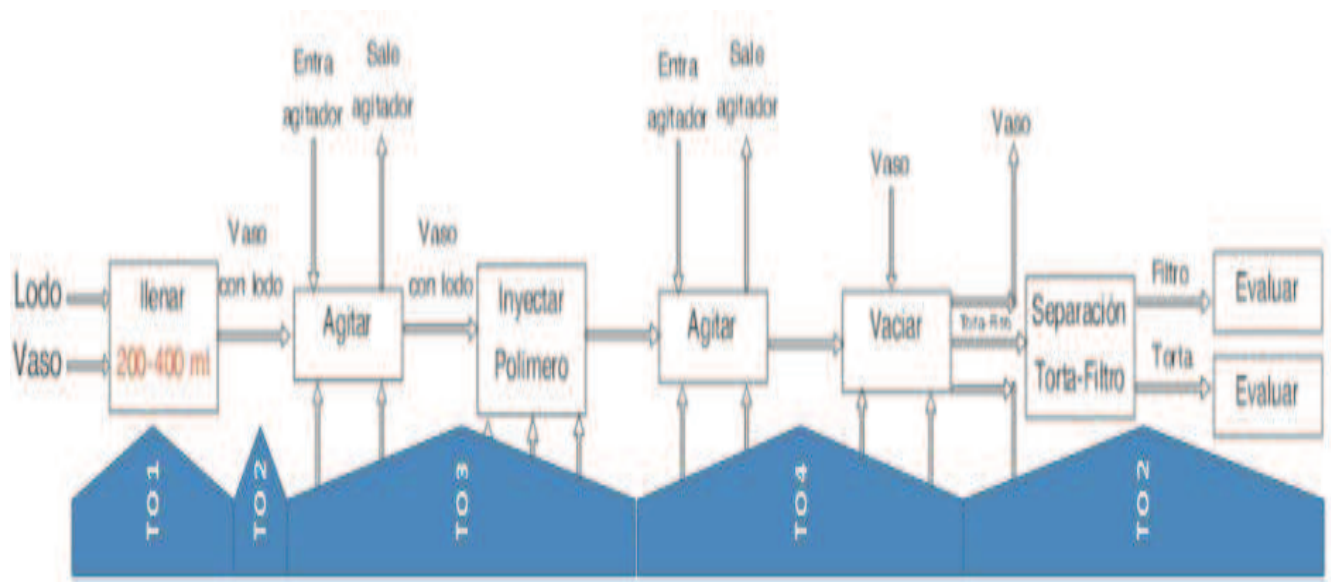


Figura 54. Comparación del proceso secuencial con los tiempos de operación

4.1.5 Cartas de función

Como herramienta de descomposición, las cartas de función secuencial SFC describen gráficamente el comportamiento secuencial de un programa de control, permitiendo mostrar las características físicas y lógicas de todo el proceso. Del Anexo 1 se establece las bases teóricas del uso de Grafcet con SFC, del cual para el desarrollo de este trabajo se usará el nivel 2 de abstracción para el Grafcet y nivel 3 para las funciones SFC, pues la implementación de la UC está definida para microcontroladores y el objetivo no es la implementación mediante dispositivos PLC.

Los FBC del sistema físico representan las SFC elementales de un proceso. De acuerdo a la clasificación de POUS, Tabla 2, las operaciones de ejecución representan funciones básicas que pueden ser usadas en todos los grupos de operación, mientras que los bloques funcionales representan a las funciones de módulos; finalmente el grupo de operación de planeación y de configuración representa al tipo programa.

Las variables de las funciones de los módulos, en forma general son de cuatro tipos: de operaciones de función, de estados de módulos, de datos de comunicación y de configuración de salidas, Tabla 28. Por convención se usarán las etiquetas w- eventos v- variable, d- dato y p- parámetro para asociarlas directamente con variables de programación.

Tabla 28. Tipos de variables en las funciones de módulo

Variables	Operaciones de función	Estados de módulos		Datos de comunicación		Configuración de salidas
Etiqueta	Eventos w	Sistema v0	Sensores v1	Dato entrada d_i	Dato salida d_o	Valores de pines I/O
Tipo	Operaciones de eventos y tareas	Estados de tareas	Estados de sensores	Variables: - tareas - parámetros - configuración	Estado: - sistema - sensores - actuadores - tareas	Pines entrada: - sensores Pines salida: - sistema - actuadores

Para cada sistema se establecerán las variables de las funciones con base en la Tabla 28. Se toma en cuenta a partir de los esquemáticos y de la jerarquía de componentes para relacionar la parte física, Figuras 36 a 42 y Tablas 17 a 19, y 21; los estados de inicio y las funciones de operación para relacionar la parte lógica del proceso, Tablas 22 y 25. La integración de estos conceptos junto con las variables de funciones de SFC permite mantener el enfoque modular contemplando el proceso físico y lógico, Tablas 29 a 32.

Tabla 29. Variables y funciones de módulo 1

Sistema: Función	Entradas/salidas físicas	Parámetros de operación	Variables, estados	Variables, datos de comunicación
Carrusel: Mover	Encóder Actuador M1	- Dirección - Tipo de giro	Estado sensores Estado de tarea	- Tareas y estados de sistema - Configuración de módulo - Estados lógicos
Orientación contenedor: Orientar B y C	Posición HB,HC Posición LB,LC Actuador M2, M3	- Estado proceso - Dirección		
Posición batidora: Mover	Posición HD Posición LD Actuador M4	- Dirección batidora - Estado de proceso		

Tabla 30. Variables y funciones de módulo 2

Sistema: Función	Entradas/salidas físicas	Parámetros de operación	Variables, estados	Variables, datos de comunicación
Batidora: • Mezclar • Batir	Encóder Actuador M5	- Velocidad y tiempo de mezclado - Estado proceso	Estado sensor Estado de tarea	- Tareas y estados de sistema - Configuración de módulo - Estados lógicos

Tabla 31. Variables y funciones de módulo 3

Sistema: Función	Entradas/salidas físicas	Parámetros de operación	Variables, estados	Variables, datos de comunicación
Dosificación efluente: Dosificar	Sensor de volumen Actuador M6	- Estado proceso - Cantidad de dosificación	Estado sensor Estado de tarea	- Tarea y estados de sistema - Configuración de módulo - Estados lógicos

Tabla 32. Variables y funciones de módulo 4

Sistema: Función	Entradas/salidas físicas	Parámetros de operación	Variables, estados	Variables, datos de comunicación
Dosificación floculante: • Vías • Dosificar	Sensor de flujo Actuador M7-M10	- Dirección de flujo - Cantidad dosificación - Tipo floculante	Estado sensores Estado de tarea Estados de vías	- Tareas y estados de sistema - Configuración de módulo - Estados lógicos

Para el siguiente desarrollo de las SFC de las operaciones de tipo ejecutivo, se analiza las funciones de los sistemas con sus variables, estados y datos en el proceso, sin tomar en cuenta las operaciones de tipo planeación y configuración, variables de operación de función w, pues éstas representan el GRTAFCET. Se toman en cuenta los siguientes puntos que no alteran ni cambian las propiedades de los SFC, solo permite organizarlos para su posterior implementación en microcontroladores:

- Cada SFC de los módulos se definen con nivel 3 de detalle para tener definidos todos los estados del proceso y permitir establecer su jerarquía en Grafcet.
- Durante cada etapa todas las salidas lógicas mantienen sus estados hasta que termine, sin necesidad de cambiar el estado en la siguiente etapa. Los bloques de enclavamiento, se siguen en orden de pila (última enclavada, primera en ser finalizada).
- Las variables modificadas en cada etapa mantienen su valor durante todo el proceso.
- Las entradas, variables o estados que dependan de sensores cuyos valores sean determinados por funciones específicas y no sea de interés su desarrollo a detalle, se representan con una “@”, como velocidad, volumen, flujo, etc.
- Los tiempos de operación y los estados iniciales de cada tarea determinan las condiciones de inicio y fin de laca SFC, de manera que se mantenga el proceso.

En las Tablas 33 a 39 se establecen las variables y estados, y en las Figuras 55 a 61 los SFC de los componentes del sistema mecánico respectivamente.

Tabla 33. SFC sistema 1a de módulo 1

E1	Inicio de SFC 1, mover posición de carrusel para contenedores, con actuador asociado m1 y sus encoders 1 de giro completo y 2 de medio giro		
Etapas	Estados	Transición siguiente	Descripción
X1.1	p_o[0] - Pin tarea de módulo p_o[2] - Pin dirección de 1a d_i[0] - Dato de entrada dirección d_o[5] - Dato de salida tarea sistema 1a	1, 5 - d_i[1] : Tipo de tarea de módulo estructura	Indica a maestro que inicia tarea, estado en transición y enclava dirección de sistema 1a. Determina giro por elección
X1.2 X1.5	p_o[3] - Pin PWM m1 v0[0] - Estado tarea sistema 1a	2,6 - p_i[0] : Sensor encóder principal	Inicia movimiento con enclavamiento de PWM de motor 1, para giro completo 1.2 o medio 1.5
X1.3 X1.6	p_i[0] - Sensor encóder principal p_i[1] - Sensor encóder auxiliar v1[0] - Estado encóder principal v1[1] - Estado encóder auxiliar d_o[3,4] - Dato de salida estado encoders 1a	3 - v1[0] : Estado encóder, giro completo 7 - v1[1] : Estado encóder, giro Medio	Transición y giro: Completo, 1.3 Medio, 1.6 Actualiza estado de encoders
X1.4 X1.7	p_o[3] - Pin PWM m1 v0[0] - Estado tarea sistema 1a	4,8 - v0[0] : Estado tarea sistema	Cambio de posición completado, desenchava PWM de m1
S1	p_o[0] - Pin tarea de módulo p_o[2] - Pin dirección de 1a v0[0] - Estado tarea sistema 1a d_o[5] - Dato de salida tarea sistema 1a		Fin de SFC 1, mover posición. Se actualiza estado de sistema 1a, se desenchava dirección y pin de tarea de módulo

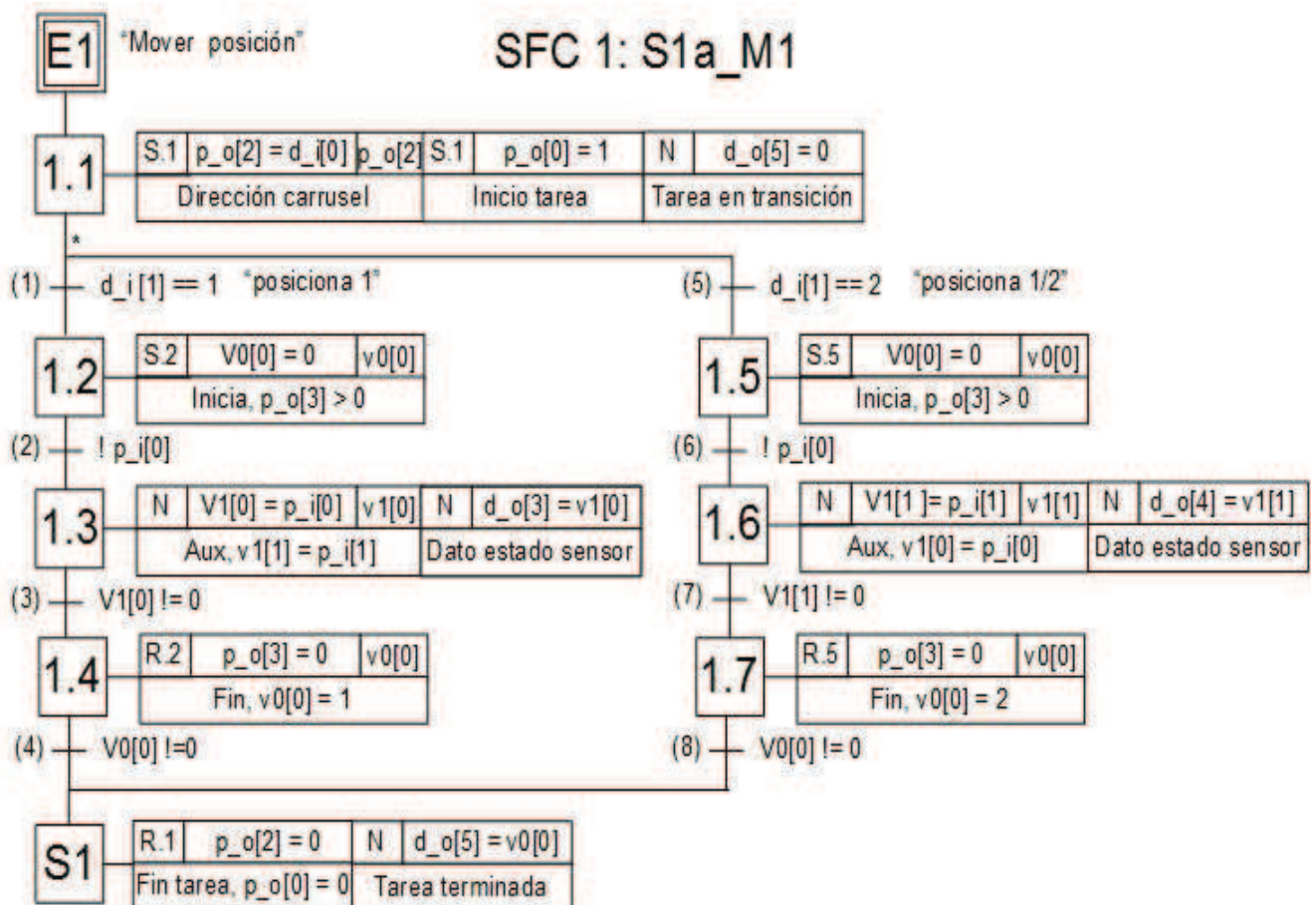


Figura 55. SFC 1, sistema carrusel de módulo 1

Tabla 34. SFC sistema 1b y 1c de módulo 1

E2	Inicio SFC 2, orientación para sistemas 1b, 1c; con actuadores asociados m2 y m3 con sensores de posición b1 L, b2 H, y c1 mezclado, c2 vaciado. Funciones de mover giro completo y medio giro		
Etapas	Estados	Transición siguiente	Descripción
X2.1	p_o[0] – Pin de tarea de módulo d_o[10] – Dato de salida tarea sistema 1b, 1c	1,10 - d_i[1]: Tipo de tarea de módulo estructura	Indica a maestro que inicia tarea, se determina el sistema a orientar por elección
X2.2	v0[1] – Estado tarea sistema 1b	2,5- p_i[2]: Sensor posición L, 1b	Tarea de orientación sistema 1b, se identifica posición y se pone en transición
X2.3 X2.5	p_o[4] – Pin de dirección 1b v1[2] – estado sensor L, 1b v1[3] – estado sensor H, 1b	3 - v1[2]: Estado sensor L, 1b 6 - v1[3]: Estado sensor H, 1b	Cambio de dirección de orientación a partir de identificar la actual
X2.4 X2.6	p_o[5] – Pin PWM m2 p_i[2] – Sensor posición L, 1b p_i[3] – Sensor posición H, 1b v1[2] – Estado sensor L, 1b v1[3] – Estado sensor H, 1b d_o[6] – Dato de salida sensor 1 d_o[7] – Dato de salida sensor 2	4 - v1[2]: Estado sensor L, 1b 7 - v1[3] Estado sensor H, 1b	Movimiento hasta concretar cambio de orientación. Se almacena estado de sensores
X2.7	v0[1] – Estado tarea sistema 1b p_o[4] – Pin dirección 1b	8 – v0[1]: Estado tarea sistema 1b	Orientación completada en 1b, desenchava dirección
X2.9	v0[2] – Estado tarea sistema 1c	11, 15 – d_i[1]: Tipo de tarea módulo estructura	Tarea de orientación sistema 1c, se determina orientación por elección y se pone en transición
X2.10 X2.13	p_o[6] – Pin dirección 1c p_o[7] – Pin PWM m3 p_i[4] – Sensor posición mezcla p_i[5] – Sensor posición vaciado v1[4] – Estado sensor mezcla v1[5] – Estado sensor vaciado	12 – v1[4]: Estado sensor posición mezclado 16 – v1[5]: Estado sensor posición vaciado	Inicia movimiento: 10- Hacia vaciado, espera cambio de estado en mezcla, v1[4] 13- Hacia mezclado, enclava dirección 1c, espera cambio de estado en vaciado, v1[5]
X2.11	p_i[5] – Sensor posición vaciado v1[5] – Estado sensor vaciado d_o[9] – Dato de salida estado sensor vaciado	13 – v1[5]: Estado sensor posición vaciado	Espera de posicionamiento en vaciado, almacenamiento de dato y estado
X2.12 X2.15	p_o[7] – pin PWM m3 p_o[6] – Pin dirección 1c v0[2] – Estado tarea sistema 1c	14, 18 – v0[2]: Estado tarea sistema 1c	Orientación completada en 1c, desenchava dirección y PWM
X2.14	p_i[4] – Sensor posición mezclado v1[4] – Estado sensor mezclado d_o[8] – Dato de salida estado sensor mezclado	17 – v1[4]: Estado sensor posición mezclado	Espera posicionamiento en mezclado, almacenamiento de dato y estado
X2.8 X2.16	d_o[10] – Dato de salida tarea sistema 1b,1c v0[2] – Estado tarea sistema 1c	9, 19 – d_o[10]: Dato de salida tarea sistema 1b,1c	Actualiza dato de estado tarea, identificador de tipo de orientación completada
S2	p_o[0] – Pin de tarea de módulo		Fin de SFC 2, orientación. Desenchava tarea de módulo

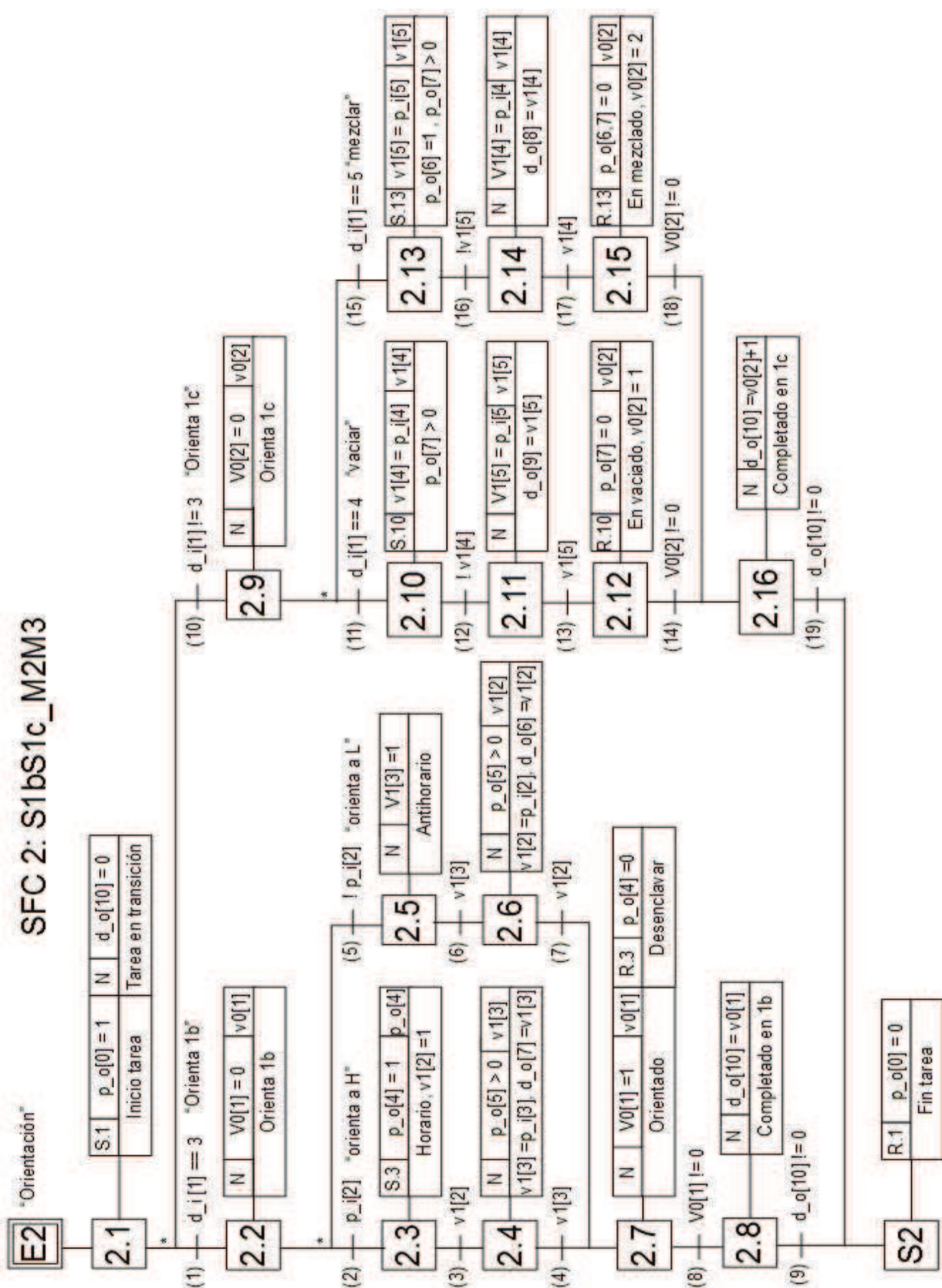


Figura 56. SFC 2, sistema orientación de módulo 1

Tabla 35. SFC sistema 1d de módulo 1

E3	Inicio de SFC 3, mover posición de batidora, con actuador asociado m4 y sus sensores de posición d1 alto y d2 bajo. Funciones de subir y bajar batidora		
Etapa	Estados	Transición siguiente	Descripción
X3.1	p_o[0] - Pin tarea de módulo d_o[14] - Dato de salida tarea sistema 1d	1, 5 - d_i[1] : Tipo de tarea de módulo estructura	Indica a maestro inicio de tarea, estado sistema en transición, determina cambio de posición por elección
X3.2 X3.5	p_o[9] - Pin PWM m4 p_o[8] - Pin dirección m4 v0[3] - Estado tarea sistema 1d	2 - p_i[6] : Sensor posición arriba 6 - p_i[7] : Sensor posición abajo	Inicia movimiento, enclavamiento de PWM para m4 con movimiento de bajada en 3.2, de subida en 3.5 con enclavamiento de dirección
X3.3 X3.6	p_i[7] - Sensor posición alta p_i[6] - Sensor posición baja v1[7] - Estado sensor posición B v1[6] - Estado sensor posición A d_o[12,13] - Dato de salida estado sensores 1d	3 - v1[7] : Estado sensor, posición baja 7 - v1[6] : Estado sensor, posición alta	Transición y posicionamiento: Alto a bajo, 3.3 Bajo a alto, 3.6 Actualiza estado de sensores
X3.4 X3.7	p_o[9] - Pin PWM m4 p_o[8] - Pin dirección de 1d v0[3] - Estado tarea sistema 1d	4,8 - v0[3] : Estado tarea sistema	Cambio de posición completado, desenchava PWM de m4 y dirección
S3	p_o[0] - Pin tarea de módulo v0[3] - Estado tarea sistema 1d d_o[14] - Dato de salida tarea sistema 1d		Fin de SFC 3, mover posición. Se actualiza estado de sistema 1d , se desenchava pin de tarea de módulo

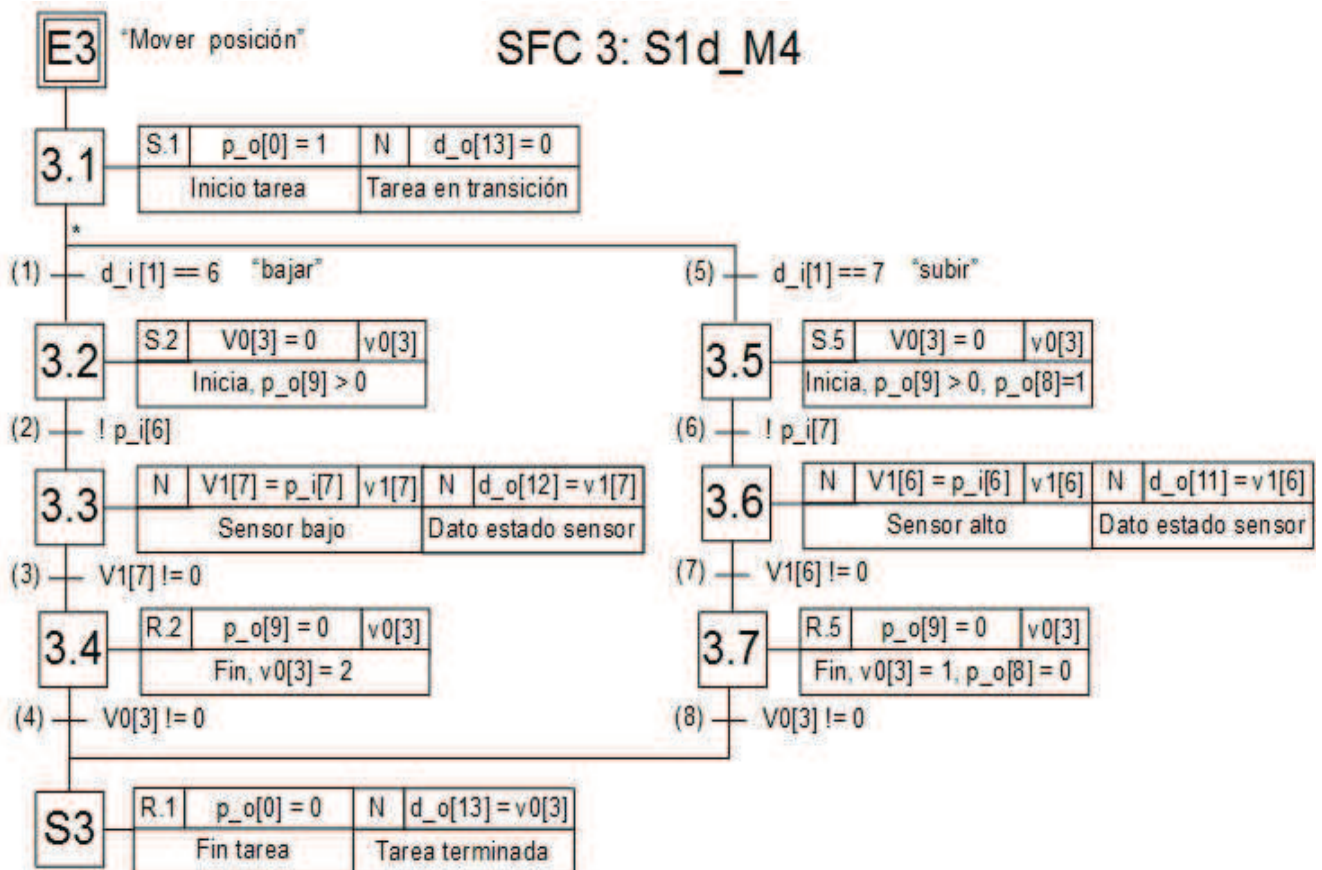
*Figura 57. SFC 3, sistema posición batidora de módulo 1*

Tabla 36. SFC sistema 2a de módulo 2

E4	Inicio de SFC 4, mezclado, con actuador asociado m4 y encóder 1 para velocidad. Funciones de revolver y batir		
Etapas	Estados	Transición siguiente	Descripción
X4.1	p_o[0] – Pin tarea de módulo d_o[4] – Dato de salida tarea sistema	1, 5 – d_i[0] : Tipo de tarea para módulo mezclado	Indica a maestro inicio de tarea, estado de sistema en transición; tipo de tarea por elección
X4.2	p_o[2] – Pin PWM p_i[0] – Encóder v1[0] – Estado encóder v0[0] – Estado tarea sistema d_o[3] – Dato de salida encóder	2 – v1[0] : Estado encóder , asociado a velocidad	Tarea revolver en transición, enclava actuador m1 en función del dato de entrada de velocidad y actualiza dato encóder
X4.3	Estado de espera de transición	3 – d_i[0]: Dato entrada de tipo de tarea para módulo mezclado	Espera cambio de dato entrada de tarea por maestro
X4.4	p_o[2] – Pin PWM v0[0] – Estado tarea sistema	4 – v0[0]: Estado tarea sistema	Desenclava actuador m1, espera y actualiza estado tarea revolver hasta terminarla por maestro
X4.5	p_o[2] – Pin PWM p_i[0] – Encóder v1[0] – Estado encóder v0[0] – Estado tarea sistema d_o[3] – Dato salida estado encóder d_i[2] – Dato entrada tiempo Variables locales de control	6 – V1[0]: Estado encóder, asociado a velocidad	Tarea batir en transición. Enclava actuador m1 en función de dato entrada de velocidad, actualiza estados de sistema y sensor. Inicializa variables de control de velocidad
X4.6	p_o[2] – Pin PWM m1 p_i[0] – Encóder v1[0] – Estado encóder d_o[3] – Dato salida estado encóder Variables locales de control	10 – v1[0]: Estado encóder, asociado a velocidad	Actualiza el valor de PWM para m1 en función de la ganancia uT, el estado y el dato de encóder por velocidad. Espera estado velocidad a 0
X4.7	Variables locales de control	7 – eT: Variable control, error de control de estado velocidad	Calculo de la ganancia uT por retro de estado
X4.8	Variables locales de control	8,9 – tA: Variable control, tiempo	Actualización de variables locales de control
X4.9	p_o[2] – Pin PWM p_i[0] – Encóder v1[0] – Estado encóder	10 – v1[0]: Estado encóder, asociado a velocidad	Espera estado velocidad a 0
X4.10	v0[0] – Estado tarea sistema d_o[3] – Dato salida estado encóder	11 – v0[0]: Estado tarea de sistema	Fin de tarea batir, se actualiza estado de tarea
S4	p_o[0] – Pin tarea módulo v0[0] – Estado tarea sistema d_o[4] – Dato salida tarea sistema		Fin de SFC 4, mezclado. Se desenclava estado tarea de módulo y actualiza dato de tarea.

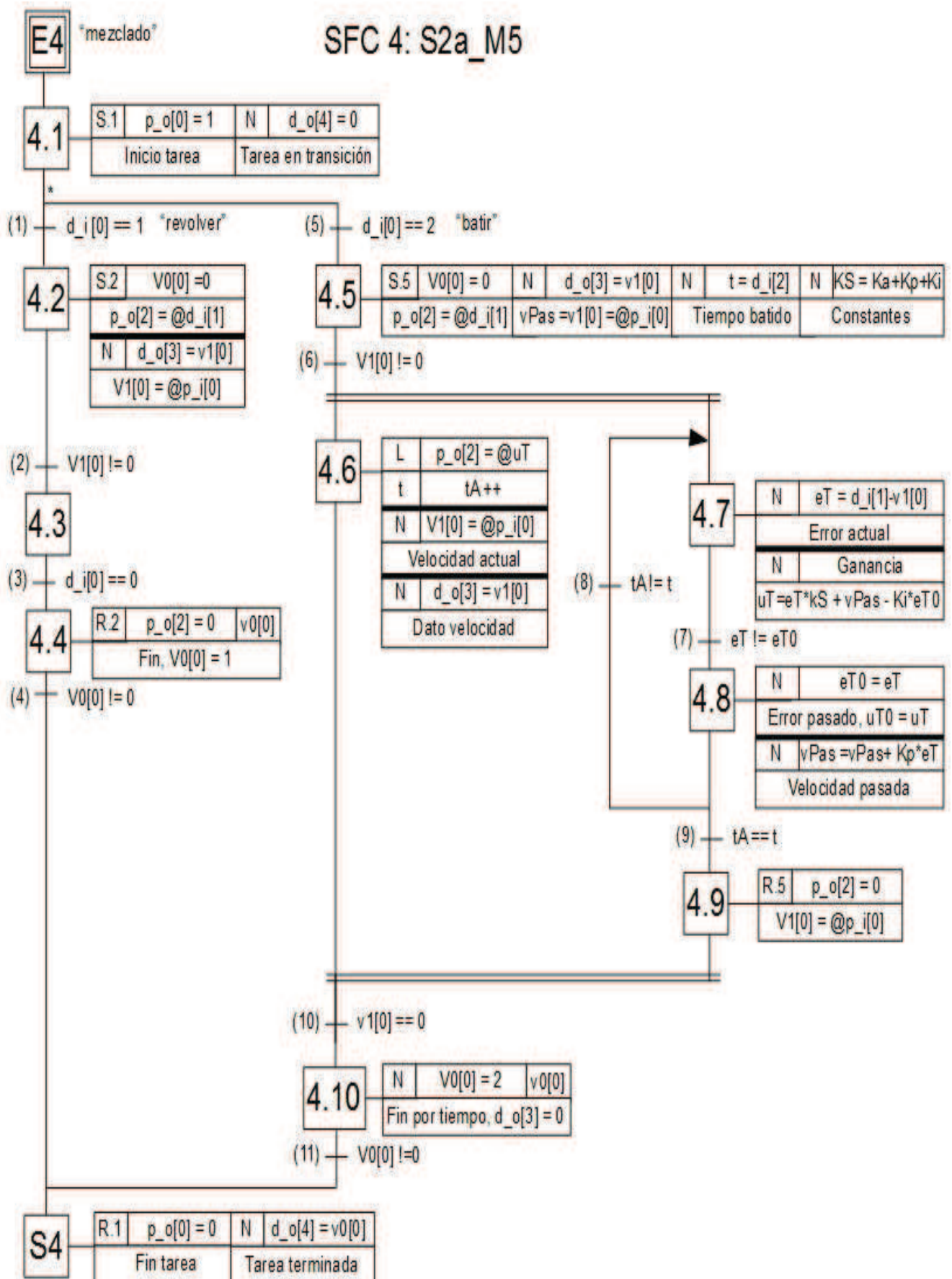
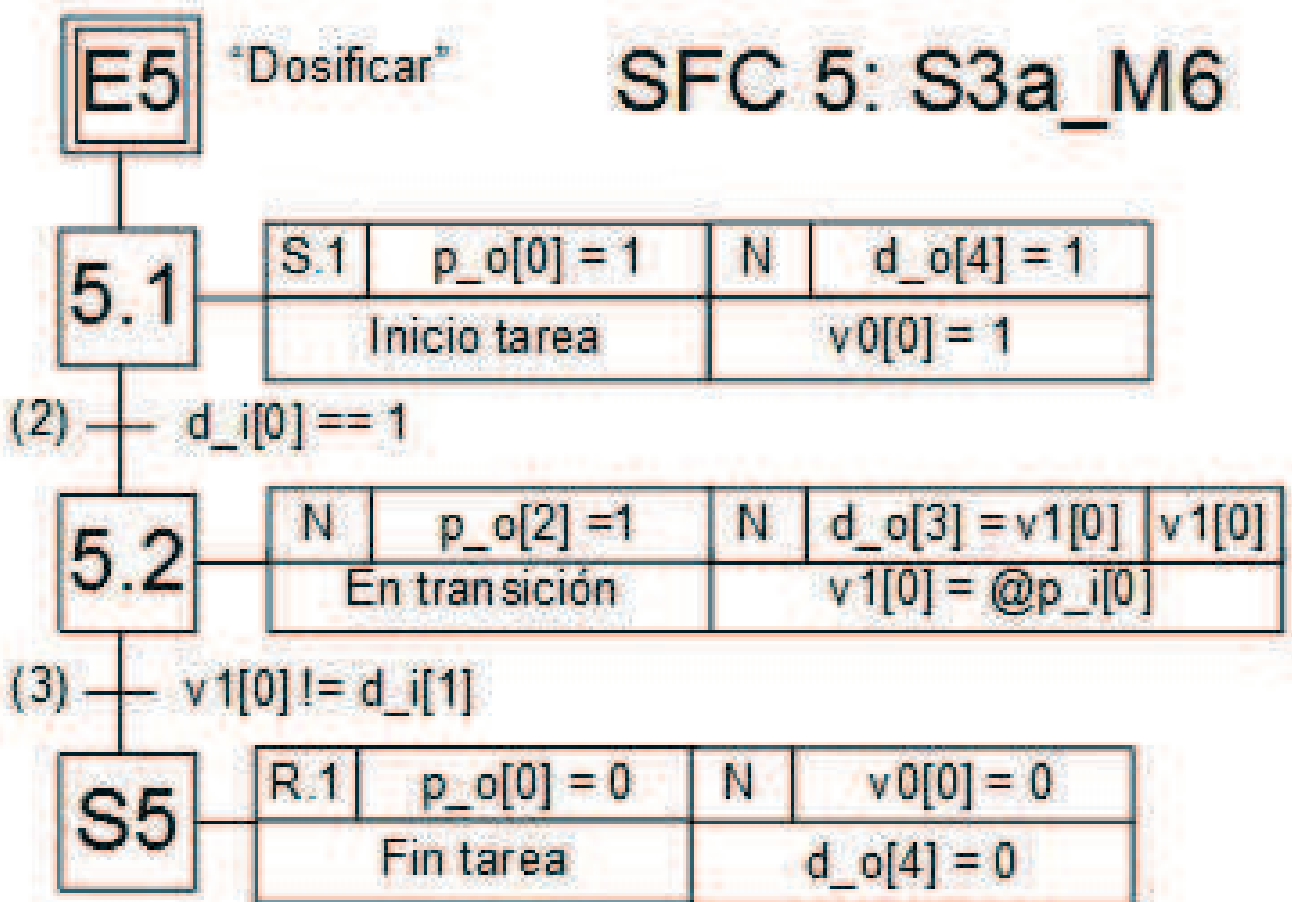


Figura 58. SFC 4, sistema mezclado de módulo 2

Tabla 37. SFC sistema 3a de módulo 3

E5	Inicio de SFC 5, dosificado de efluente con actuador asociado m6, y sensor de volumen liquido ml. Función de dosificar efluente		
Etapa	Estados	Transición siguiente	Descripción
X5.1	p_o[0] – Pin tarea de módulo v0[0] – Estado tarea sistema d_o[4] – Dato de salida tarea sistema	1 – d_i[0] : Tipo de tarea de módulo lodo	Indica a maestro que inicia tarea, dato de estado en proceso de llenado y verifica tarea módulo
X5.2	p_o[2] – Pin enable m6 p_i[0] – Sensor volumen, ml v1[0] – Estado tarea sistema d_o[3] – Dato salida de estado sensor	2 – v1[0] : Estado tarea sistema	Inicia dosificado, espera lectura de sensor , guarda estado hasta terminar tarea
S5	p_o[0] – Pin tarea de módulo v0[0] – Estado tarea sistema d_o[4] – Dato de salida tarea sistema		Fin de SFC 5 dosificado de efluente. Desenclava pin de tarea de módulo y actualiza estado

*Figura 59. SFC 5, sistema dosificado efluente de módulo 3***Tabla 38.** SFC sistema 3b1 de módulo 4

E6	Inicio de SFC 6, dosificar vías de floculante, con actuadores asociados del m7 al m10 con sus respectivos sensores de flujo, ml. Funciones de llenar y vaciar vías
----	--

Etapas	Estados	Transición siguiente	Descripción
X6.1	p_o[0] - Pin tarea de módulo d_i[0] - Tipo de tarea de módulo d_o[8] - Dato de salida tarea módulo d_o[7] - Dato de salida tarea vías, 3b1 Variables locales de vías: B - Contador pines de salida y estados sistema C - Contador estado sensores y pines entrada ref - Valor de referencia fluido en vías	1 - d_i[0] : Dato entrada tipo de tarea de módulo 4	Indica a maestro que inicia tarea, estados de tarea en transición e inicialización de parámetros. Se comprueba tarea
X6.2	v0 [B] - Estados tarea sistema 3b1. Variables locales de vías 3b1	2,6 - d_i[0] : Dato entrada tipo de tarea de módulo 4	Actualización estados de sistema, se determina tipo de tarea por elección
X6.3	p_o[3+B] - Pin PWM de actuadores p_i[C] - Sensores de flujo v1[C] - Estado sensores de flujo Variables locales de vías 3b1	3 - v1[C] : Estado sensores de flujo, en función de contador	Inicio de tarea de llenado de vías, enclavamiento de actuadores y actualización de sensor de flujo
X6.4 X6.7	p_i[C] - Sensores de flujo v1[C] - Estado sensores de flujo Variables locales de vías 3b1	4,8 - v1[C] : Estado sensores de flujo, en función de contador	Espera finalización de vaciado y llenado de vías por referencia. Se actualiza valor estado de sensor de flujo
X6.5	p_o[3+B] - Pin PWM de actuadores p_i[C] - Sensores de flujo v1[C] - Estados sensores v0[B] - Estados tarea sistema 3b1 d_i[0] - Dato tipo de tarea de módulo Variables locales de vías 3b1	5 - v1[C]: Estado sensores de flujo, en función de contador	Finaliza tarea de llenado de vías, desenchava actuador y actualiza estado de sensor; espera a que se detenga el flujo
X6.6	p_o[3+B] - Pin PWM de actuadores p_o[2+B] - Pin dirección de actuadores p_i[C] - Sensores de flujo v1[C] - Estado sensores de flujo Variables locales de vías 3b1	7 - v1[C]: Estado sensores de flujo, en función de contador	Inicio de tarea de vaciado de vías, enclava actuador y dirección; actualiza estado sensor de flujo
X6.8	p_o[3+B] - Pin PWM de actuadores p_o[2+B] - Pin dirección de actuadores p_i[C] - Sensores de flujo v1[C] - Estado sensores de flujo v0[B] - Estado tarea sistema 3b1 d_i[0] - Dato tipo de tarea de módulo Variables locales de vías 3b1	9 - v1[C]: Estado sensores de flujo, en función de contador	Finaliza tarea de vaciado de vías, desenchava actuador y dirección; actualiza estado de sensor y espera a que se detenga el flujo
X6.9	Variables locales de vías 3b1	10,11 - C: Contador de referencia para bomba actuada	Incrementa contadores de referencia B y C para estados y sensores. C como auxiliar de iteraciones de bombas
S6	p_o[0] - Pin tarea de módulo d_o[7] - Dato de salida tarea sistema 1a d_o[8] - Dato de salida tarea sistema 1a		Fin de SFC 6, mover posición. Se actualiza estado de datos de salida de tarea y vías de dosificación. Desenchava pin de tarea de módulo

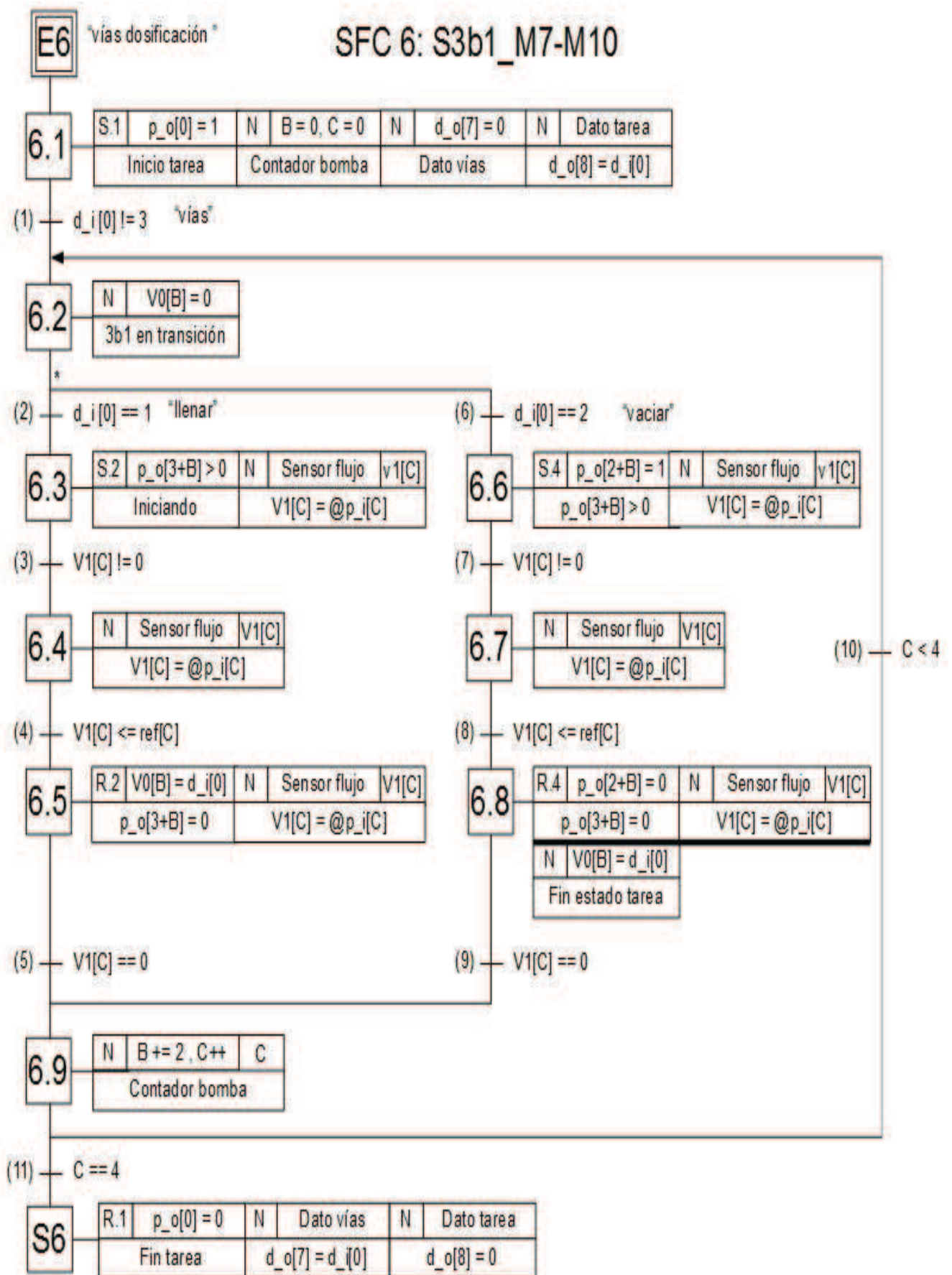


Figura 60. SFC 6, sistema dosificado vías de módulo 4

Tabla 39. SFC sistema 3b2 de módulo 4

E7	Inicio de SFC 7, dosificado de floculante, con actuadores asociados m7 a m10, con respectivos sensores de flujo		
Etapa	Estados	Transición siguiente	Descripción
X7.1	p_o[0] – Pin tarea de módulo 4 d_o[8] – Dato de salida tarea sistema Variables locales del sistema 3b2: B – Bomba, auxiliar en estados y entradas C – Cantidad a dosificar en bomba Id – Estado bomba, auxiliar en PWM Constantes locales del sistema 3b2: máx. – Cota superior de PWM min. – Cota inferior de PWM E – Error de dosificación	1 – d_i[1] : Tipo de bomba para dosificación	Indica a maestro que inicia tarea, actualiza dato de estado de tarea, verifica selección de bomba para dosificación
X7.2	v0[id] – Estado tarea sistema d_i[1] – Tipo de bomba para dosificación d_i[B+1] – cantidad de dosificación para cada bomba Variables locales del sistema	2 – C : Cantidad a dosificar en bomba	Actualiza valores para las variables de trabajo locales, verifica entrada de cantidad a dosificar
X7.3	p_o[id+2] – Pin PWM de bomba, 32b p_i[B-1] – Sensores de flujo, ml v1[B-1] – Estado sensor volumen d_o[B+2] – Dato de salida estado sensor Constante local de sistema 3b2	3 – v1[B-1] : Estado sensor de flujo	Dosificado de floculante al 50% con PWM máximo, actualización de estados de dosificación en función de sensor y datos de salida
X7.4	p_o[id+2] – Pin PWM de bomba, 32b p_i[B-1] – Sensores de flujo, ml v1[B-1] – Estado sensor volumen d_o[B+2] – Dato de salida estado sensor Variables locales de sistema 3b2 Constantes locales de sistema 3b2	4 – v1[B-1] : Estado sensor de flujo	Dosificación de floculante con PWM variable en función de dosificación, actualización de estados y datos de salida
S7	p_o[0] – Pin tarea de módulo 4 v0[id] – Estado tarea sistema d_o[8] – Dato de salida tarea sistema		Fin de SFC 7 dosificado. Desenclava pin de tarea de módulo

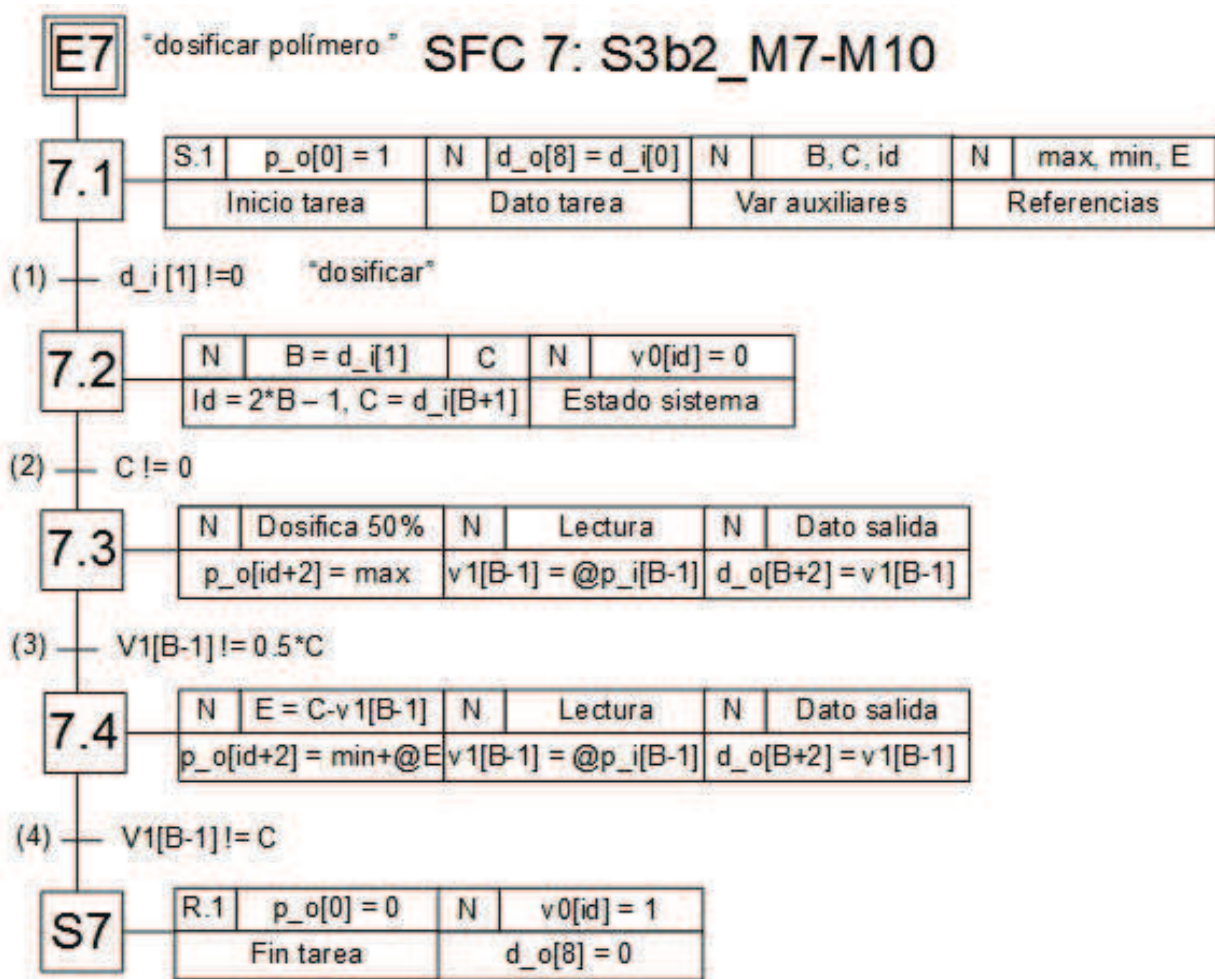


Figura 61. SFC 7, sistema dosificado floculante de módulo 4

4.1.6 GRAFCET de operaciones

La construcción de un sistema automático requiere establecer las relaciones causa efecto de los eventos del proceso y la actuación de la unidad de control, UC, a partir de un orden secuencial y jerárquico en el sentido del autómatas programable, esta relación puede definirse dependiendo del diseñador. Para este trabajo se seguirá la estructura deliberativa para el establecimiento de las funciones de operación de la unidad de control mientras que las funciones de operación del proceso siguen la jerarquía de componentes.

Es importante hacer la distinción entre operaciones de la UC y del proceso debido a que la primera se determina por las características del autómatas (tipo de estructura de comunicación y operaciones de planeación) mientras que el proceso define las funciones de ejecución. Como el objetivo no es implementar el sistema en un dispositivo PLC existe esta distinción, pues la metodología de Grafcet para automatización de procesos se enfoca en su implementación; además estas distinciones no define al sistema modular, si no a sus funciones de operación.

Todos los sistemas y sus funciones mantienen una relación jerárquica en el proceso por las condiciones de operación, lo que permite representarlas por un diagrama de Grafcet para darle una estructura organizada, relacionar las operaciones físicas como lógicas del proceso, de FBC a SFC, como funciones (parte ejecutiva de un autómatas programable).

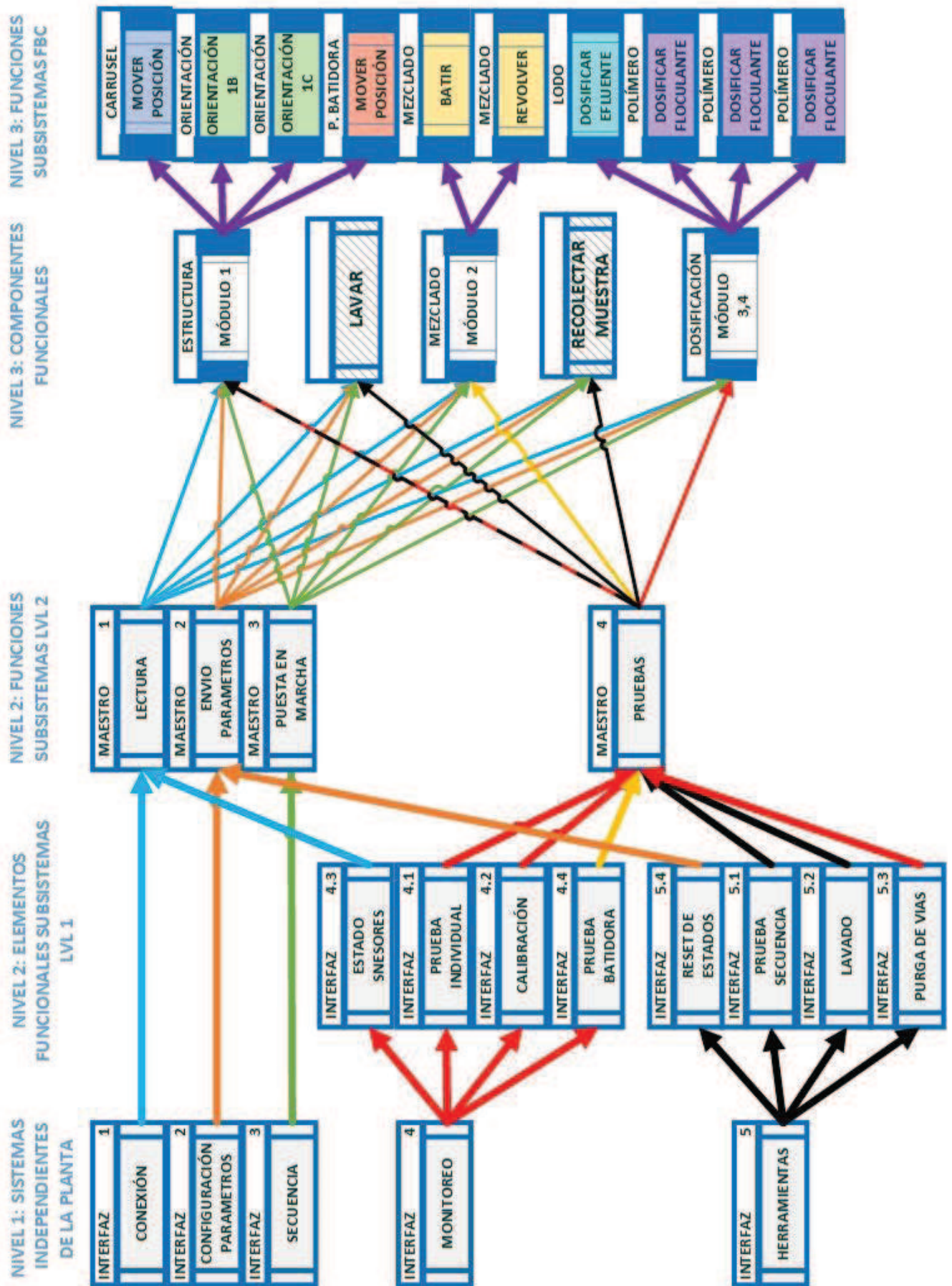


Figura 62. Jerarquía de funciones de operación

Es importante señalar que para este trabajo existieron condiciones de diseño por el sistema mecánico que desarrolló el grupo de trabajo de mecánica del proyecto, cap. 3.1.1.; si para el análisis del proceso no existiese dicho modelo se agregaría la etapa de diseño mecánico antes de la arquitectura del proceso para definir los sistemas físicos considerando los actuadores y sensores candidatos para la automatización del proceso. En cuanto a las jerarquías de funciones de operación, en comparación con el análisis de componentes de FBC existe una jerarquía por tipo de POUS para las operaciones de un proceso en tres niveles, Tabla 2; el establecimiento de esta jerarquía se basa en las especificaciones del diseño del sistema físico que representan las condiciones de las funciones de operación del proceso en la UC.

A partir de las Tablas 27 y 26, que describen los procesos de operación a nivel de planeación y configuración por tiempos de operación, y por las condiciones de las funciones descritas en el cap. 4.1.5 se establece en la Figura 62 el esquema representativo de la jerarquía de funciones de operación. Es representativo debido a que no todas las funciones de los grupos de operación de planeación y configuración pueden ser puestas en dicho modelo, pues aun cuando algunas funciones tienen un nivel tres de jerarquía por sus características se ejecutan durante todo el proceso.

Hasta este punto del diseño, sin pasar al GRAFCET, el análisis exhaustivo del proceso permite inferir de la Figura 62 los siguientes puntos:

- La unidad de control es modular en el nivel tres para los sistemas de FBC.
- La interfaz junto con el maestro sustituyen las operaciones del operador.
- La ejecución de operaciones en paralelo, multitarea, depende de las capacidades del maestro, y no de los módulos de los sistemas.
- Por el punto anterior, las operaciones del proceso que pueden ser representadas por Grafcet son las que son representadas por tiempos de operación.

De las funciones y grupos de operaciones, Tabla 26 y 27, se definen cuatro grupos de funciones que establecen tiempos de operación, para las funciones de monitoreo y configuración se representarán en bloques funcionales y dentro de la estructura de comunicación de la UC, por los puntos anteriores. Por otra parte se requiere definir la estructura de función de los módulos para ejecución y configuración de tareas. De esta forma se plantea un grupo de 6 Grafcet de tipo encapsulación para representar los tiempos de operación de la Tablas 21, 22 y 25. Las consideraciones para el establecimiento de los diagramas son los siguientes:

- Cada Grafcet sigue el nivel 2 de abstracción por representar operaciones del proceso
- El módulo maestro sigue la misma estructura del módulo esclavo en cuanto a ejecución de operaciones de proceso, siendo que para esclavo hace llamada a SFC y para maestro hace llamada a operaciones de UC
- Las variables de módulos mantendrán la forma general de la Tabla 28, se hará referencia en el comentario de cada etapa, aunque no a detalle como en los SFC; se retomaran en los bloques funcionales de la UC
- Para los macros de las cartas de función y de solicitud tarea se usa # para hacer referencia al módulo y en su descripción la referencia al SFC
- Las operaciones de lavado y recolección muestra no se agregan en el desarrollo de los Grafcet, pues únicamente se indicaron para los tiempos de operación pues a la fecha no se han desarrollado, noviembre 2015.
- Los encapsulamientos siguen la estructura de tiempo de operación: se carga y configura el inicio del TO, se ejecutan secuencialmente y se verifica por cada uno el estado del sistema.

Tabla 40. Variables y funciones de módulo maestro interfaz

Sistema: Función	Entradas/salidas físicas	Parámetros de operación	Variables, estados	Variables, datos de comunicación
Módulo maestro	Tarea Error Comunicación	- Tareas - Datos	- Estado sensores - Estado de tarea	- Tareas y estados de sistema - Configuración de módulo - Estados lógicos
Interfaz	Comunicación	- Estado proceso - Solicitud de operación		

En la Tabla 40 se muestran de forma general las variables para los grupos de operación del módulo maestro e interfaz a desarrollar. No hace falta describir a detalle estas variables debido a que para el Grafcet recurre a la abstracción de nivel dos, pero se retomaran para el desarrollo de los bloques funcionales. En los siguientes diagramas se representan los Grafcet encapsulados de las operaciones.

Módulo maestro/esclavo

Para la representación de las operaciones para ejecución de tarea en los módulos se definen dos macro etapas como en los SFC, por una parte, se establece el modelo general para tarea de cada módulo, donde se determina el tipo de tarea W del cual se espera cambio de estado de tarea, Figura 63. Por otra parte, se define una macro etapa que incluye a la anterior, la cual gestiona la ejecución de las solicitudes de tarea por finalización esperada o por error, además representa la unión entre las operaciones del proceso y de la UC, Figura 64.

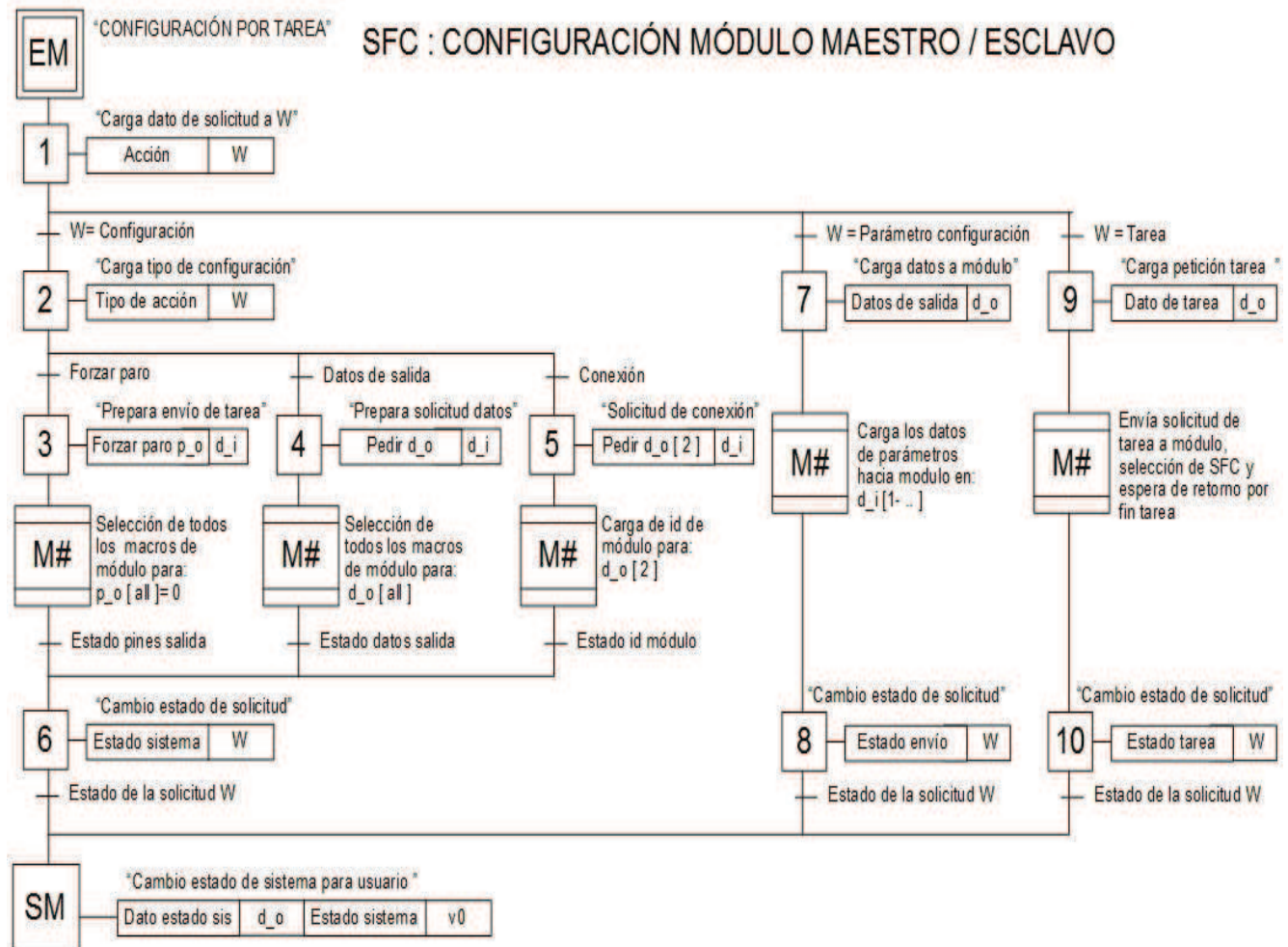


Figura 63. SFC de Módulo, configuración de ejecución de tarea

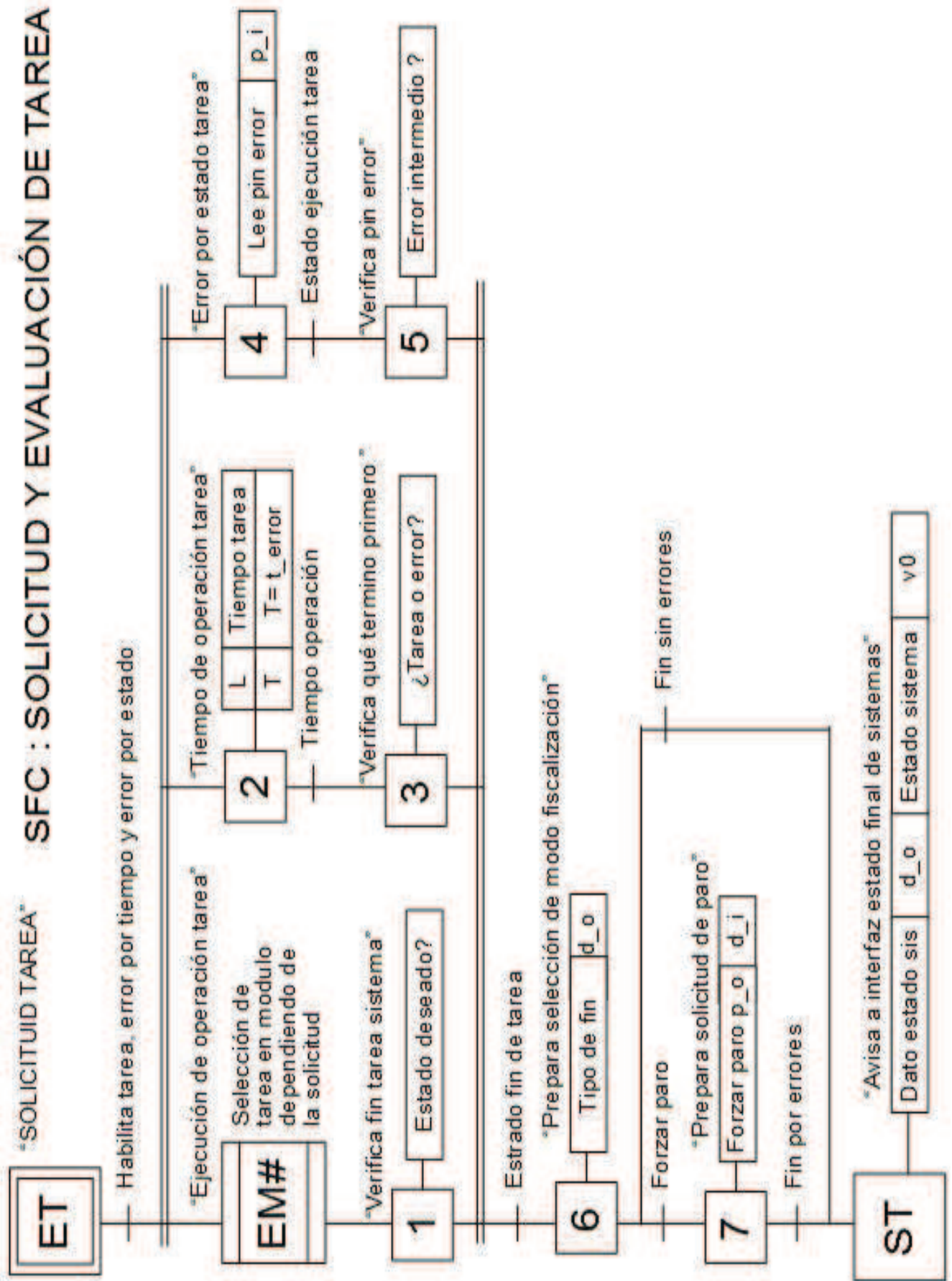


Figura 64. SFC de Módulo, solicitud de tarea

Preparación

Figura 68, la preparación usa la función de prueba de secuencia para módulo 1, sigue el diagrama de tiempo de operación de la Figura 47; genera una notificación a usuario.

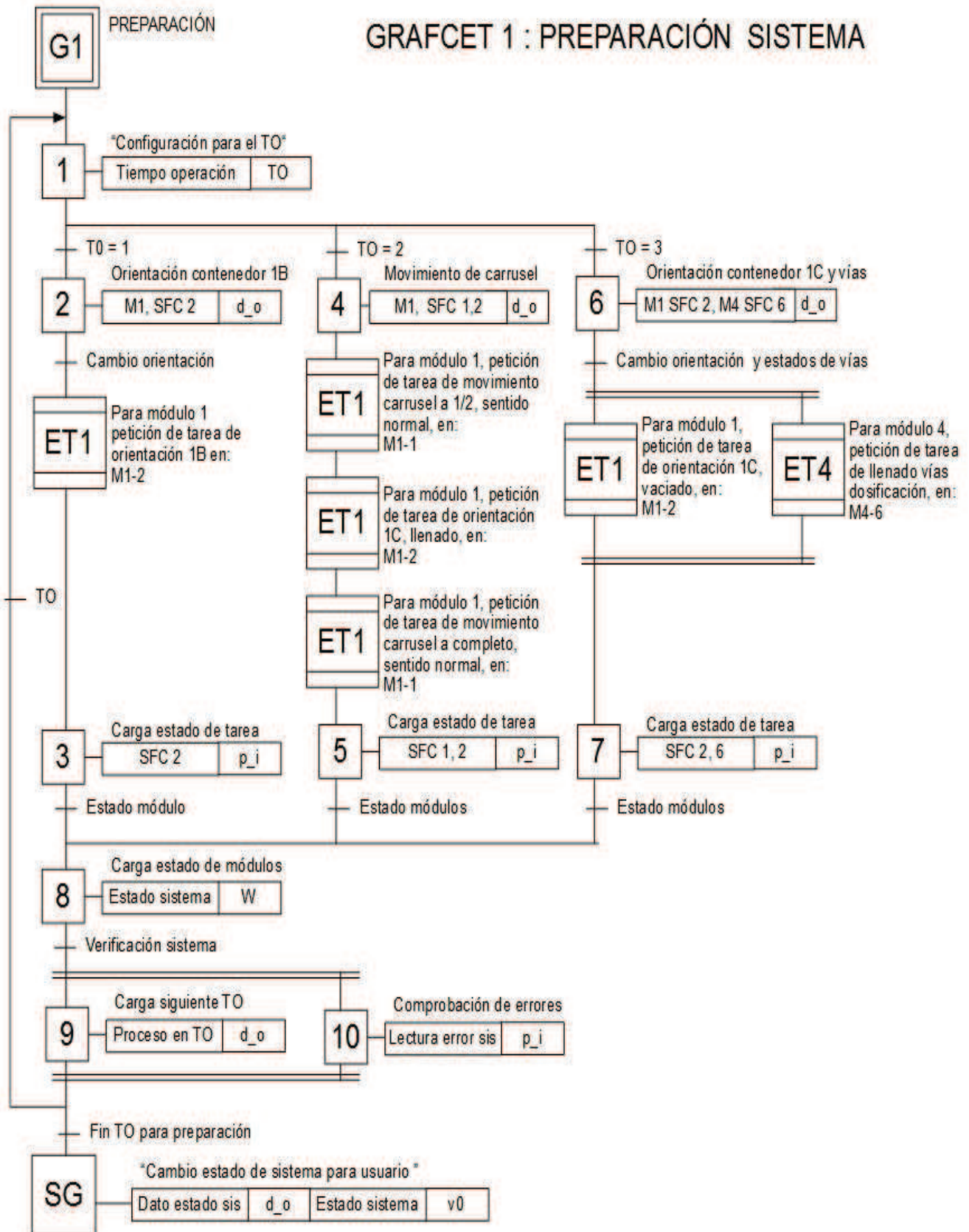


Figura 65. Grafcet encapsulado 1, Preparación sistema

Puesta en marcha

Figura 66, la puesta en marcha usa la función de secuencia para módulo 1, 2, 3 y 4, sigue el diagrama de tiempo de operación de la Figura 48; genera notificación y confirmación a usuario.

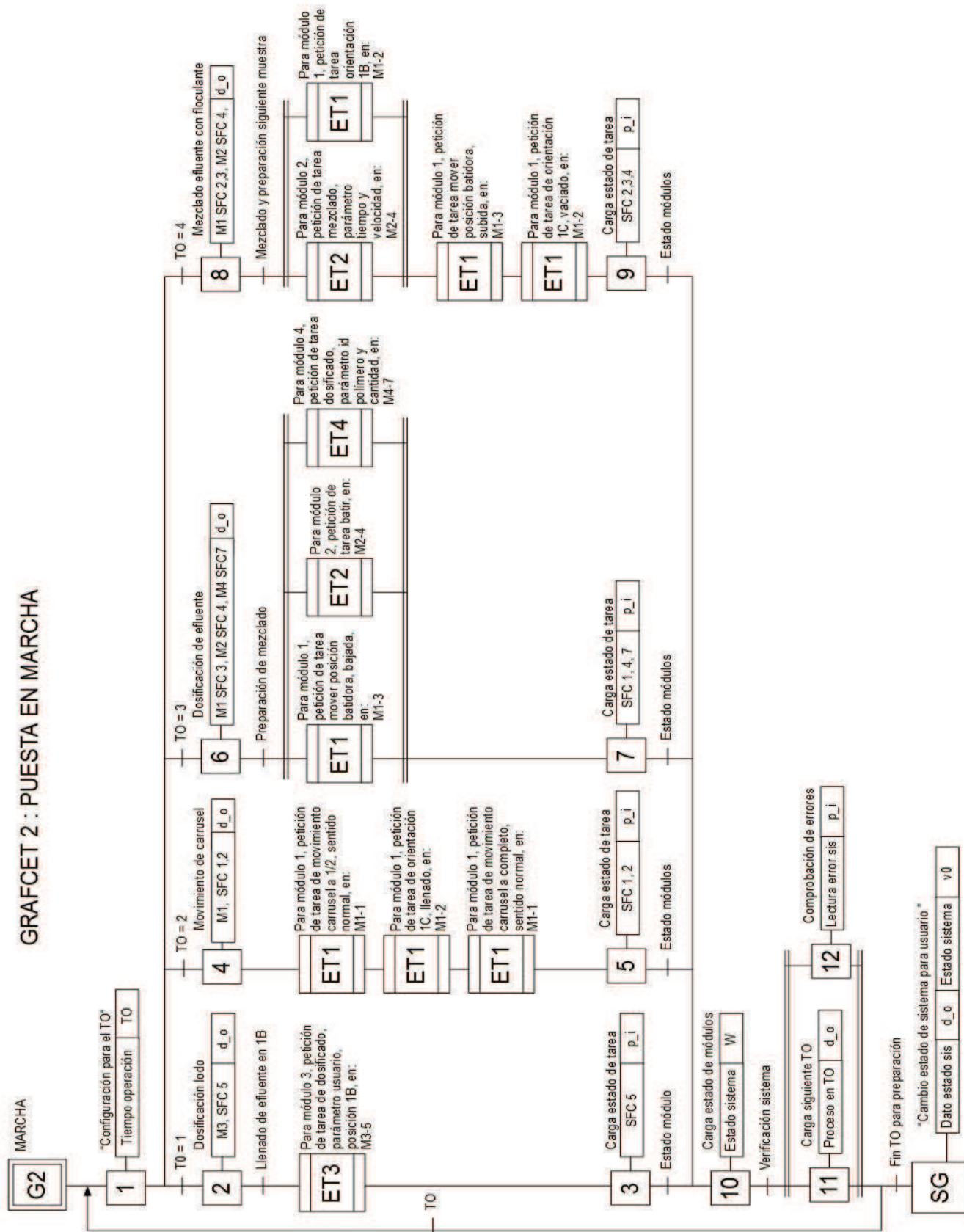


Figura 66. Grafcet encapsulado 2, Puesta en marcha

Fin

Figura 67, el fin de operación ocurre cuando se tiene éxito en la finalización esperada de operaciones, representa al tiempo de operación de la Figura 51, usa las funciones de Herramienta para módulo lavado y de estructura; genera notificación a usuario.

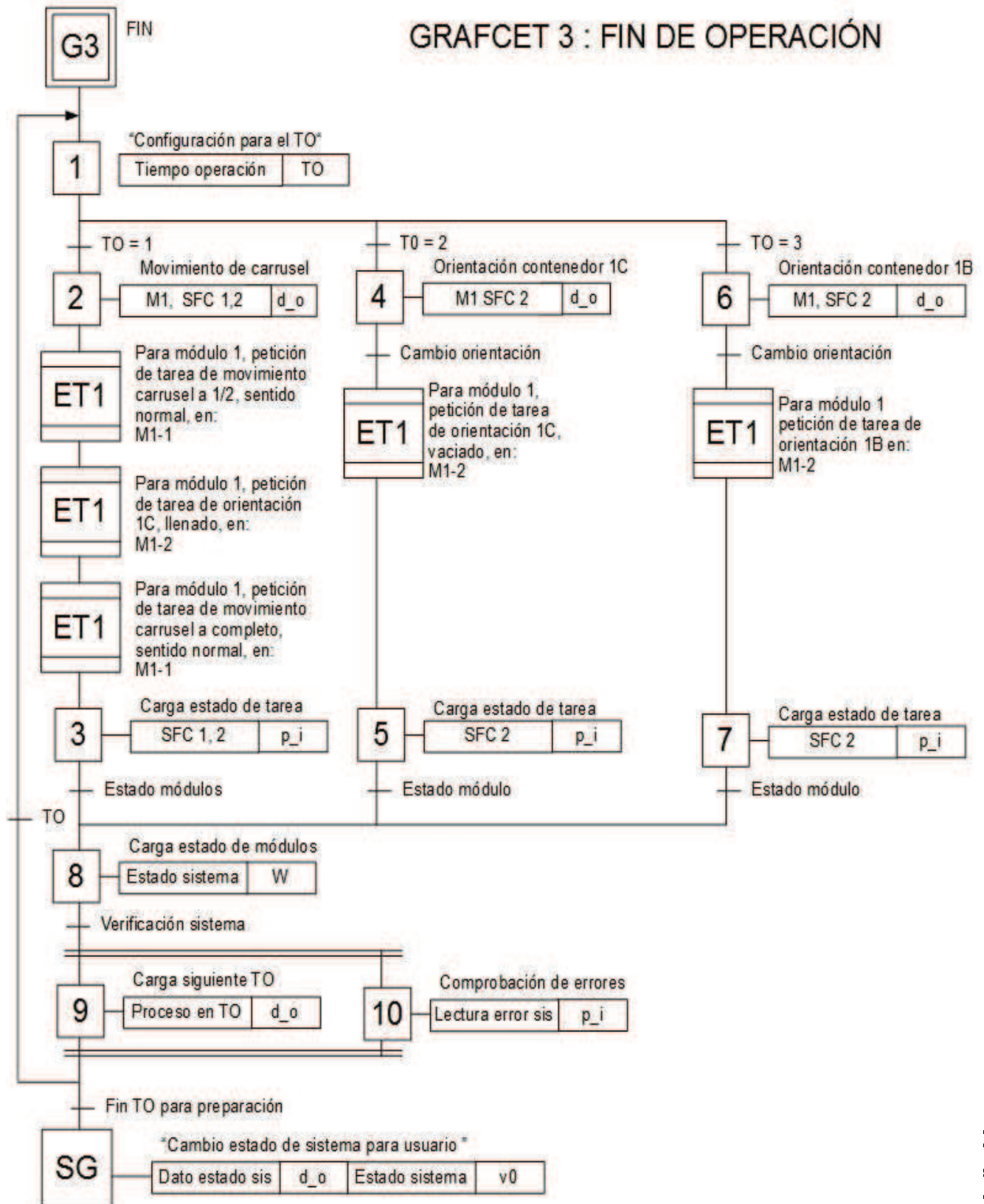


Figura 67. Grafcet encapsulado 3, Fin de operación

Fin por errores

Figura 68, fin por error en operación ocurre cuando en la ejecución de una operación se detecta algún tipo de error, forzando el paro de todo el sistema, sigue el tiempo de operación de la Figura 52 para todos los módulos; genera mensaje de error a usuario.

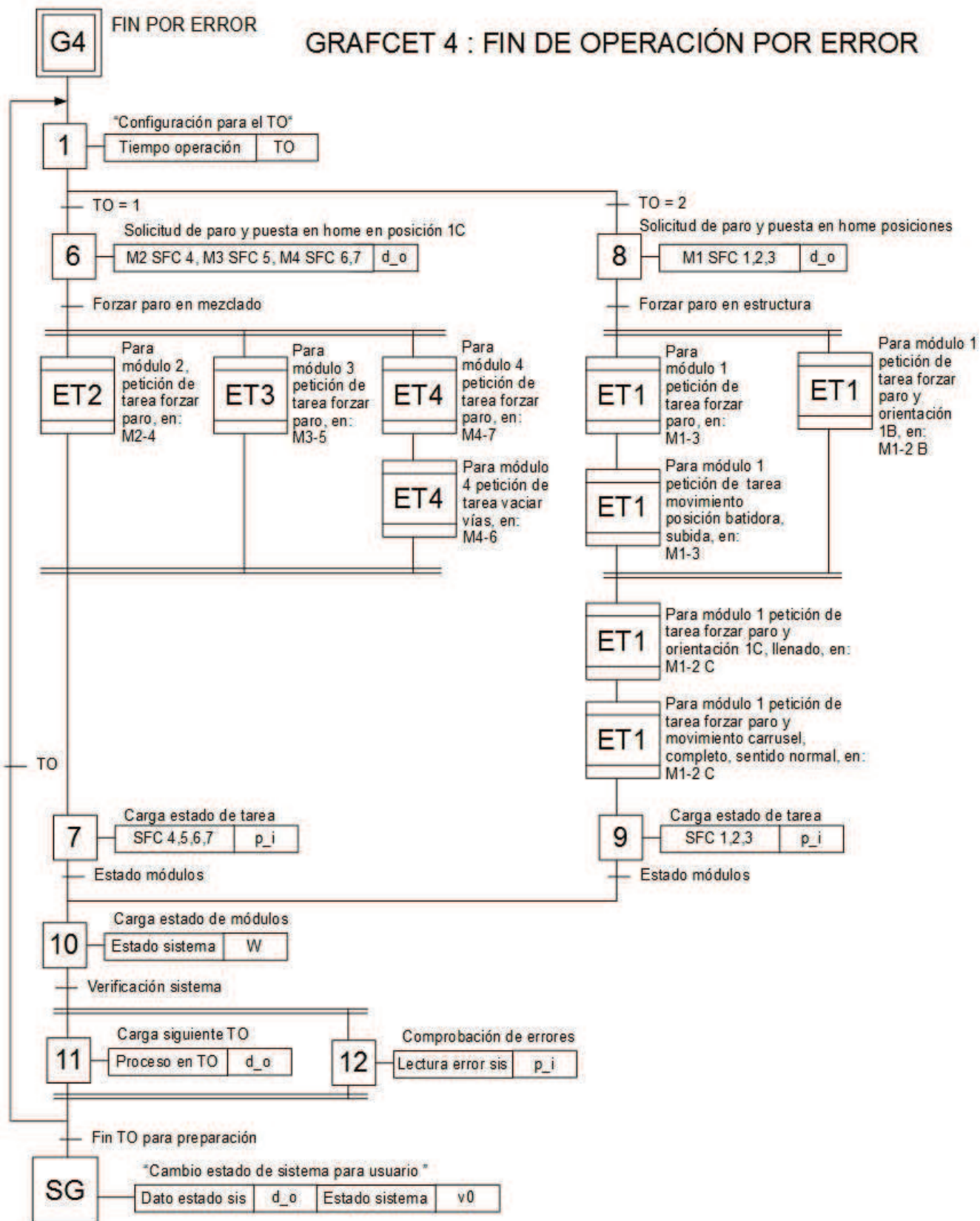


Figura 68. Grafcet encapsulado 4, Fin de operación por error

Como se mencionó al inicio del capítulo, el desarrollo del Grafcet de todo el proceso no se desarrolla debido a que su representación implica varias estructuras especiales y no es el objetivo su implementación en PLC, si no de organizar y dar estructura a las operaciones del proceso manteniendo la relación física como lógica. La implementación, en caso de darle continuidad, es definida y condicionada por el dispositivo PLC pues el desarrollo depende de las características del mismo, existen dispositivos que permiten uso de interrupciones, contadores, times y de protocolos de comunicación específicos. La norma IEC no es específico en cuanto a las reglas de evolución por lo que la simplicidad de los diagramas dependen del diseñador a partir del conocimiento del autómatas programable seleccionado.

Finalmente, las consideraciones para la definición de los módulos a partir de microcontroladores dependen de los paradigmas de programación, para este trabajo se usa los eventos para darle estructura a las operaciones de la unidad de control y el deliberativo para la estructura de tareas de cada módulo. Se retoman las cartas de función para cada Fase de trabajo pasando a bloques funcionales, y los Grafcet encapsulados se usan para la estructura de comunicación y de configuración de los módulos.

4.2 Modularidad del proceso

La modularidad de la unidad de control usa como herramienta la descomposición a nivel software, como sucede en su contra parte de la definición de FBC para sistemas físicos; para la descomposición existen diversos métodos, en este trabajo se desarrolla a partir de las definiciones de funciones de operación de un proceso con las cartas de función secuencial SFC, pues éstas describen gráficamente el comportamiento secuencial de un programa de control; de este modo se estructura la organización interna de un programa descomponiendo el problema de control en partes manejables sobre las SFC, mientras se mantiene una visión de conjunto en cada funcionamiento físico del proceso. Por el análisis basado en componentes y operaciones del proceso se ha definido:

- Estados y condiciones de componentes del proceso.
- Funciones y grupos de operación del proceso.
- Tiempos de operación y de jerarquía de operaciones.
- Cartas de función SFC de las operaciones de módulos.
- Grafcet de los tiempos de operación y estructuras para detección de fallo.

Para la implementación de las operaciones en los módulos se usará microcontroladores PIC, estos no alteran ninguna condición definida en el proceso ni en los tiempos de operación, pues la modularidad, la flexibilidad de operaciones y la multitarea son propiedades que se define por el proceso y sus operaciones, no por el autómatas programable, siendo éste el que se adaptan a las operaciones de gestión y de configuración mediante los paradigmas. Mediante los paradigmas se mantiene los principios de la modularidad sobre el lenguaje de programación: los módulos sólo interactúan con el maestro, son finitos y específicos y definidos en cuanto a tareas para reducir la comunicación, el maestro es el que define al proceso y los datos de los módulos están divididos en configuración, tareas y datos de salida.

El desarrollo de un modelo genérico para los módulos será a partir de las especificaciones, las limitaciones, las entradas y las salidas a nivel de electrónica en sensores, en actuadores y en controladores con base en el proceso.

4.2.1 Módulos

Un módulo de una unidad de control se define como un dispositivo independiente con entradas y salidas bien definidas, donde las operaciones determinan el nivel de escalabilidad y de implementación con relación a las capacidades físicas del controlador. Para este trabajo se toma en cuenta que la unidad de control debe ser modular, por lo que se requiere una estructura de comunicación general, es decir un sistema a modo plantilla para que en el desarrollo de las Fases la implementación del sistema se enfoque en las funciones de operación. A partir la arquitectura del proceso, las especificaciones y limitaciones del proyecto, se definen las características tanto físicas como de operación del sistema modular, éstas son:

Características físicas:

- Acondicionamiento de entradas y de salidas.
- Puertos de comunicación generalizada.
- Indicadores de funcionamiento.
- Fuentes independientes para las etapas de acondicionamiento, control y potencia.

Características de operación:

- Estructura deliberativa: sensado, planeación y actuación.
- Estructura Bitbus de comunicación.
- Estados de Error/Tarea como auxiliares en comunicación.
- Planeación orientada a eventos.
- Independencia en operaciones de ejecución.
- Gestión de comunicación y de proceso bajo un módulo maestro.

Los módulos de la UC se dividen en dos a partir de la distinción de las operaciones del proceso y de las operaciones lógicas. Del análisis de del proceso, se definición al grupo de ejecución como los módulos esclavo por ser directamente los sistemas físicos, mientras que los grupos de planeación y de configuración representa las erogaciones lógicas de la UC, representados por el módulo maestro y el sistema de interfaz. Los módulos de ejecución son:

- Estructura para sistema carrusel, orientación contenedor y posicionamiento de batidora.
- Mezclado, para sistema de mezclado.
- Dosificación lodo, para el sistema de dosificación de efluente.
- Dosificación polímero, para el sistema de dosificación de floculante.

Los módulos de planeación y configuración son:

- Maestro: representa las operaciones de planeación
- Interfaz: para sistema de pantalla y generación de reportes en operaciones de configuración

En el cap. 3.5 se define el concepto de la unidad de control a partir de las especificaciones del proyecto, Figura 24 y 25; para el enfoque modular se establece el concepto de la estructura de comunicación tipo Bitbus que cumple con las especificaciones de la Tabla 16; por otra parte, cada módulo debe estar estructurado de forma genérica para que la implementación de los sistemas de

cualquier etapa de proyecto se enfoquen en las operaciones de proceso. De esta forma, cumplir las características mencionadas depende del establecimiento de la estructura de comunicación modular.

4.2.1.1 Especificaciones y limitaciones

En el cap. 3.3 se establecieron las Tablas 9 y 10 sobre las limitaciones y el grado de restricción, antes del inicio de desarrollo del presente trabajo el equipo de electrónica estableció junto con el equipo de mecánica los actuadores y los sensores de la primera etapa del proyecto, se planteó desarrollar el sistema de control secuencial mediante microcontroladores de la plataforma Arduino por su facilidad de configuración, programación en C y gran cantidad de ejemplos y de literatura en la web.

El sistema inicial fue propuesto con una interfaz intuitiva para el operador en una pantalla gráfica y una unidad de control centralizada en un único microcontrolador de la plataforma Arduino, las características de estos elementos se describen en la Tabla 41. Mientras que la parte de actuadores y sensores por el sistema mecánico se definieron de forma separada, es decir, los actuadores no constan de sensores integrados, salvo el de batidora, para que el sensado no depende del movimiento de los mismos y en caso de fallo puedan ser reemplazados individualmente, Apéndice B. En la Tabla 42 se describen los actuadores y sensores con sus características para la etapa 1 2 y 3. Para la continuidad de las demás etapas no se consideró el incremento de microcontroladores.

Tabla 41. Especificaciones iniciales del sistema de control			
Sistema	Características técnicas	Programación	Funciones
Interfaz	Pantalla 3.5" con touch resistivo, versión 1	Configuración por comandos en RS232 a partir de librería de Arduino	• Tipo, tamaño y color de letra • Orientación • Color de fondo • Detención de una coordenada de toque a la vez
Control	Arduino mega, microcontrolador atmega2560 de 54 pines, a 5 [V] y 16 [MHz]	C en IDE de Arduino	Funciones de configuración genéricas

Tabla 42. Especificaciones iniciales de actuadores y sensores			
Tipo	Señal	Acondicionamiento	Características
Actuador	PWM a movimiento mecánico	Etapa de potencia, por puente h	Diferentes tipos por cada sistema
Sensor	Analógica a ADC	Etapa de acondicionamiento por filtrado	Igual para cada sistema, a 5 [V] ópticos y de carrera

A partir del desarrollo de la etapa dos del proyecto, en el Apéndice B se establece la necesidad de usar un sistema de control modular para cumplir con las especificaciones originales del proyecto, Tabla 11, además se mantiene la interfaz con el microcontrolador Arduino debido a que las librerías de configuración de pantalla están dedicadas al mismo por el fabricante, Apéndice C. Por lo que se tomaron las especificaciones iniciales como limitaciones. Para poder satisfacer el concepto de la unidad de control consideradas las especificaciones del proyecto se establece una separación para la implementación con microcontroladores, las operaciones de ejecución y de planeación se usan con microcontrolador PIC, mientras que para las operaciones de configuración se usa Arduino.

A su vez, debido a las funciones de la pantalla y la generación de reportes se divide la implementación de las operaciones de configuración en dos microcontroladores para cumplir los objetivos de la Tabla 13 del cap. 3, de tal forma que se usa uno para la pantalla y otro para los reportes; grosso modo la razón por la que desarrollan las operaciones de reporte en un micro tipo Arduino es por la comunicación SPI que requiere una memoria SD y por estar directamente comunicado con el primer Arduino.

El uso de microcontroladores PIC16F887 se fundamenta en las especificaciones del proyecto, Tabla 11; durante el desarrollo de la etapa dos del proyecto, las limitaciones que representó la interfaz sobre el microcontrolador Arduino mega no permitían resolver todos los sistemas de forma secuencial, por otra parte, en un inicio se planteó la solución del proyecto por microcontroladores, por tanto se propuso el uso de microcontroladores PIC para darle continuidad al sistema de control por la comparación de costo y operaciones. En la Tabla 43 se describen los objetivos del concepto inicial de la UC, Tabla 13, sobre la implementación de microcontroladores. En la Tabla 44 se describen las consideraciones para usar microcontroladores PIC como solución de los módulos de operaciones de planeación y ejecución en un sistema modular.

Tabla 43. Definición de los microcontroladores por limitaciones			
Objetivos	Microcontrolador	Descripción	Procesos de operación
Parámetros de proceso	Arduino Mega, atmega2560	Operaciones de interfaz para configuración de información visual para usuario, se comunica con reportes y módulo maestro	Configuración
Muestra de estado de sistema			
Operaciones de prueba de sistema			
Configuración de sistema	Arduino uno, atmega328p	Guarda eventos durante el proceso en una memoria SD para generar reportes de su uso, para documentación y para mantenimiento.	
Reporte de eventos			
Proceso secuencial	PIC 16F887	Operaciones de planeación y ejecución del proceso, comunicación paralela entre módulos y serial con la interfaz	Planeación Ejecución
Cambios en proceso			
Detección de errores			
Mantenimiento de sistema			

Tabla 44. Comparación de microcontroladores			
Características	Arduino, 8bit	Atmel, 8bit	PIC, 8 bit
Velocidad procesamiento	16 [MHz]	16 [MHz]	20 [MHz]
Programación	C	C, ensamblador	C, ensamblador
Costos [\$MX], abril 2014	800 mega 480 uno	60	55
Documentación de uso	Si, configuración general	Si, configuración específica	Si, configuración específica
Tiempo de implementación	corto	Largo*	Largo*
* Depende del conocimiento en el microcontrolador, en comparación de Arduino			

Cabe aclarar que un dispositivo PLC puede cumplir con las operaciones de planeación y de ejecución, pero no fue el objetivo del proyecto usarlos como solución, por otra parte, otros microcontroladores como de Atmel o Texas representaron un tiempo para su estudio e implementación; de esta forma la causa principal de escoger PIC fue por el conocimiento de sus funciones de configuración desde ensamblador hasta las aplicaciones en C, su funcionamiento y capacidades durante ejecución; además su la implementación pese a ser extensa por código no afecta en tiempo de desarrollo por el conocimiento del micro.

Finalmente, la justificación para la implementación de los módulos con microcontroladores PIC ante Arduino fue por la relación de las características de los módulos, las especificaciones y los objetivos del proyecto ante la flexibilidad del uso de las funciones del micro en tiempo de ejecución. Estas consideraciones son descritas en la Tabla 45.

Tabla 45. Funciones de microcontrolador PIC para las características de módulo		
Características de módulo	Funciones y aplicación	Propiedades
Comunicación paralela	I2C: para módulos	• Interrupción • Bidireccional síncrona por maestro hacia esclavos • 254 direcciones de esclavos
Comunicación en serie	RS232: para maestro-interfaz	• Interrupción • Bidireccional asíncrona que presenta menos fallos de compatibilidad con dispositivos de diferentes fabricantes
Detección de errores por maestro	Interrupción en puerto Cambio de estado: entrada	• Interrupción • Se debe darle sentido al cambio de estado
Lectura velocidad	Interrupción externa Cambio de flanco: en entrada de encóder	• Interrupción • Se puede cambiar durante ejecución el tipo de flanco de interrupción
Lectura tiempo	Timer 1: tiempo ejecución de operación y cambio de flanco	• Interrupción por desborde • Se puede obtener contador • Configuración de lectura del oscilador
Entradas y salidas generales	Configuración y lectura de puertos	• Lectura y escritura de puerto en una sola instrucción

4.2.1.2 Pines de control y comunicación

El análisis de entradas y salidas establece al microcontrolador el número de pines de entrada y de salida que debe tener un módulo para la interacción con sus componentes electrónicos y con otros módulos para la ejecución y la configuración de funciones de operación. En el estudio del proceso se establece la relación con el sistema físico con la arquitectura y con la jerarquía de componentes, elementos descritos en Apéndice B; estos elementos definieron las condiciones y estados de operación de las funciones de cada módulo para los SFC del proceso.

Para los pines de I/O de los módulos se debe definir las señales de comunicación de forma general y las señales de control de forma específica, tomando en cuenta la implementación en microcontroladores, para que la implementación dependa de las operaciones de cada módulo. De esta

forma se va analiza y clasifica los estados de entradas y de salidas de los módulos en dos tipos de acuerdo a la funcionalidad: por componentes de ejecución y de comunicación. Para determinar el número de pines por cada módulo se inicia con el análisis específico de componentes de ejecución para determinar el número y disponibilidad de pines en el microcontrolador, posteriormente se establecen de forma general por módulos y finalmente se establecen los de comunicación.

De forma general, la mayoría de los pines del microcontrolador PIC16F887 son multipropósito como se muestra en la Figura 69, cuenta con 40 pines de los cuales 36 pueden ser usados de forma digital o analógica, en funciones específicas de comunicación o en interrupciones. Por ejemplo, la asignación RA3/AN3/Vref+/C1IN+ para el quinto pin del microcontrolador indica que éste dispone de las siguientes funciones:

- RA3 Tercera entrada/salida digital del puerto A
- AN3 Tercera entrada analógica
- Vref+ Referencia positiva de voltaje
- C1IN+ Entrada positiva del comparador C1

Pin Diagrams – PIC16F884/887, 40-Pin PDIP

40-pin PDIP

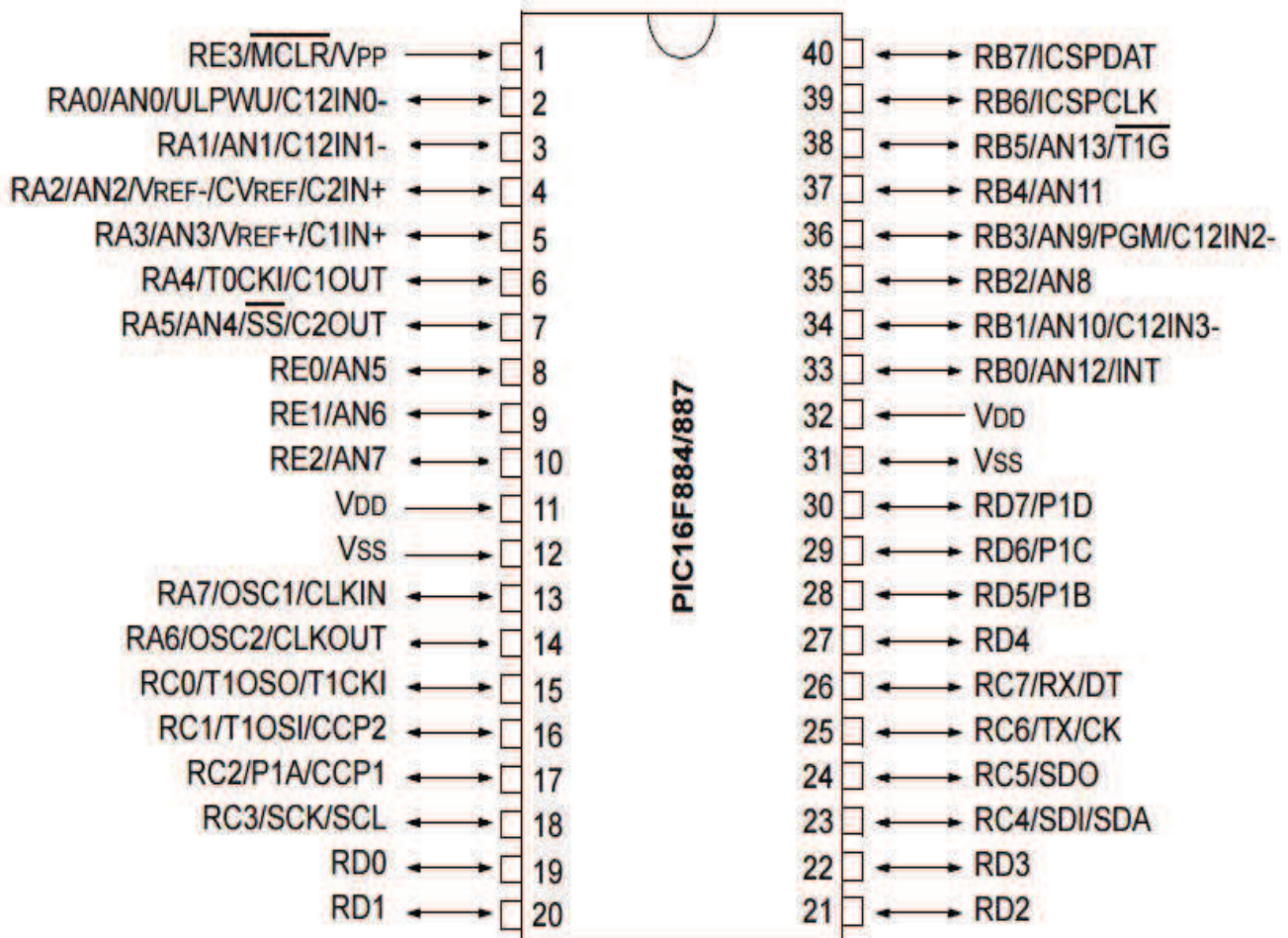


Figura 69. Diagrama de pines del microcontrolador 16F887, datasheet Microchip

La funcionalidad de los pines del microcontrolador es muy útil puesto que permite un mejor aprovechamiento de los recursos del microcontrolador sin afectar a su funcionamiento. Estas funciones de los pines no se pueden utilizar simultáneamente, sin embargo, se pueden cambiar en cualquier instante durante el funcionamiento. En el desarrollo de cada módulo se definirán los pines correspondientes y sus funciones para la implementación. En la Tabla 46 se muestra los puertos del microcontrolador con la distribución de pines de forma general a manera de referencia en disponibilidad para componentes de ejecución y de comunicación.

Tabla 46. Puertos del microcontrolador PIC16F887				
Puerto – Pines	Tipo de entrada	Tipo de salida	Función especial	Uso recomendado*
A – 8	Digital, 8 pines Analógica, 5 pines	Digital, 8 pines	Oscilador externo Convertidor D-A	Entradas analógicas y auxiliar en digital
B – 8	Digital, 8 pines Analógica, 6 pines	Digital, 8 pines	Interrupciones Convertidor D-A Programación	Entradas digitales
C – 8	Digital, 8 pines	Digital, 8 pines Analógica, 2 pines	RS232 I2C	Comunicación y PWM
D – 8	Digital, 8 pines	Digital, 8 pines		Salidas digitales
E – 4	Digital, 4 pines Analógica, 3 pines	Digital, 4 pines	Reset Programación	Auxiliar en digital y en analógico
*Por experiencia de usuario				

La arquitectura y la jerarquía de los componentes, cap. 4.1.2 y 4.1.3, identifican los elementos de actuación y sensado junto con la relación de su proceso de operación, Tabla 14, 16 a 19 y 21; en cuanto a la unidad de control, se analiza la electrónica para definir las variables y elementos de entrada y salida para el microcontrolador con el fin de reducir las operaciones lógicas y establecer una estructura general para el módulo. En la Tabla 47 se describen los actuadores con sus sensores y su tipo de operación con lo que se define el tipo de pin del microcontrolador en referencia a las Tablas 29 a 32.

Tabla 47. Pines de I/O para componentes de ejecución					
Módulo	Etiqueta-Actuador: Sensor	Movimiento: Sensado	Tipo de control	Tipo de sensado	Pines de control
Estructura	M1 - Carrusel: Encóder - Principal - Auxiliar	Bidireccional : Pulsos encóder	Habilitado por PWM, dirección por un pin	Filtrado por comparador	I: 2 O: 2
	M2 - Orientación 1B: Posición - Alto - Bajo	Bidireccional: Estado lógico	Habilitado por PWM, dirección por dos pines	Filtrado por paso bajas	I: 2 O: 3
	M3 - Orientación 1C: Posición - Alto - Bajo	Bidireccional: Estado lógico	Habilitado por PWM, dirección por dos pines	Filtrado por paso bajas	I: 2 O: 3
	M4 - Posición: Posición Alto	Bidireccional: Estado lógico	Habilitado por PWM, dirección por un pin	Filtrado por paso bajas	I: 2 O: 2

	• Bajo				
Mezclado	M5 - Batidora: Encóder • Velocidad	Unidireccional: Pulsos encóder	Habilitado por PWM y un pin de dirección	Filtrado por comparador	I: 1 O: 2
Lodo	M6 - Efluente: Volumen • Ultrasónico	Unidireccional: Tiempo respuesta	Habilitado por un pin de dirección	Filtrado digital por promedios	I: 1 O: 2
Polímero	M7 - Flocculante 1: Encóder	Bidireccional: Pulsos encóder	Habilitado por PWM, dirección por un pin	Filtrado por paso bajas	I: 1 O: 2
	M8 - Flocculante 2: Encóder	Bidireccional: Pulsos encóder	Habilitado por PWM, dirección por un pin	Filtrado por paso bajas	I: O: 2
	M9 - Flocculante 3: Encóder	Bidireccional: Pulsos encóder	Habilitado por PWM, dirección por un pin	Filtrado por paso bajas	I: 1 O: 2
	M10 - Flocculante 4: Encóder	Bidireccional: Pulsos encóder	Habilitado por PWM, dirección por un pin	Filtrado por paso bajas	I: 1 O: 2

En la Tabla 48 para cada módulo de forma general se establece el número de pines para sus componentes electrónicos de ejecución; se usará como referencia para el establecimiento del sistema electrónico.

Tabla 48. Pines de control para módulos			
Módulo	Componentes	Pines de control	Funciones en microcontrolador
Estructura	Actuador, 4	10	PWM
	Sensor, 8	8	Timer
Mezclado	Actuador, 1	2	PWM
	Sensor, 1	1	Timer Interrupción
Lodo	Actuador, 1	2	PWM
	Sensor, 1	1	Timer
Polímero	Actuador, 4	8	PWM
	Sensor, 4	4	Timer Interrupción
Maestro	Tarea	1*	Timer
	Error	9*	Interrupción
Reportes	Memoria SD	5	SPI
*Se considera un sistema de control con 8 módulos para operaciones de ejecución , 8 pines de error por cada módulo, y 1 pin para lectura general de error y tarea			

Para los componentes de comunicación se ha establecido por el cap. 3 el concepto de la estructura de comunicación para el sistema de control y se establece como modular; el modelo Bitbus tiene una estructura de árbol donde cada nodo puede generar una comunicación paralela o serial, a modo de repetidor, de maestro a esclavo, cap. 2.4.4. El microcontrolador PIC16F887 permite el uso de comunicación paralela I2C y punto a punto RS232, por configuración puede usarse ambas para una misma aplicación en dos formas: por interrupción o por lectura. Esta característica permite la

independencia de operaciones y una flexibilidad de comunicación; es importante que se defina el tiempo de operación de las comunicaciones en el sistema y los casos de ejecución para no perder datos.

Los módulos de operaciones de planeación y configuración se requiere uniformizar el tipo de comunicación entre módulo maestro con la interfaz y ésta con los reportes debido al tipo de microcontroladores; siguiendo la estructura de Bitbus para estos casos se mantiene la comunicación serial por RS232, siguiendo la estructura de operaciones cap. 4.1.4. En cuanto a los módulos para operaciones de ejecución tenga definidas tres tipos de líneas de comunicación:

- Paralela: para comunicación con el módulo maestro, I2C
- Serial: para monitoreo en modo programador y operaciones de mantenimiento, RS232
- Estados lógicos de operación: monitoreo de estado tarea y error

Este último punto surge por la consideración del uso de I2C, de forma general al definir un módulo esclavo, éste no permite inicializar la comunicación, siendo el maestro quien se encarga de esto, de tal forma que los esclavos trabajan por interrupción, por tanto para que el maestro monitoree los estados de tare y error en tiempo de operación se requiere solicitar a todos los esclavos los estados, reduciendo la eficiencia de las operaciones de los esclavos, por lo que se propone unas líneas auxiliares que hagan independiente a la comunicación de dichos estados, así solo se solicitan datos al inicio y fin de cada tiempo de operación. Esta condición es descrita en el siguiente capítulo. Y en la Tabla 49 se muestra de forma general los pines de comunicación entre los módulos.

Tabla 49. Pines de comunicación para módulos				
Emisor	Comunicación	Receptor	Dispositivo	Características
Interfaz	RS232: 3	Reportes	Arduino	Tres RS232 dedicados*. Disponible en Arduino mega, sin interrupción. 6 pines
		Pantalla	Pantalla TFT	
		Maestro	PIC	
Reportes	RS232: 1	Interfaz	Arduino	Un RS232 dedicado, disponible Arduino uno, sin interrupción. 2 pines
Maestro	RS232: 1	Interfaz	Arduino	Un RS232 y un I2C dedicados. Dos pines para monitoreo. Uso de interrupciones. 6 pines
	I2C: 1	Esclavos	PIC	
	Error: 1			
	Tarea: 1			
Esclavos	RS232: 1	Monitoreo	Monitor serial	Un RS332 y un I2C dedicados. Dos pines para monitoreo. Uso de interrupciones. 6 pines
	I2C: 1	Maestro	PIC	
	Error: 1			
	Tarea: 1			

*El termino dedicado hace referencia a la característica del microcontrolador de tener por hardware un puerto específico para la comunicación, en tiempo de ejecución se almacena en acumulador el dato de entrada para ser tratado en siguiente ciclo máquina.

4.2.2 Estructura de comunicación de módulos

El diseño de la estructura de comunicación permite que el diseño de la implementación de los módulos sea orientado a las operaciones del proceso, durante el desarrollo de la segunda etapa del proyecto se planteó el sistema modular y con ello la necesidad de establecer una comunicación

general, eficiente y sencilla. A partir del microcontrolador propuesto se demostró que por configuración del micro se puede implementar más de un tipo de comunicación para una aplicación, con flexibilidad por el modo de interrupción para las operaciones a diferencia de Arduino, que depende de las librerías y su comportamiento requiere estudio del microcontrolador.

Siguiendo el modelo Bitbus, se asocia el modelo de la unidad de control por los elementos definidos como limitaciones de diseño y los módulos de los sistemas. En la Tabla 50 se clasifican los bloques y los niveles de la estructura Bitbus y en la Figura 70 se pone la estructura grafica del bus.

Tabla 50. Bloques de Bitbus en los módulos				
Módulo	Tipo bloque	Nivel en bus	Tipo nodo receptor	Tipo nodo emisor
Interfaz	Maestro	0		RS232
Pantalla	Esclavo	1	RS232	
Reportes	Esclavo	1	RS232	
Maestro	Esclavo Maestro	1	RS232	I2C
Estructura	Esclavo	2	I2C	RS232
Mezclado	Esclavo	2	I2C	RS232
Efluente	Esclavo	2	I2C	RS232
Floculante	Esclavo	2	I2C	RS232
Monitor serial	Repetidor	3	RS232	

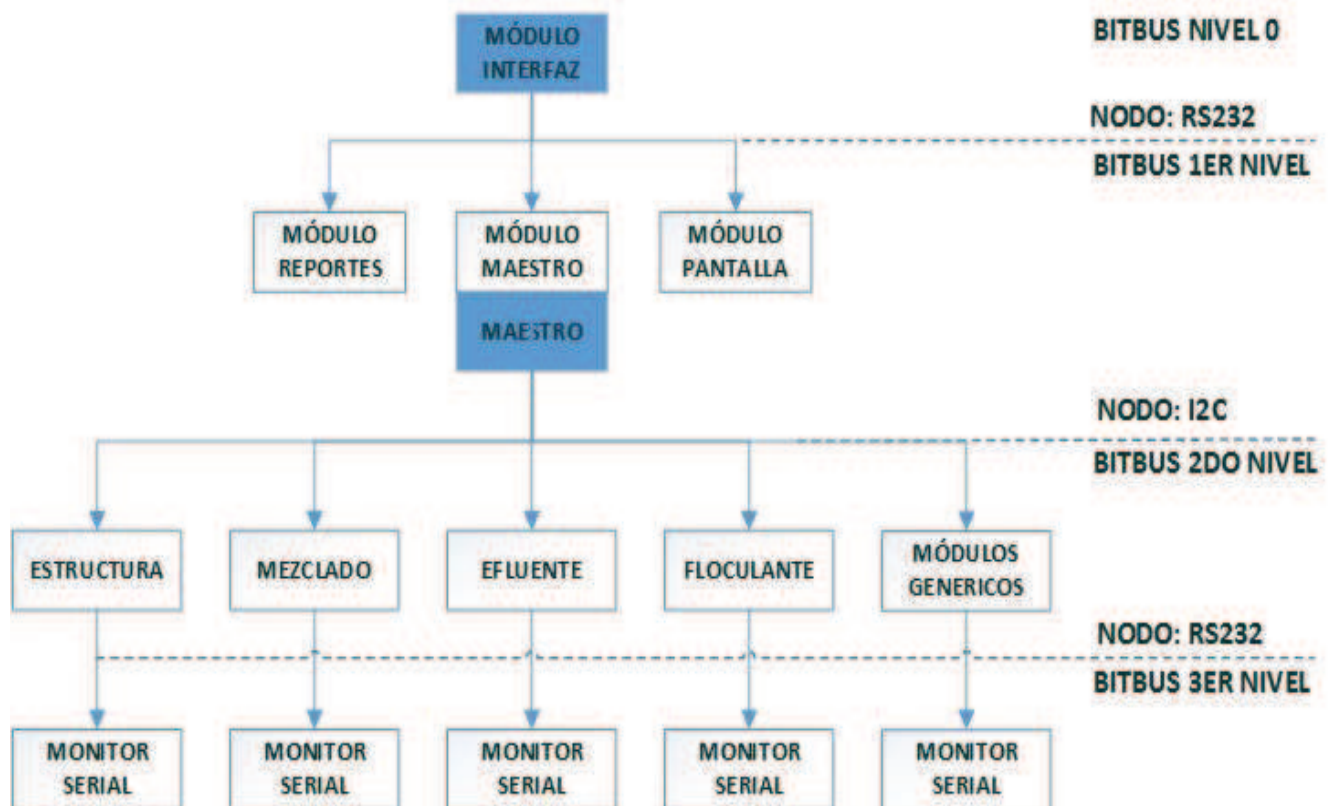


Figura 70. Estructura Bitbus de la Unidad de Control

Para cada tipo de comunicación se tiene especificaciones físicas como de configuración para su uso. El caso sencillo es RS232, el cual al ser asíncrono los datos deben ser medidos con precisión en cada hilo de comunicación, por lo que para microcontroladores con puertos dedicados no tiene

problemas aun con diferentes familias de dispositivos, su desventaja es que es punto a punto, de manera auxiliar puede usarse elementos pull-down en cada extremo del hilo para reducir el ruido del ambiente. La aplicación de la comunicación RS232 se establece con los siguientes puntos:

- Se usa una velocidad en baudaje de 115200, se envían caracteres de 8 bits en código ASCII, equivale aproximadamente a 70 [us] por byte.
- La cantidad de los datos de envío depende del tipo de operación: para solicitud de tarea se envía caracteres ASCII mayúsculas mientras que para configuración son del tipo numérico, se envía en ternas para Long y en cuartetos para Float. Por otra parte la solicitud de lectura es para un solo byte que tiene el estado del módulo.
- El envío de datos de configuración se usa la forma de ECO, los datos solo pueden ser enviados para iniciar un tiempo de operación o solicitar un dato de información. La respuesta se espera al terminar el tiempo de operación solicitado.
- La respuesta del ECO en envío de configuración o en solicitudes de tarea tiene dos variantes, mayúsculas son para fin sin errores y minúsculas fin con errores.
- Entre el maestro e interfaz no es necesario definir el conteo de tiempo de duración de operaciones, al usar ECO el maestro envía a la interfaz el dato de tipo de finalización de tarea, mayúscula o minúscula.

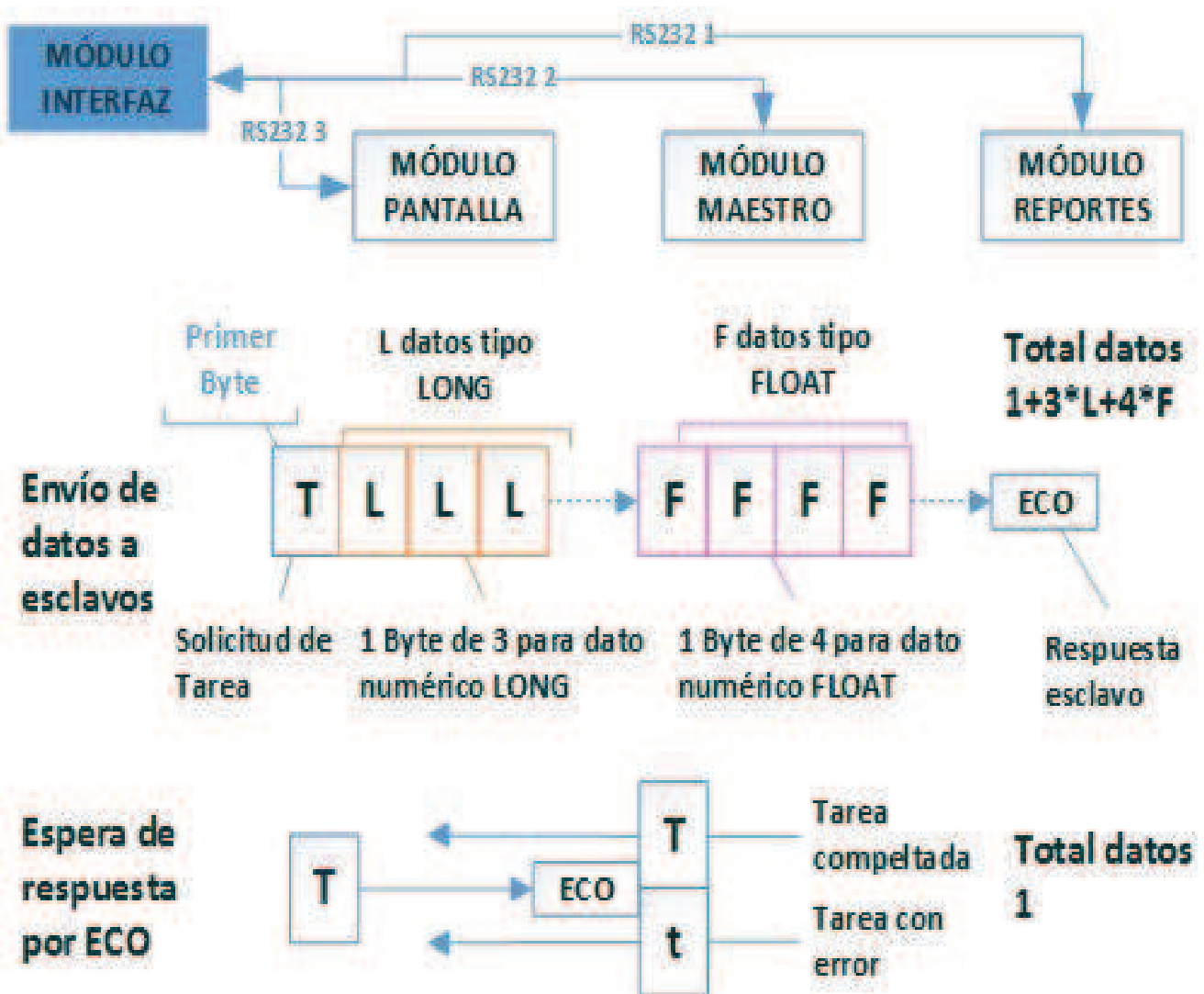


Figura 71. Estructura de datos para comunicación RS232

El uso de este protocolo queda definido para los módulos de interfaz y el módulo maestro, independientemente de los dispositivos del sistema de interfaz, el módulo maestro debe comunicarse por RS232 a este sistema debido a que las líneas de comunicación debe ser independientes entre las operaciones de ejecución y de configuración para asegurar el buen funcionamiento y evitar errores por coordinación, de esta forma la configuración de los procesos de planeación pueden ser implementados en cualquier dispositivo que soporte RS232 o USB. Por otra parte, para la estructura de envío y recepción de datos de comunicación quedan definidos en el siguiente diagrama, a partir de las consideraciones anteriores, Figura 71.

A diferencia de I2C, RS232 es más sencillo de implementar en cuanto lectura y escritura, pues se trata de una comunicación punto a punto, no hace falta distinguir direcciones y aseguramiento de envío de datos pues al momento de que inicia el envío el receptor se prepara por hardware, es el procesamiento donde se determina qué dato se ha enviado y de ahí que se identifique error por dato no esperado. Mientras que en I2C se debe definir las direcciones y asegurar los envíos de datos en match con el reloj de la comunicación, es sensible por la programación de configuración pero robusta cuando se aseguran las lecturas y envíos de datos.

La comunicación en paralelo tiene la condición física de requerir pull-ups para reducir las caídas de voltaje en función de la velocidad de transmisión. En una configuración maestro/esclavo, la comunicación únicamente la pueden inicializar los maestros, los esclavos manejan interrupción por hardware para realizar la solicitud de recepción o envío de datos. La evolución de los bloques de envío y recepción de datos no es definida en cuanto al orden, por lo que el programador debe establecer el número de datos y las direcciones de memoria; de forma general las consideraciones para el uso de I2C son:

- Se usa una velocidad en baudaje de 400 [KHz] con resistencias pull-up en ambos hilos de comunicación de acuerdo a hojas de especificaciones del microcontrolador.
- Los tipos de los datos para envío es en forma de arreglo de direcciones hexadecimales: solicitud de tarea 0x0; para datos numéricos de configuración de 0x1 a 0xA; para datos referencias a direcciones de 0xA0 a 0xF8; de 0xB a 0x9F quedan libres.
- Los datos numéricos son enviados en tercias para representar números tipo Long y en cuartetos para tipo Float; los datos de tareas son enviados como un solo dato y su valor depende el número de tareas que pueda realizar. Para generalizar, se definen 5 datos Long y 5 datos Float; correspondientes a datos numéricos de los parámetros de configuración.
- Para los datos de lectura, corresponden a los estados de tarea, de los puertos de salida y al estado de error; el maestro planifica las operaciones por lo que es quien determina en qué módulo y en qué tarea se produce error; aunque cada módulo cuenta con variables de proceso, estas son para uso de monitoreo.
- Para generalizar los módulos, los datos de envío y recepción será iguales, siendo que cada módulo determina cuáles son sus datos; esto aprovechando la velocidad de baudaje y no depender de la reconstrucción por hardware; por tanto la cadena de datos de envío será: 1 de tarea + 3*5 Long + 4*5 Float = 36 bytes.
- Es importante mencionar que para el microcontrolador PIC16F887 existen algunos errores al usar compiladores en C para I2C en modo lectura, por lo que se hará una variación en la evolución de los bloques de comunicación recomendados en hojas de especificaciones. Se considera que solamente se envía y reciben datos fuera de los tiempos de operación y únicamente se interrumpirá en caso de error.
- Del punto anterior, se motiva el uso de líneas auxiliares independientes a la comunicación para el monitoreo de estado de error y de tarea durante tiempo de operación.

En la Figura 72 se muestra la estructura de envío y recepción de datos entre el módulo maestro y los esclavos del proceso de ejecución. Nótese que a diferencia de RS232, el envío usa la línea auxiliar de tarea debido a que el paquete de datos es generalizado por tanto una vez enviado el esclavo debe reconstruir los suyos. En comparación a RS232, el ECO se sustituye por la línea auxiliar de tarea, el cual indica el inicio y fin de operaciones, como la de datos. De esta forma el maestro espera el fin de envío a su vez espera por Timer en caso de error comunicación.

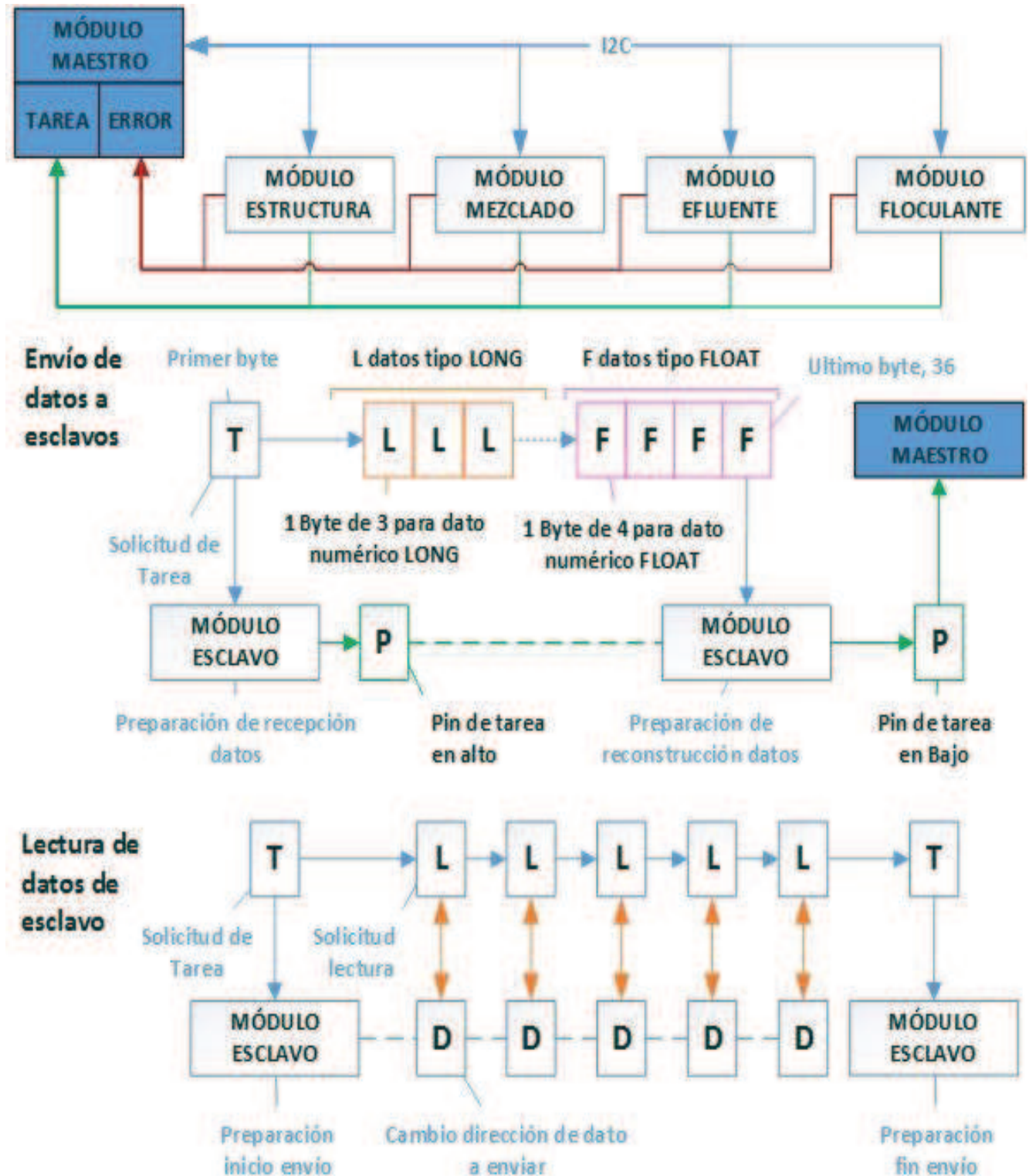


Figura 72. Estructura de datos para comunicación I2C

Tabla 51. Estructura de comunicación

TO	RS232		I2C		Auxiliar tarea
	Emisor	Receptor	Emisor	Receptor	
0	Determina tipo solicitud		Determina tipo solicitud		
1	Solicitud envío tarea	Recibe tarea	Solicitud envío tarea		
2	Espera tipo fin	Determina tipo	Inicia comunicación	Recibe inicio	
3		Ejecución tarea	Elige esclavo	Determina si es él	
4		Fin tarea	Envía tarea	Recibe tarea	
5		Determina tipo fin	Termina comunicación	Procesa elección de tarea	En alto tarea
6	Recibe tipo fin	Tipo fin	Espera fin tarea o erro	Inicia tarea	
7	Determina acción de fin			Termina tarea, determina tipo fin	
8	Solicitud envío datos	Recibe tarea	Lectura fin tarea	Fin tarea	En bajo tarea
9	Espera: time	Prepara recepción	Determina acción de fin		
10		preparado	Solicitud envío datos		
11	Envío paquete datos	Recibe paquete	Inicia comunicación	Recibe inicio	
12	Espera ECO	Reconstruye datos	Elige esclavo	Determina si es él	
13		Determina tipo fin	Envía tarea de datos	Prepara recepción datos	
14	Recibe ECO	Envía ECO	Envía paquete de datos	Recibe paquete	
15	Determina acción de ECO		Termina comunicación	Reconstruye datos	
16	Solicitud lectura datos	Recibe tarea	Solicitud lectura datos		
17	Espera dato	Prepara dato	Inicia comunicación	Recibe inicio	
18	Recibe dato	Envía dato	Selecciona esclavo	Determina si es él	
19			Envía tarea lectura	Recibe tarea lectura	
20			Reinicia comunicación	Reset registro dato	En alto tarea
21			Envía solicitud lectura por dirección	Determina si es él, Registro de lectura	
22			Lee dato	Envía dato	
23			Detiene comunicación		
24			Inicia comunicación	Recibe inicio	
25			Envía solicitud lectura por dirección	Determina si es él, Registro de lectura	
26			Lectura pivote	Envía dato pivote	
27			Detiene comunicación	Prepara siguiente	
28			Si no más datos: Inicia comunicación	Recibe inicio	
29			Selecciona esclavo	Determina si es él	
30			Envía tarea fin lectura	Recibe tarea	
31			Detiene comunicación	Reinicio registros de envío	
32			Espera fin de lectura	Fin tarea lectura	En bajo tarea

De las estructuras de datos y las consideraciones para cada tipo de comunicación, se establecen las estructuras de comunicación mediante rutinas para el envío y recepción de datos de configuración o de ejecución. En la Tabla 51 se describen las rutinas para estructura de comunicación en relación a la estructura de envío de datos de la Figura 72; para el desarrollo se considera que las solicitudes de escritura y lecturas se dividen en solicitud de tarea para representar todas las operaciones de configuración y ejecución de funciones, y en solicitud de datos como estados de puertos para I2C o estado de módulo para RS232. Las recepciones en dispositivos PIC son manejadas por interrupción, por hardware y por software, mientras que en Arduino son por hardware y por software.

Se definen tiempos de operación TO para comparación en cuanto a operaciones de la unidad de control y para establecer sus bloques funcionales de su implementación, no representan ciclos de máquina ni tiempo en segundos, representan operaciones de la unidad de control. La condición para envío y recepción es cuando no hay operaciones en ejecución para no interrumpir las operaciones del proceso, se fuerza en RS232 la espera del ECO mientras que para I2C se espera el pin de tarea en los maestros, además para I2C se modifica la lectura de datos por las consideraciones en puntos anteriores. De la Tabla 51 se tienen las siguientes observaciones para la comunicación RS232 entre dispositivos de operaciones de configuración:

- La solicitud de envío de tarea toma 7 TO.
- La solicitud de envío de datos toma 7 TO.
- La solicitud de lectura de dato toma 3 TO.
- El dato de lectura en TO 18 es uno y su valor depende del estado del módulo para tarea o error, no se requiere ECO.
- Los tipos de fin van en relación al ECO, siendo Mayúscula el término exitoso y minúscula por error, es el maestro quien determina la acción después de la lectura, dependiendo del tiempo de operación de proceso en ejecución.

Las observaciones para la comunicación I2C entre dispositivos de operaciones de planeación y ejecución son:

- Para las solicitudes, en el primer TO el maestro prepara por software los registros a enviar.
- La solicitud de envío de tarea toma 8 TO, se espera a que se procese la tarea en las líneas auxiliares. Para el error se determina en su bloque funcional por requerir interrupción.
- La solicitud de envío de datos toma 6 TO, al no requerir confirmación de dato es rápido, pues en envío por hardware permite asegurarlo y no falla por esta operación.
- La solicitud de lectura de datos toma 17 TO, está dividido en 3 partes, la que anuncia al esclavo que se va a requerir lectura 16-19, la que procesa la lectura 20-27, y la que termina la que anuncia al esclavo que se termina la lectura 28-32.
- Se recibe un dato entre el tiempo de operación 16-27, repitiéndose del 20 al 27 hasta que el maestro termine de leer. Se establece de esta forma, por las consideraciones del uso de I2C.
- En los tiempos de operación donde se espera el fin de tarea, puede habilitarse el fin por error por Timer.

La descripción de las operaciones específicas para las estructuras de datos permite plantear los bloques funcionales para microcontroladores. En este punto del diseño, al estar orientado a microcontroladores no se desarrolla Grafcet por el dispositivo; en caso de seguir con PLC se retoma su construcción al quedar definidos los pines de entrada y de salida, las estructuras de datos y las estructuras de comunicación. Los procesos a describir son dos: reconstrucción de datos y detección de errores en maestro y esclavo.

Reconstrucción de datos

Para el envío de parámetros de configuración y de datos en los módulos esclavo está puesto en paquetes, la estructura general es: el primer dato indica la tarea, del segundo hasta $L*3$ datos representan numéricos tipo Long y finalmente hasta $4*F$ datos numéricos tipo Float. De esta forma se plantea asegurar la generalización de la comunicación al no atender de forma particular cada módulo, así el receptor debe de separar los datos que le corresponden.

Los parámetros fueron planteados en la arquitectura del proceso para el proyecto, por lo que durante su desarrollo no se espera que incrementen pues las demás etapas son de limpieza de contenedores, de análisis y de recolección de muestras, los cuales en su ejecución representan movimientos mecánicos sin requerir parámetros de configuración. Los parámetros a enviar separados por tipo están en la Tabla 52.

Tabla 52. Parámetros en datos de comunicación				
Tipo dato	Parámetro	Unidades	Rango de valor*	Módulo Receptor
LONG	Iteraciones	Numérico	0-1000	Maestro, Reportes
	Velocidad de mezclado	Rpm	0-500	Mezclado, Reportes
	Tiempo de mezclado	S	0-100	Mezclado, Reportes
	Cantidad de efluente	MI	200-400	Efluente, Reportes
	Tipo de polímero 1	Numérico	0-1000	Reportes
	Tipo de polímero 2	Numérico	0-1000	
	Tipo de polímero 3	Numérico	0-1000	
	Tipo de polímero 4	Numérico	0-1000	
FLOAT	Cantidad floculante 1	MI	100.00-300.00	Floculante, Reportes
	Cantidad floculante 2	MI	100.00-300.00	
	Cantidad floculante 3	MI	100.00-300.00	
	Cantidad floculante 4	MI	100.00-300.00	
*Del Apéndice A se definen los rangos de parámetros del proceso en laboratorio				

Para el caso de la comunicación en los módulos de planeación y ejecución se plantean 5 Long y 5 Float como factor de seguridad en caso de escalamiento en la aplicación; este número de datos no implica que la interfaz deba enviar esa cantidad de datos, es el módulo maestro quien acomoda el paquete. Por otra parte, la descomposición de los datos numérico es para asegurar la reconstrucción del dato pues la comunicación se basa en protocolos de 8 bits, por tanto la reconstrucción por software permite no depender ni modificar funciones internas de comunicación y generalizar el envío de datos. El sobre escalamiento de datos no afecta en tiempos de ciclo máquina ni en el proceso, debido a que por condición se envían y reciben datos al inicio y fin de operaciones de proceso, mientras que se hace el envío a manera de interrupción en caso de error.

El dato Long puede descomponerse por representación de números de 0 a 99: $27,474 = 02[2] 74[1] 74[0]$, donde $[]$ indica la posición en el arreglo; la razón por la cual dejarla en un rango de hasta 999,999 es por el rango de los parámetros de envío y como factor de seguridad para otro tipo de aplicaciones al ser la estructura de comunicación genérica. Mientras que el dato Float puede descomponerse por representaciones 4 dígitos, de igual forma que el Long pero agregando 2 decimales. Esto por las consideración de los rangos de los parámetros, así $27474.77 = 02[3] 74[2] 74[1] 77[0]$ con un rango de hasta 999999.99. Por programación en C un dato Float puede representarse por 4 datos tipo int8, pero por el tipo de aplicación y el tipo de datos no se recurre a esa conversión.

Para la descomposición se siguen un algoritmo simple: mediante divisiones por 10 se separa en pares el número por unidades a miles; se inicia con división entre 10 para separar el residuo como el primer byte, si la parte entera puede ser dividido entre 10 se vuelve a separar. Para el caso dato Float se inicia con la separación de la parte decimal hasta que el número ya no es divisible entre 10. Por condición del rango de datos, no se espera perder información.

Para la reconstrucción el receptor debe conocer qué datos del paquete le corresponden, para los datos Float se multiplican por 10 de forma sucesiva los 3 componentes, mientras que para el Float como el primer byte indica los decimales, se multiplican por 10 de forma sucesiva los 3 restantes y finalmente se suma el primer dato dividido entre 10.

Detección de errores en maestro y esclavo, de operaciones de ejecución planeación

La operación de detecciones errores se ejecutan en paralelo a las funciones de operación el proceso mediante conteo del tiempo de operación. En el Graficet de las operaciones de planeación se observa la distribución de estas operaciones junto con las cartas de función de los módulos, Figura 64. Tanto el módulo maestro como el esclavo pueden realizar la corroboración del tiempo, depende de las operaciones.

Al ser una unidad de control modular, la detección y el tratamiento de errores es inmediato en las operaciones de paro por error, de identificación de errores y de mantenimiento pues el maestro realiza la planeación y los módulos ejecutan tareas de forma independiente. En caso de fallo el maestro puede identificar el momento del error, detener el proceso y solicitar a los módulos un reposicionamiento seguro; en caso de fallar el maestro los módulos terminan sus tareas las cuales por condiciones de operación se permite ejecutar funciones que no tengan conflicto físicamente por el análisis de las operaciones del proceso. Esta ventaja es clara ante un sistema secuencial centralizado en un dispositivo pues para la identificación de error se debe reiniciar el dispositivo y leer los registros en memoria además de su reemplazo es costoso, mientras que en el sistema modular se identifica al momento y la sustitución es individual.

A modo de referencia, el microcontrolador PIC16F887 cuenta con tres Timer que pueden usarse para generar interrupción por desborden pero por experiencia de usuario el uso recomendado es:

- Timer 0, reloj interno para el WDT.
- Timer 1, reloj interno o externo en modo comparador y de interrupción.
- Timer 2, reloj externo en modo comparador y PWM.

El procedimiento de detección de error ocurre durante ejecución de tareas, como se muestra en la Tabla 51, el maestro espera de forma secuencial el fin de las tareas solicitadas y a su vez espera dos interrupciones para detección de error: por desborde de Timer para el tiempo esperado y por cambio de estado en línea de error; mientras que para el módulo esclavo la tarea se ejecuta de forma secuencial y se inicia el conteo de tiempo de operación para esperar error por desborde de Timer. De esta forma la unidad de control puede detectar el error de forma segura.

Otras funciones como la solicitud de tareas, operaciones de monitoreo o de pruebas y procesos en tiempo de ejecución son definidas de forma particular para el cap. 4.4 para la interfaz y el módulo maestro; así el diseño de la comunicación establece las bases de la UC. Los bloques funcionales de la estructura de comunicación se desarrollan en los siguientes diagramas con base en los puntos anteriores: En las Figuras 73 y 74 se desarrolla operaciones de reconstrucción y detección errores, en las Figuras 75, 76 y 77 se desarrolla funciones de comunicación para RS232 e I2C.

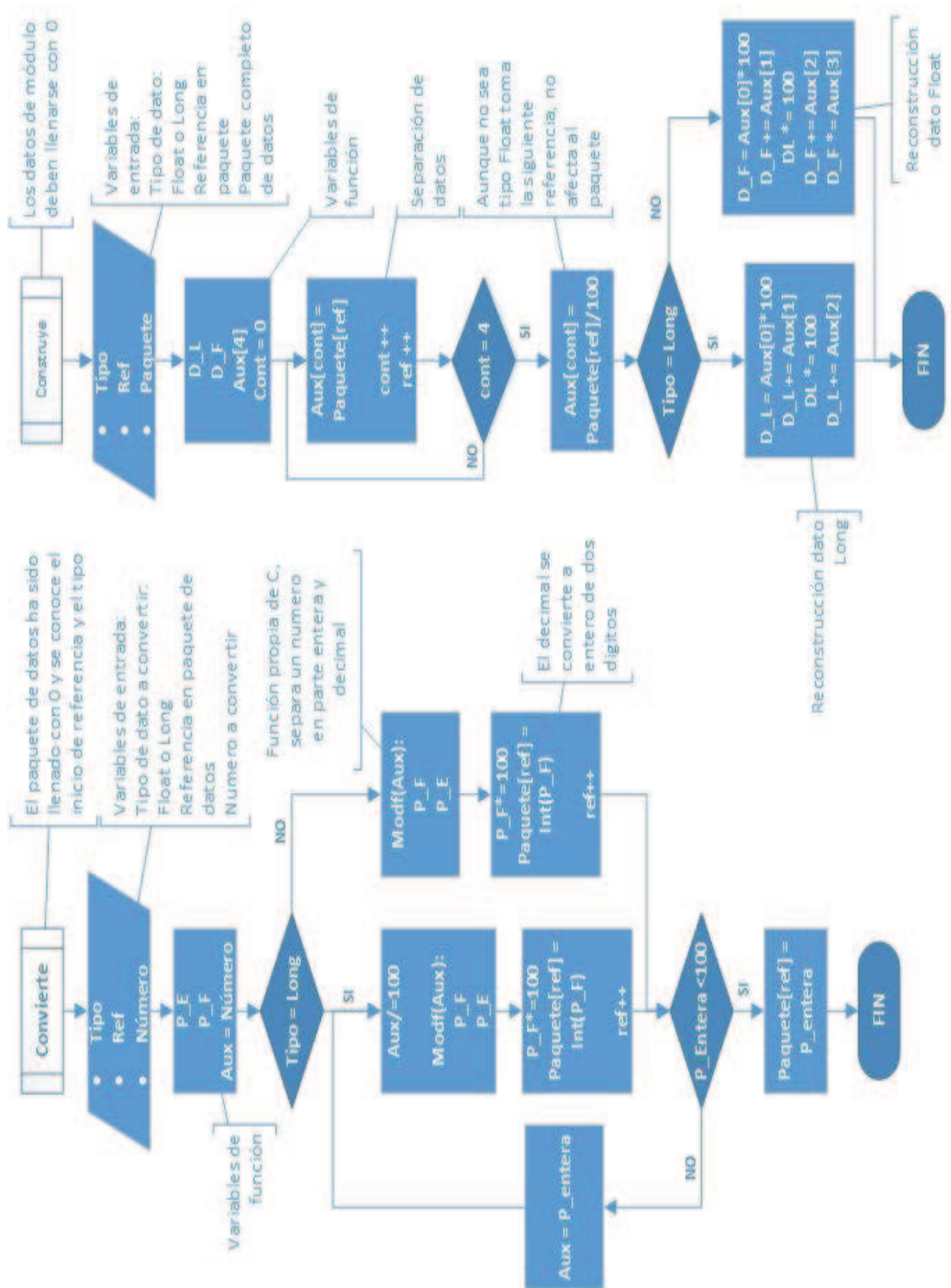


Figura 73. Conversión de datos, operaciones de comunicación

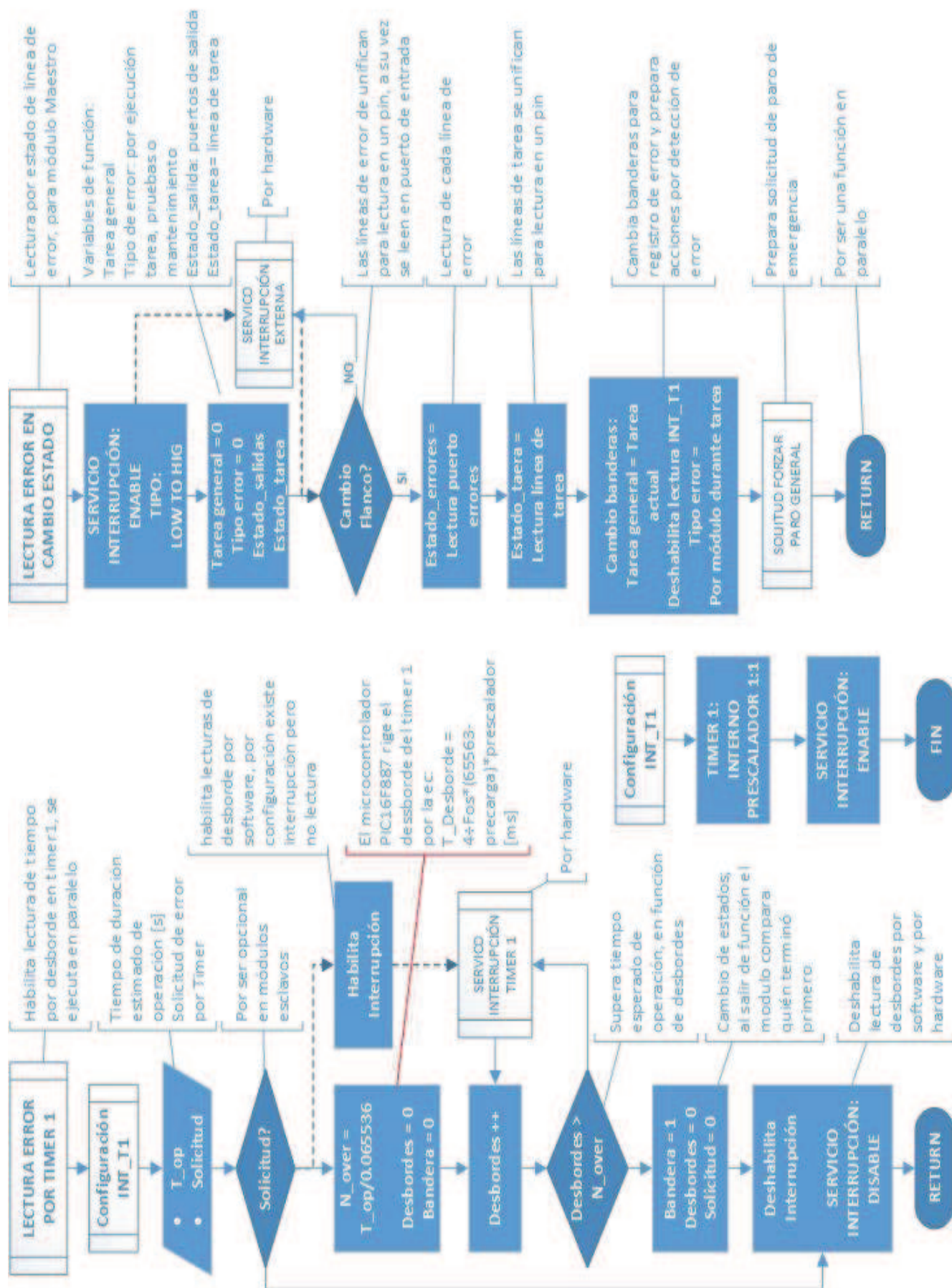


Figura 74. Identificación de errores, operaciones de comunicación

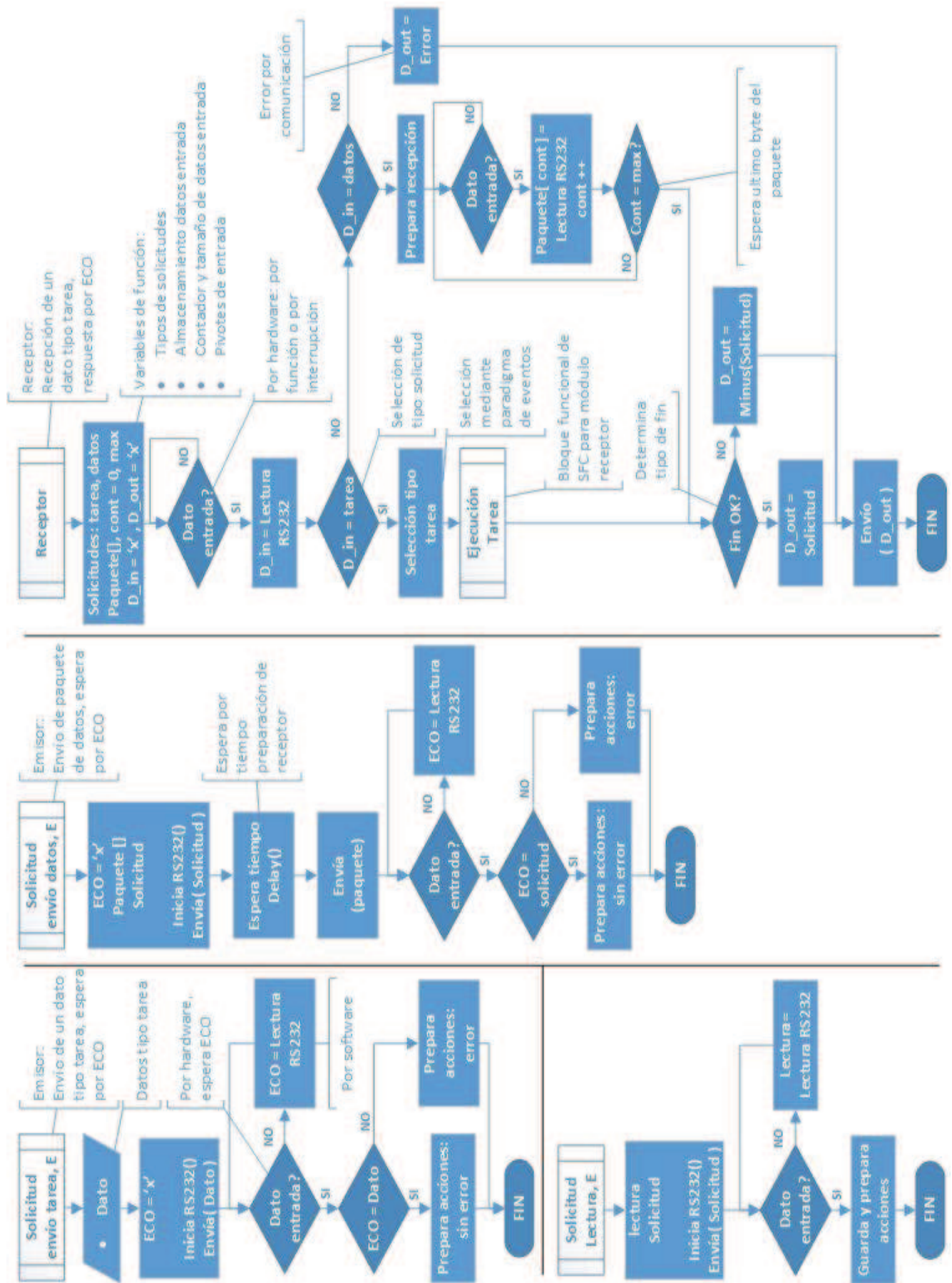


Figura 75. Funciones de emisor-receptor para RS232, estructura comunicación

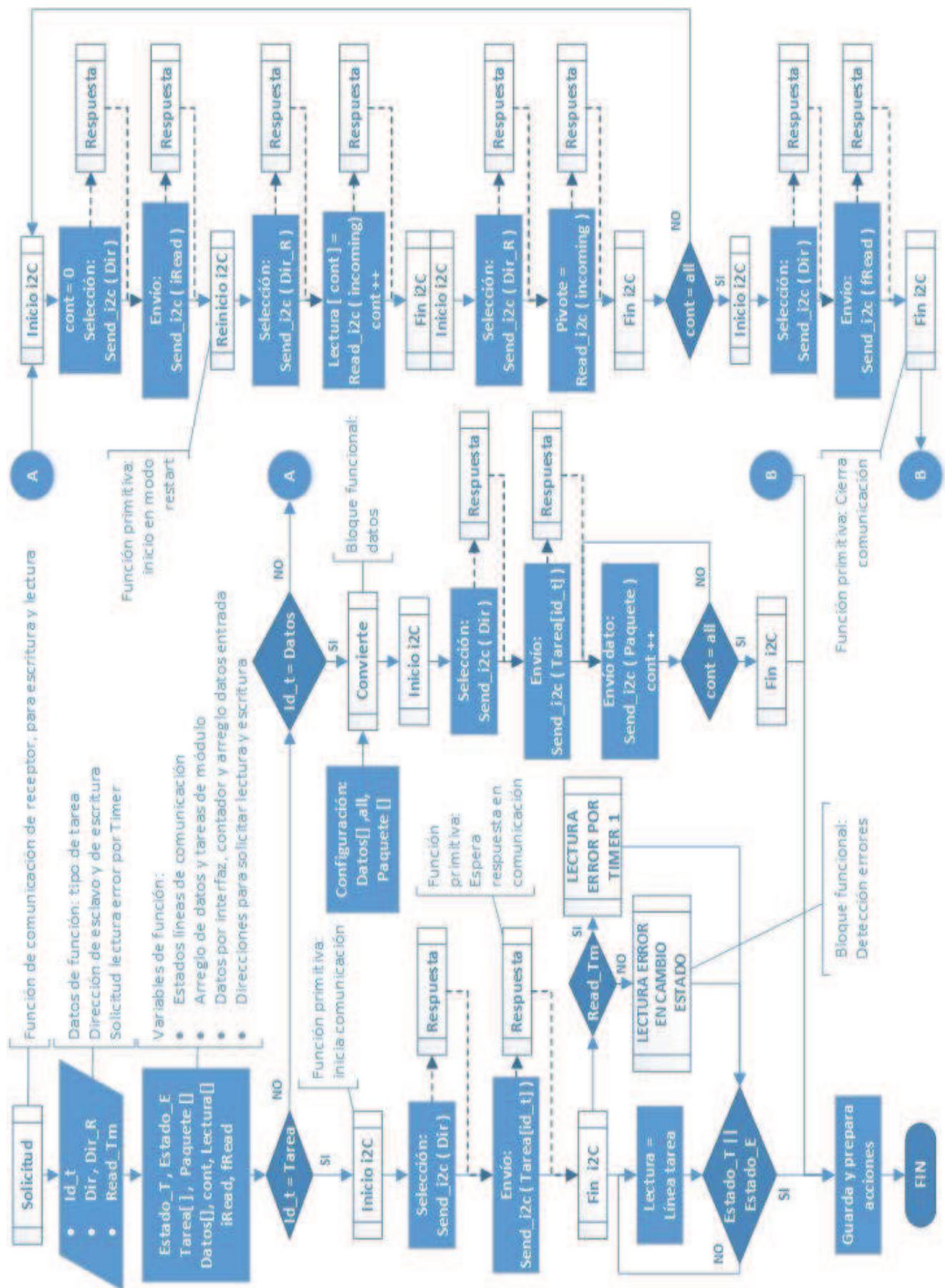


Figura 76. Función de emisor para I2C, estructura comunicación

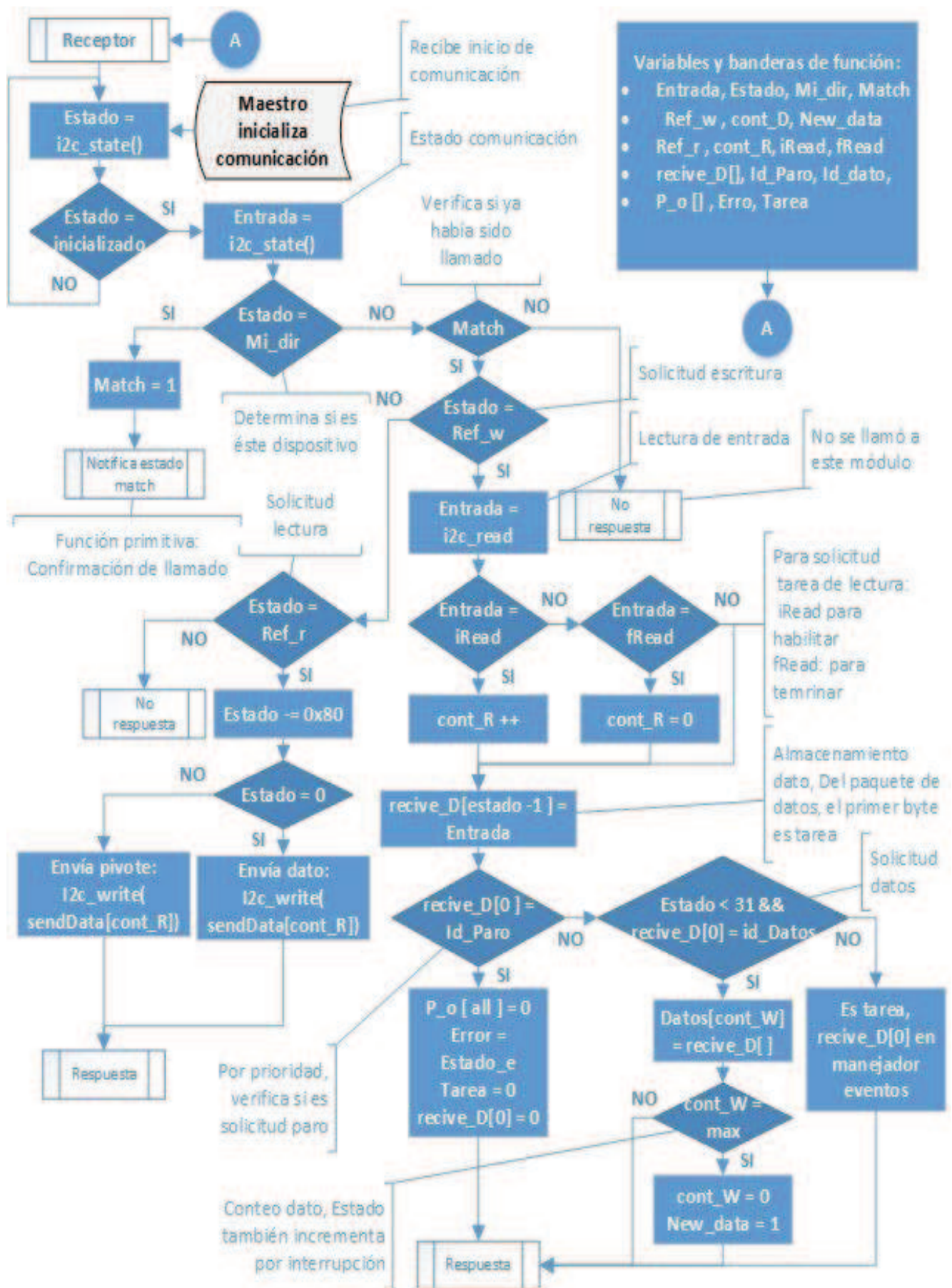
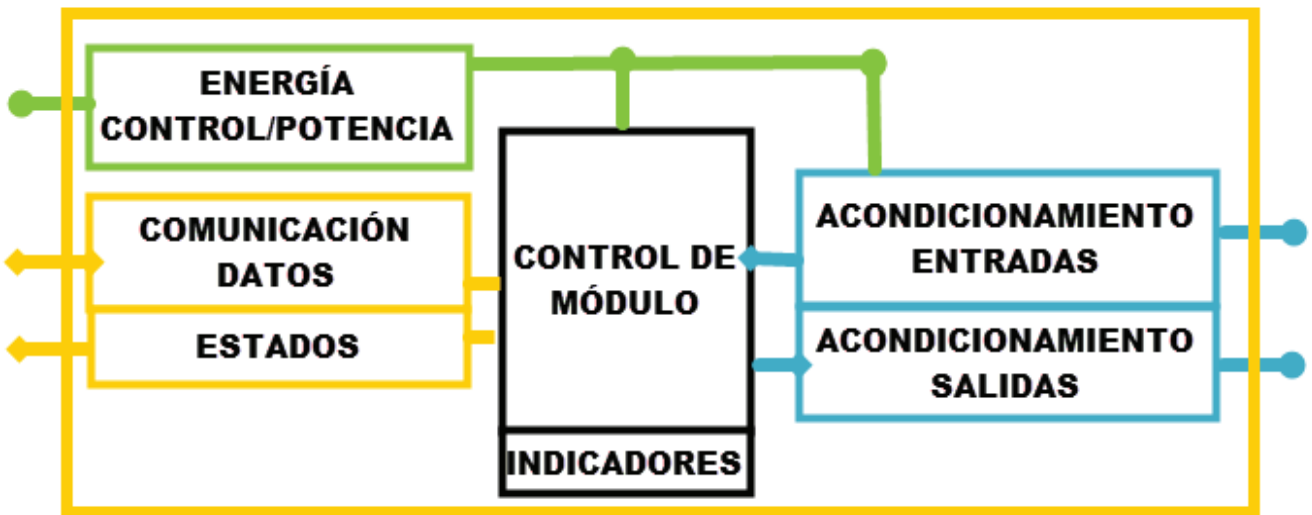


Figura 77. Función de receptor para I2C, estructura comunicación

De la estructura de comunicación definidas ya las funciones y operaciones, y establecidas las operaciones de los procesos por configuración, planeación y ejecución, se establecen estructuras genéricas para el sistema de control. En la Figura 78 se muestra para el módulo maestro y de los sistemas físicos en amarillo, por las operaciones del proceso y de comunicación requieren:

- Amarillo, comunicación entre módulos de planeación y ejecución.
- Rojo, Comunicación entre módulo de planeación y configuración.
- Verde, fuente de alimentación.
- Azul, acondicionamiento de entradas y salidas.
- Negro, etapa de control, microcontrolador.

Módulo esclavo



Módulo maestro

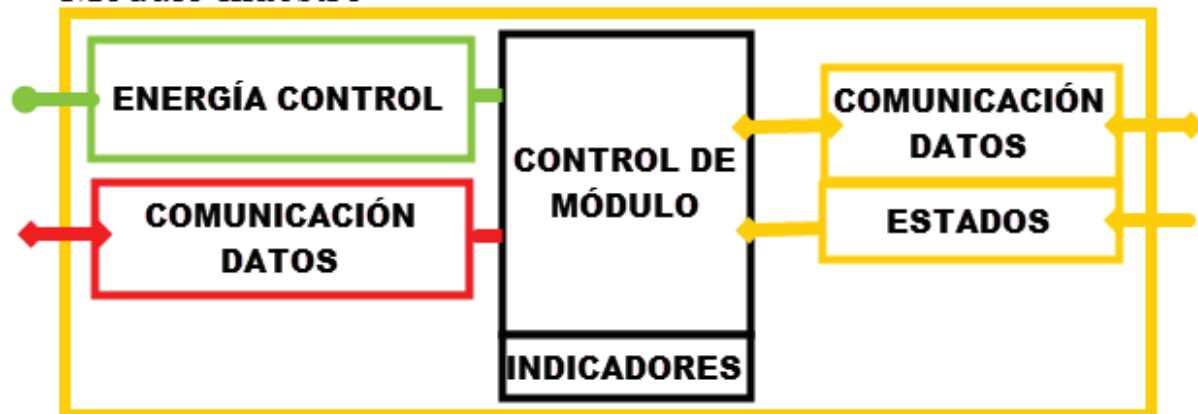


Figura 78. Estructura de módulos de operaciones de planeación y ejecución

En la Figura 79 se muestra la estructura de los módulos para las operaciones de configuración, en rojo, las operaciones no interactúan directamente con el sistema físico; estos módulos requieren:

- Rojo, comunicación entre módulos de configuración y de planeación.
- Gris, acondicionamiento de datos, por dispositivos pasivos (pantallas, memorias, etc.).
- Verde, fuente de alimentación.

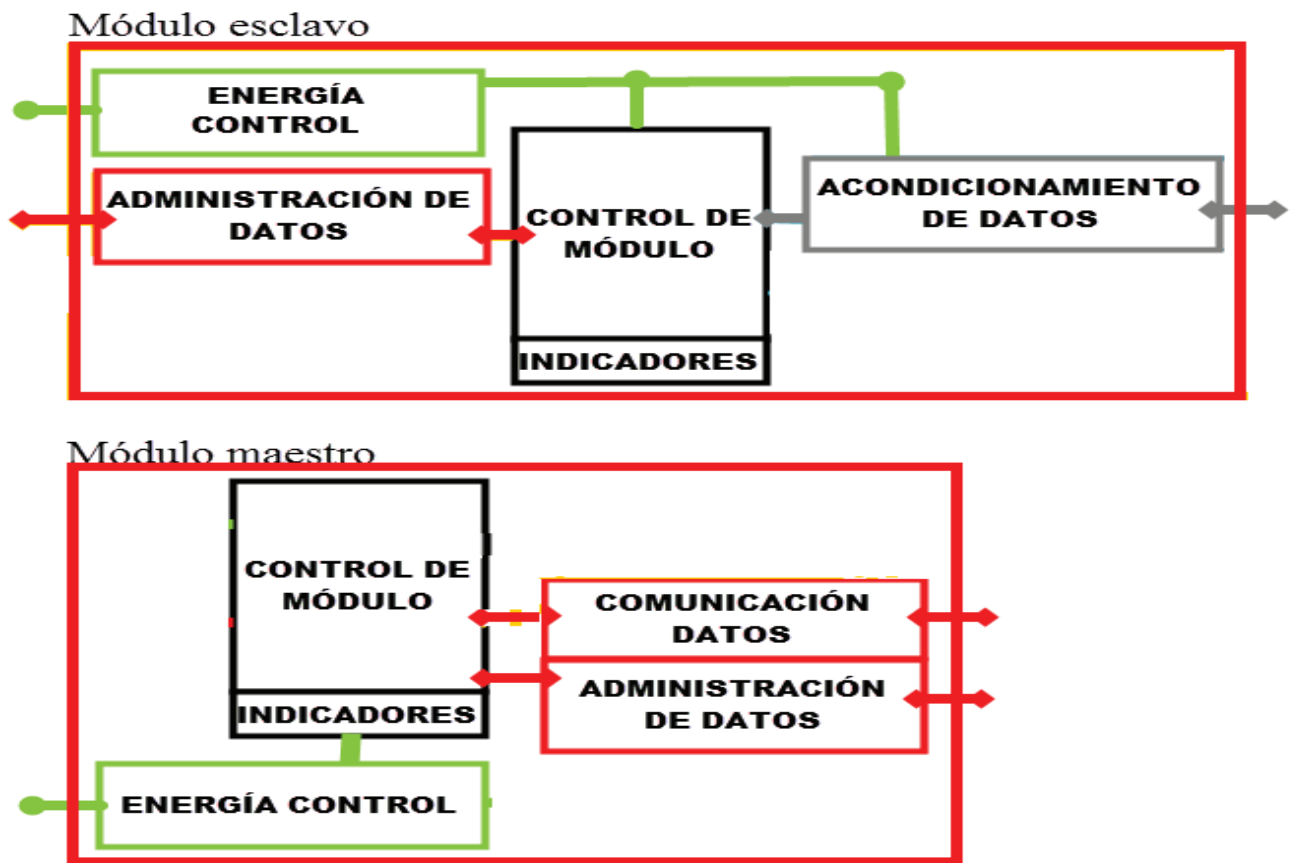


Figura 79. Estructura de módulos de operaciones de configuración

En la Figura 80 se muestra la estructura del sistema modular considerando las operaciones de comunicación y la distribución de los módulos en las operaciones del proceso junto con el modelo Bitbus. El desarrollo de cada módulo debe ser a partir de esta estructura de comunicación, pasando de las SFC del análisis del proceso a bloques funcionales pues ya están consideradas todas las condiciones y estados de operación, mientras que con las especificaciones, limitaciones y los pines de entrada y salida desarrolladas se definen los periféricos en el microcontrolador.

De las funciones de operaciones de proceso se definieron un grupo de Grafcet encapsulados que representan tiempos de operación, cap. 4.1.6, no se desarrolló el Grafcet principal debido a la implementación de mi controladores; ahora que se ha definido la estructura de comunicación se define para la UC un conjunto de datos para la configuración y la planeación de las funciones de operación que determinan la evolución del proceso y sus condiciones como *set de instrucciones*. La definición en el diseño de la estructura de comunicación permite darle independencia en cuanto a dispositivos de implementación. Las consideraciones para el establecimiento del set son:

- Se generan a partir de las operaciones de configuración, siguiendo la interfaz de usuario.
- Se representa por líneas de operación para indicar la duración de solicitud de operaciones.
- Todas las operaciones tiene una instrucción de referencia.
- En toda la línea de comunicación se mantiene la misma instrucción para una misma operación
- Solamente el modo de programador puede acceder a todas las instrucciones, por lo que la interfaz de usuario solo se permite la llamada de operaciones predefinidas.
- Las operaciones en los módulos de ejecución siguen la referencia T#, por lo que tiene un significado diferente dependiendo del receptor.

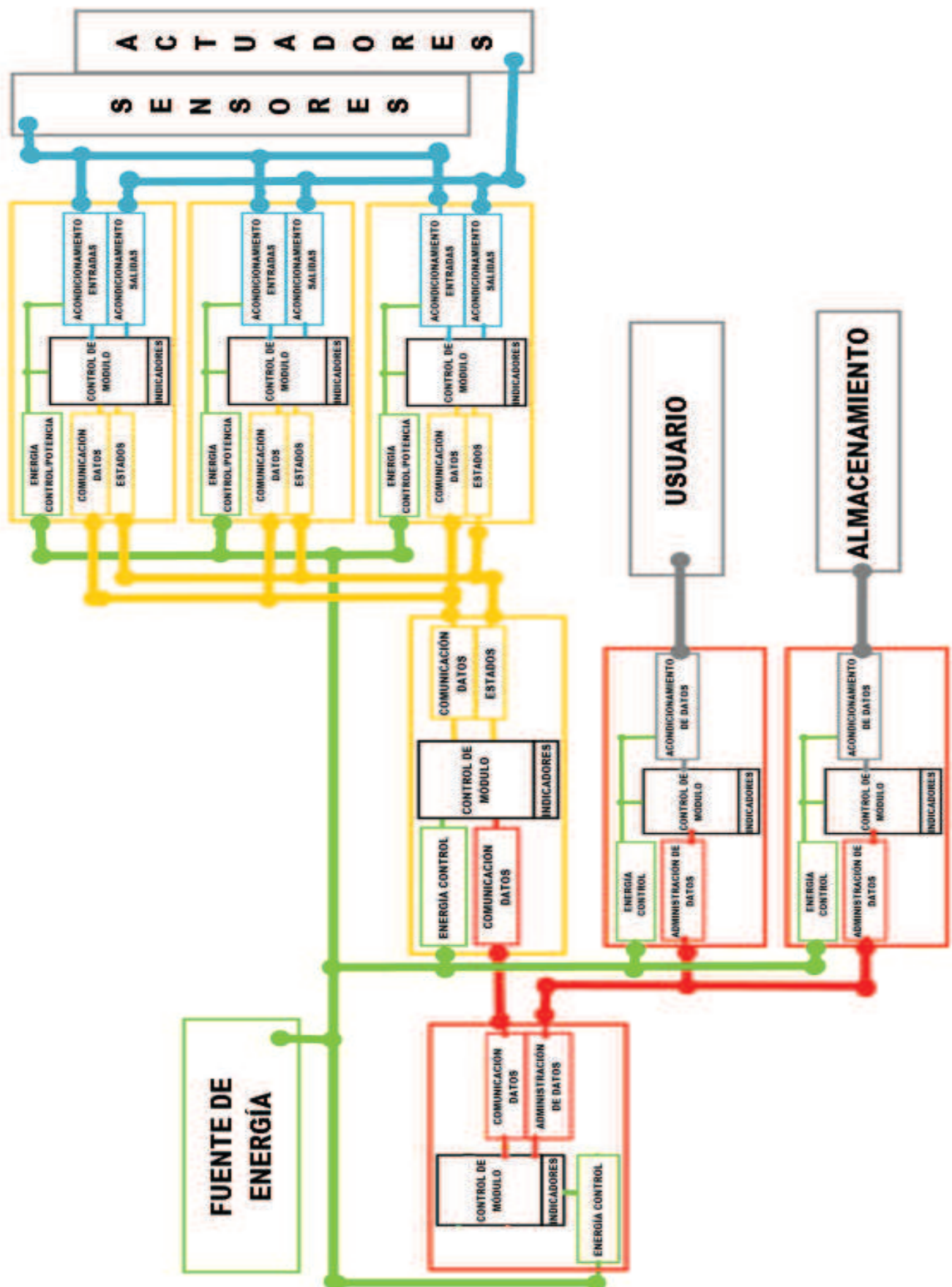


Figura 80. Estructura del sistema de control modular

En la Tabla 53 se muestra de forma general el set de instrucciones a partir de la jerarquía de funciones de operación y de componentes, las operaciones de planeación y configuración de las Tablas 21, 26, 27 y 51, y de las Figuras 45, 62 y 70. El módulo maestro traduce la configuración en ejecución mediante la relación de operaciones, para los módulos específicos de la interfaz no se desarrollan instrucciones específicas al considerarlo un módulo único para generalizar las funciones.

Tabla 53. Set de instrucciones de la estructura de comunicación			
Operación	Módulo: sistema	Instrucción	Función de operación
Ejecución	M1 – Estructura: Carrusel	T0	Mover inverso
		T1	Mover completo
		T2	Mover a ½
	M1 – Estructura: Orientación contenedor	T3	Orienta 1B
		T4	Orienta 1C, vaciado
		T5	Orienta 1C, mezclado
	M1 – Estructura: Posición batidora	T6	Mover a L
		T7	Mover a H
	M2 – Mezclado: Batidora	T1	Revolver, control
		T2	Batir
	M3 – Efluente: Lodo	T1	Dosificar
	M4 – Floculante: Polímero	T1	Llenar vías
		T2	Vaciar vías
		T3	Dosificar
		T4	Bomba 1
		T5	Bomba 2
		T6	Bomba 3
		T7	Bomba 4
	Módulos en general	TP	Forzar paro
		TD	Datos
		TC	Conexión
		TL	Lectura
Configuración	Conexión	C	Corroboración conexión
		E	Solicita estado error conexión
	Configuración parámetros	D	Envío datos
		Datos[]	Paquete de datos
	Secuencia	S	Inicio/Fin operaciones de secuencia
		Y	Reportes: Fin iteración
	Monitoreo	N	Operaciones de monitoreo
	Monitoreo: Motores	M	Inicio/Fin operaciones de motores
		C	Carrusel a ½
		I	Carrusel a completo
		V	Orienta 1B
		E	Orienta 1C
		R	Posición batidora
	Monitoreo: Prueba estructura	F	Prueba movimiento
	Monitoreo: Lectura estados	E	Inicio/Fin operaciones de lectura datos
		S	Sensores
		B	Banderas

	Monitoreo: Prueba Mezclado	P	Prueba control velocidad
	Herramientas	H	Inicio/Fin operaciones de herramientas
		L	Lavado
		V	Limpia vías, guardar floculante
		B	Limpia vías sin guardar floculante
		Z	Reinicio de estados módulos

En las siguientes Tablas, de 54-56 se desarrolla la evolución de las principales operaciones del proceso, se recurren a los diagramas de tiempos de operación, Figuras 49 a 55. De color verde se ilustran datos de envío, de rojo datos de recepción, de azul la línea de tarea y de gris operaciones que no afectan la comunicación pero es importante indicirlas.

Tabla 54. Instrucciones de operaciones: Conexión					
Interfaz	Maestro	Estructura	Mezclado	Efluente	Floculante
C	C				
	TC	TC			
	Id_M1 / 0	Id_M1 / 0			
	TC		TC		
	Id_M2 / 0		Id_M2 / 0		
	TC			TC	
	Id_M3 / 0			Id_M3 / 0	
	TC				TC
	Id_M4 / 0				Id_M4 / 0
C / c	C / c				
E	E				
E / Id_M 0	E / Id_M 0				

Tabla 55. Instrucciones de operaciones: Configuración parámetros					
Interfaz	Maestro	Estructura	Mezclado	Efluente	Floculante
D	D				
Datos[]	Datos[]				
	M1 – TD	M1 – TD			
		Pin Tarea H			
	Datos[]	Datos[]			
	Pin Tarea L	Pin Tarea L			
	M2 – TD		M2 – TD		
			Pin Tarea H		
	Datos[]		Datos[]		
	Pin Tarea L		Pin Tarea L		
	M3 – TD			M3 – TD	
				Pin Tarea H	
	Datos[]			Datos[]	
	Pin Tarea L			Pin Tarea L	
	M4 – TD				M4 – TD
					Pin Tarea H
	Datos[]				Datos[]
	Pin Tarea L				Pin Tarea L
D / d	D / d				

Tabla 56. Instrucciones de operaciones: Secuencia					
Interfaz	Maestro	Estructura	Mezclado	Efluente	Floculante
S	S				
TO-0	Envío de datos	Envío de datos			
TOP-1	M3 – T1			M3 – T1	
				Pin tarea H	
	Pin tarea L			Pin tarea L	
TOP-2	M1 – T2	M1 – T2			
		Pin tarea H			
	Pin tarea L	Pin tarea L			
	M1 – T5	M1 – T5			
		Pin tarea H			
	Pin tarea L	Pin tarea L			
	M1- T1	M1 – T1			
		Pin tarea H			
	Pin tarea L	Pin tarea L			
TOP-3	M1 – T6	M1 – T6			
	M2 – T2		M2 – T2		
	M4 – T3				M4 – T3
	M4 – T[4-7]				M4– T[4-7]
		Pin tarea H	Pin tarea H		Pin tarea H
	Pin tarea L	Pin tarea L	Pin tarea L		Pin tarea L
TOP-4	M1 – T2		M2 – T1		
	M1 – T3	M1 – T3			
		Pin tarea H	Pin tarea H		
	Pin tarea L	Pin tarea L	Pin tarea L		
	M1 – T7	M1 – T7			
		Pin tarea H			
	Pin tarea L	Pin tarea L			
	M1- T4	M1 – T4			
		Pin tarea H			
	Pin tarea L	Pin tarea L			
Y	Y				
S / s	S / s				

4.2.3 Estructura de funciones de operación en la UC

Del cap. 3.5 se estableció el concepto de la unidad de control, en el desarrollo del diseño se ha analizado las operaciones del proceso y las operaciones de comunicación de forma independiente y genérica para que la implementación en las Fases de trabajo no requiera estructuras particulares. En el cap. 4.1 se estableció las funciones de operación a nivel de proceso, éstas son las que se implementan en los módulos como bloques funcionales y de forma particular.

La estructura de comunicación permite establecer la evolución de las operaciones del proceso, pero para atender cada operación se debe definir las operaciones de la UC en los módulos. Definirlos de forma general, como la comunicación, permite darle flexibilidad durante la implementación, permite encontrar y corregir errores de código de forma general e identificar la evolución del proceso.

La estructura de operaciones para este trabajo se centra en la implantación de paradigmas que dan un orden y un nivel de ejecución de las tareas durante el proceso. En el cap. 2.4.1 se describió a los paradigmas deliberativo y orientado a eventos, estos se relacionan por las operaciones regentes al módulo para la ejecución de las operaciones, manteniendo así la independencia con las operaciones del proceso. Se describen las características importantes para su implantación, además de estos se requiere definir las variables pertenecientes a las operaciones de la UC, que son diferentes a las variables del proceso, Tabla 28.

Paradigma Deliberativo

La automatización de un proceso implica la transición del proceso por operador a uno por máquina, en este sentido el paradigma sigue la estructura de desarrollo de actividades de forma secuencial: Sensar, planear y actuar. Al ser un sistema basado en microcontroladores su aplicación permite ordenar los comandos, la ejecución de funciones, la evolución en las líneas de código genéricas y en operaciones de mantenimiento o el escalamiento del sistema; aunque solo permite ejecución de una tarea a la vez se compensa al implementarse en un sistema modular, por otra parte el uso de funciones de interrupción permite darle la flexibilidad en cuanto a detección de errores por funcionamiento y reducción de código en cuanto a operaciones de la ejecución. El uso del paradigma se asocia a las siguientes operaciones:

- **Sensar:** lectura de sensores para función específicos; los valores de los puertos de entrada se almacenan en variables del proceso, de forma que únicamente en este paso sea la lectura para asegurar que la evolución del programa sea uniforme y por tiempos de operación en ejecución de código.
- **Planear:** de acuerdo a las operaciones y estados lógicos para la función, se realizan cambios o actualizan contadores o banderas o variables con forma a los estados de entrada, para definir los valores de actuación, de esta forma toda la parte de paliación se realiza por operaciones lógicas.
- **Actuar:** las acciones de actuación son referentes a los actuadores de la función; a partir de los estados cambados en la planeación se actualizan los puertos de salida para encender, apagar, habilitar, deshabilitar o modificar los valores de salida.

La ventaja de usar esa separación en el desarrollo de las funciones es la de tener un mejor control sobre los actuadores y los eventos por ciclo de operación; mediante los servicios de interrupción se usan estados de prioridad para terminar el ciclo de función o detener la actuación de forma rápida y detectar fallos.

Paradigma orientado a eventos

En cuanto a los sistemas para las funciones de las operaciones del proceso, por la estructura de comunicación los módulos debe esperar la instrucción de solicitud para determinar qué tipo de tarea ejecutar, este tipo de selección puede ser estructurado de muchas formas bajo diferentes paradigmas o métodos, para este trabajo al considerar la comunicación I2C y RS232 se debe establecer una estructura general que organice las solicitudes de tarea y de orden y prioridad a las funciones además debe ser independiente a las operaciones de comunicación y de ejecución de funciones, en este sentido el paradigma orientado a eventos permite manejar la selección de los datos de comunicación mediante eventos que pueden ser reasentados por entradas, para el caso de líneas de error y tarea, o por datos de entrada por comunicación, y al igual que el paradigma deliberativo, solo permite

manejar un evento a la vez sin permitir secuencias de funciones. El uso del paradigma se asocia con las siguientes definiciones:

- **Manejador de eventos, ME:** un evento es una entrada bien definida que durante ejecución de alguna operación modifica a dicha función independientemente de las operaciones en ejecución. De esta forma se establece niveles de manejadores, siendo el principal el de que maneja las solicitudes de operaciones del módulo, mientras que por cada operación ejecutada se define un manejador de evento para operaciones relacionadas al módulo que son independientes de las que ejecuta la función.
- **Eventos de datos:** Representa al grupo de solicitudes que modifican los datos de configuración o de memoria.
- **Eventos de operaciones:** Representa al grupo de solicitudes de funciones de operación, en este grupo puede configurarse la ejecución de servicios de interrupción para detectar errores el estado de la función o detener su función.
- **Eventos de sistema:** Representa al grupo de operaciones de módulo, estas son encargadas de las solicitudes de maestro para mantenimiento, por interfaz de usuario no se puede acceder a estas funciones pues no son definidas en el set de instrucciones de la estructura de comunicación, son propias de cada módulo.
- **Eventos en funciones de operación:** Representa a los eventos dentro de una función de operación, se encargan únicamente de las operaciones de módulo como solicitudes de paro o configuración, estos eventos son generados por servicios de interrupción.

A manera de comparación, el paradigma orientado a eventos usa las operaciones de sensado y planeación de tal forma que genera manejadores de eventos permite generar variables globales en la estructura de operaciones de la UC.

Variables de operación

La definición de variables para operación en la UC permite estructurar los elementos mínimos para el atender las operaciones de cada módulo, como se ha mencionado se quiere una independencia entre las estructuras de comunicación, de operación de la UC y del proceso para generar una UC que permita la definición de más módulos para el proceso en las siguientes etapas del proyecto, y en general para cualquier aplicación. La UC requiere seis grupos principales de variables para la configuración y ejecución de los paradigmas, estos son:

MACROS

Se usan para referencia en configuración de puertos, no tienen un tipo definido de datos, estos son:

- pinSM : número de salidas digitales
- pinEM : número de entradas digitales
- pinPM : número de PWM
- pinAM : número de analógicos

CONSTANTES

Son usadas para referencias de valores de PWM, para operaciones matemáticas, valores de tiempos o referencias numéricas para contadores o comparaciones, son propias de cada módulo.

VARIABLES DE LOOP DE TRABAJO

Hacen referencia a los manejadores de eventos para las solicitudes de operación y de función, se usa el identificador w.

W [2] = {0}

- w [0] loop de trabajo nivel 1, manejador de eventos para operaciones.
- W [1] loop de trabajo nivel 2, ciclos de funciones de operación.
- W [2] loop de trabajo nivel 3, ciclos de funciones específicas dentro de operaciones.

VARIABLES GLOBALES DE PROPOSITO GENERAL

Se declaran en forma de arreglos para generalizar las variables y depende solo de su referencia en el arreglo, se separan en cuatro tipos: las variables para el manejador de eventos principal ME, las variables para los eventos de funciones MF, variables de propósito general vA para operaciones de funciones deliberativas y variables tipo booleanas para las operaciones de sensado.

- Long v1 [10] = {0}: Variables de contador de eventos, nivel1, para entradas de sensores o datos que generen evento en el manejador de eventos ME.
- Long v2 [10] = {0}: Variables de contador de eventos, nivel2, para entradas de señales que generen evento en una función, en el manejador de funciones MF.
- Long vA [10] = {0}: Variables de propósito general.
- booleanas para estados de sensores:
 - senED: para identificar cambios entrada digital que genera evento.
 - senEA: para identificar cambios entrada analógica que genera evento.

DECLARACIÓN DE VARIABLES PINES

Hacen referencia a los pines del módulo, se generan de forma de arreglos para leerlos o modificarlos en pocas líneas de código. Son de cuatro tipos, entradas digitales, entradas analógicas, salidas digitales y salidas analógicas (PWM). Usan las definiciones de los macros de número de pines para su tamaño. Estas variables son:

Entrada

- p_e [pinEM] = {direcciones de pines configurados como entradas digitales}.
- p_a [pinAM] = {direcciones de pines configurados como entradas analógicas}.

Salida

- p_s [pinSM] = {direcciones de pines configurados como salidas digitales}.
- p_PWM [pinPM] = {direcciones de pines configurados como salidas PWM}.

VARIABLES ASOCIADAS A LECTURA DE PINES DE ENTRADA

Son variables que se usan para el almacenamiento de la lectura tanto en el manejador de eventos como en el de funciones, de esta forma el proceso de Planeación no se necesita hacer lecturas, si no referencia a su valor leído; para su declaración usan los macros de numero de pines para su tamaño, en concordancia con los arreglos de pines. Estas variables son:

pinValD [pinEM] = {0}: lecturas de entradas de pines Digitales

- pinValD [0] referencia a p_e [0], pin fisco (dirección).
- pinValD [1] referencia a p_e [1], pin fisco (dirección).
- pinValD [2] referencia a p_e [2], pin fisco (dirección), etcétera.

pinValA [pinAM] = {0}: lectura de entradas de pines Analógicos

- pinValA [0] referencia a p_a [0], pin fisco (dirección).
- pinValA [1] referencia a p_a [1], pin fisco (dirección).
- pinValA [2] referencia a p_a [2], pin fisco (dirección), etcétera.

Definidas los paradigmas y las variables, la estructura de las operaciones de la UC se presenta en los siguientes bloques funcionales:

- Figura 81: estructura general del ME para solicitudes de operaciones.
- Figura 82: funciones de sensado y selector de eventos en el ME.
- Figura 83: estructura de funciones de operación y sensado.
- Figura 84: estructura de planeación.
- Figura 85: estructura de actuación.

Finalmente, en este punto se puede apreciar que se ha dividido el diseño en tres partes: la de análisis del proceso para obtención de funciones de operación, la de diseño de estructura de comunicación y de estatura de operaciones en la UC, estas se desarrollan de forma independiente, por lo que la implantación de las funciones de operación no modifican las estructuras antes mencionadas, además, al ser un sistema modular conviene generalizar todos los muros para las futuras etapas de desarrollo, en este sentido hasta ahora sea desarrollado las estructuras para un sistema de control modular general, en las Fases junto con el análisis de las operaciones del proceso se particulariza la aplicación de este sistema.

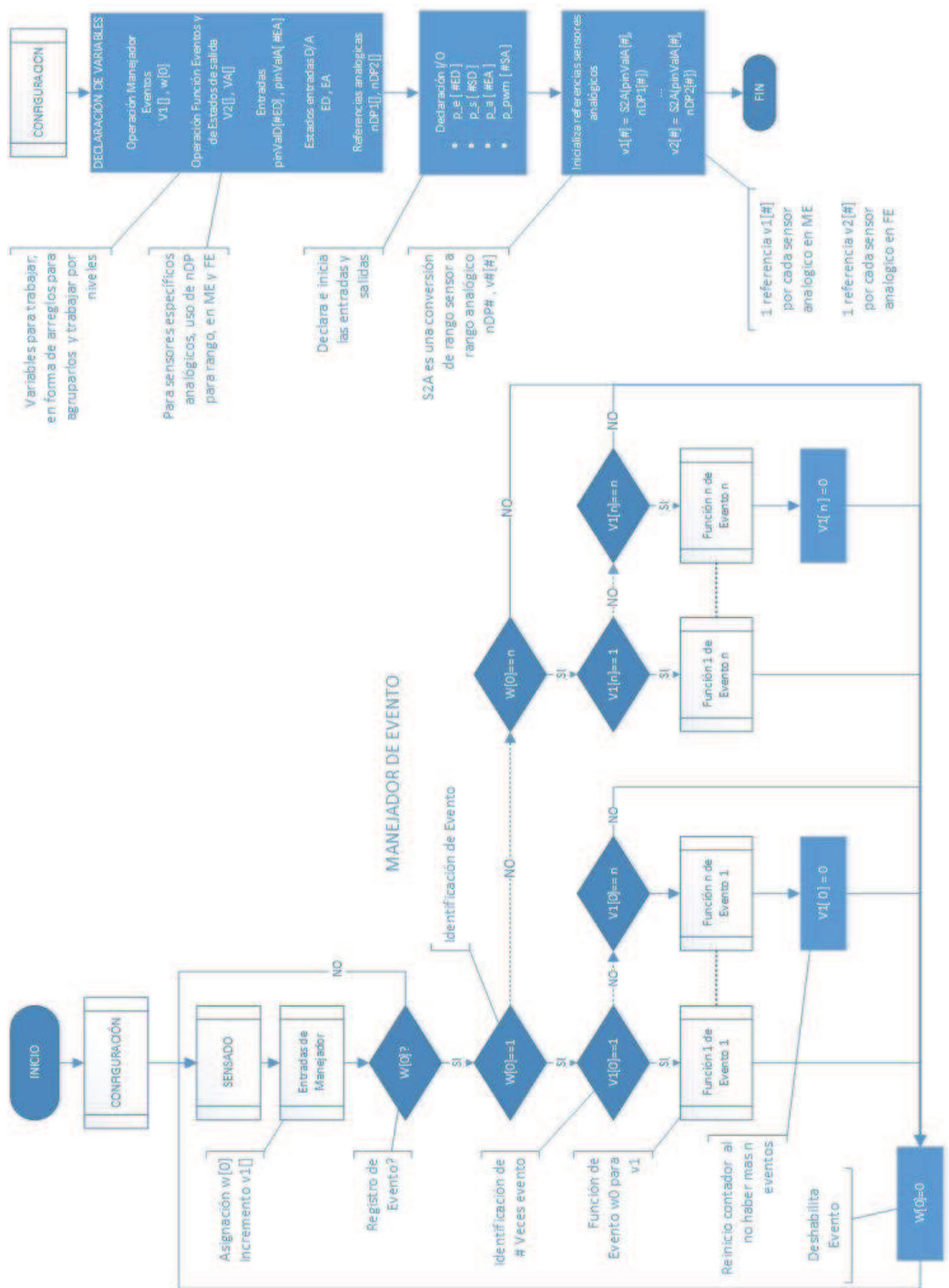


Figura 81. Función de operación en UC: manejador eventos

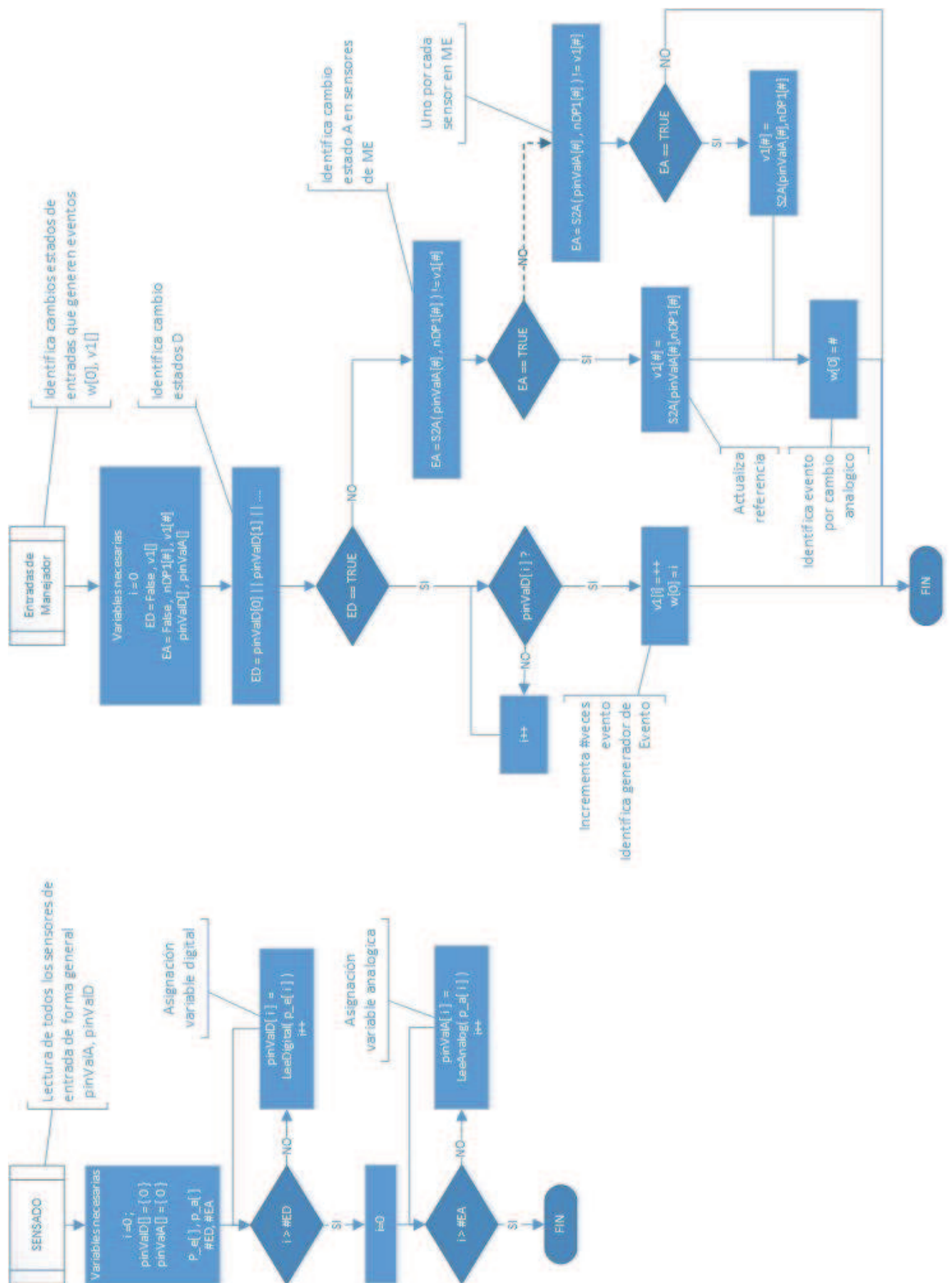


Figura 82. Función de operación en UC: Sensado para manejador eventos

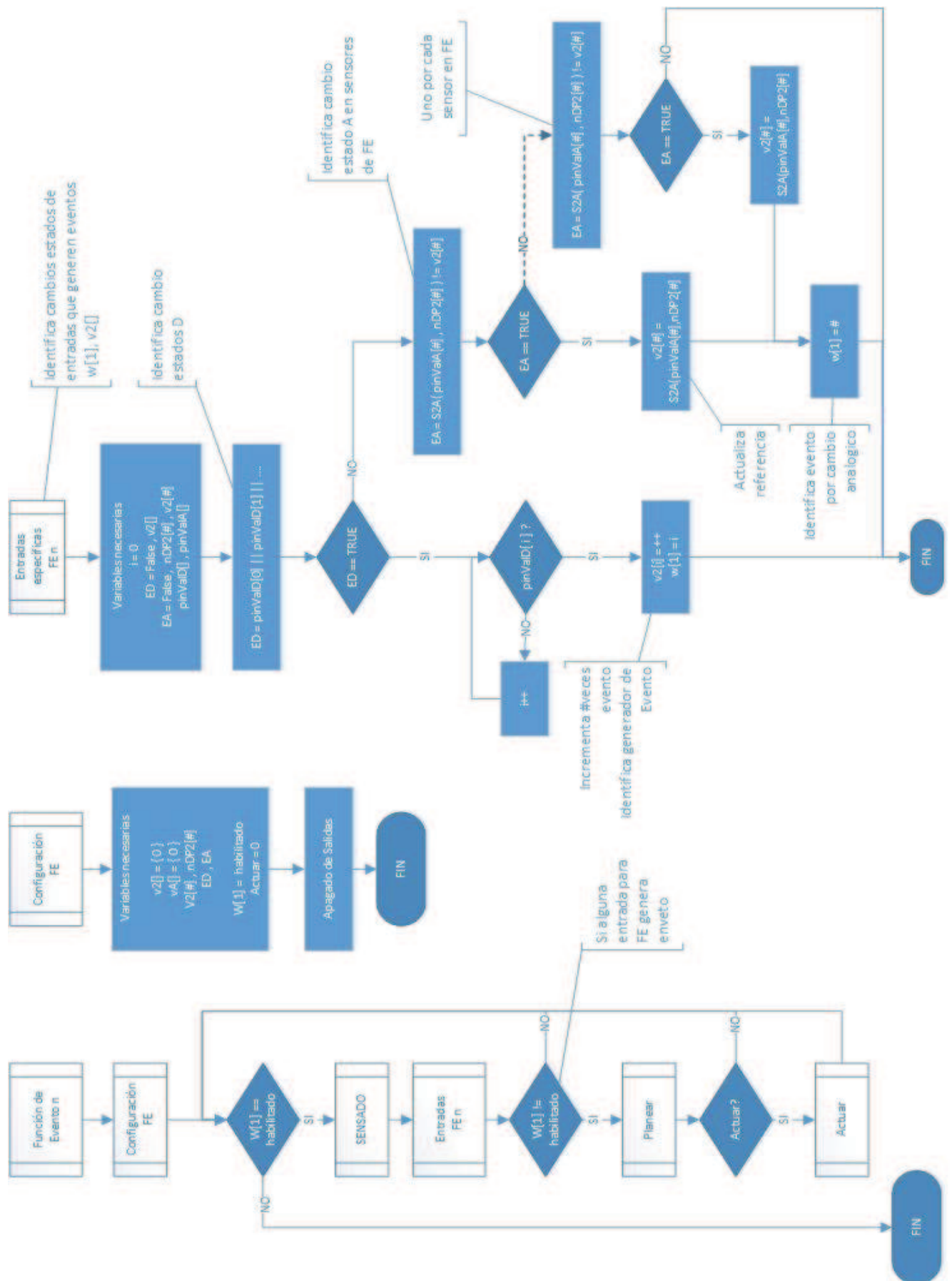


Figura 83. Función de operación en UC: Funciones de eventos y sensado

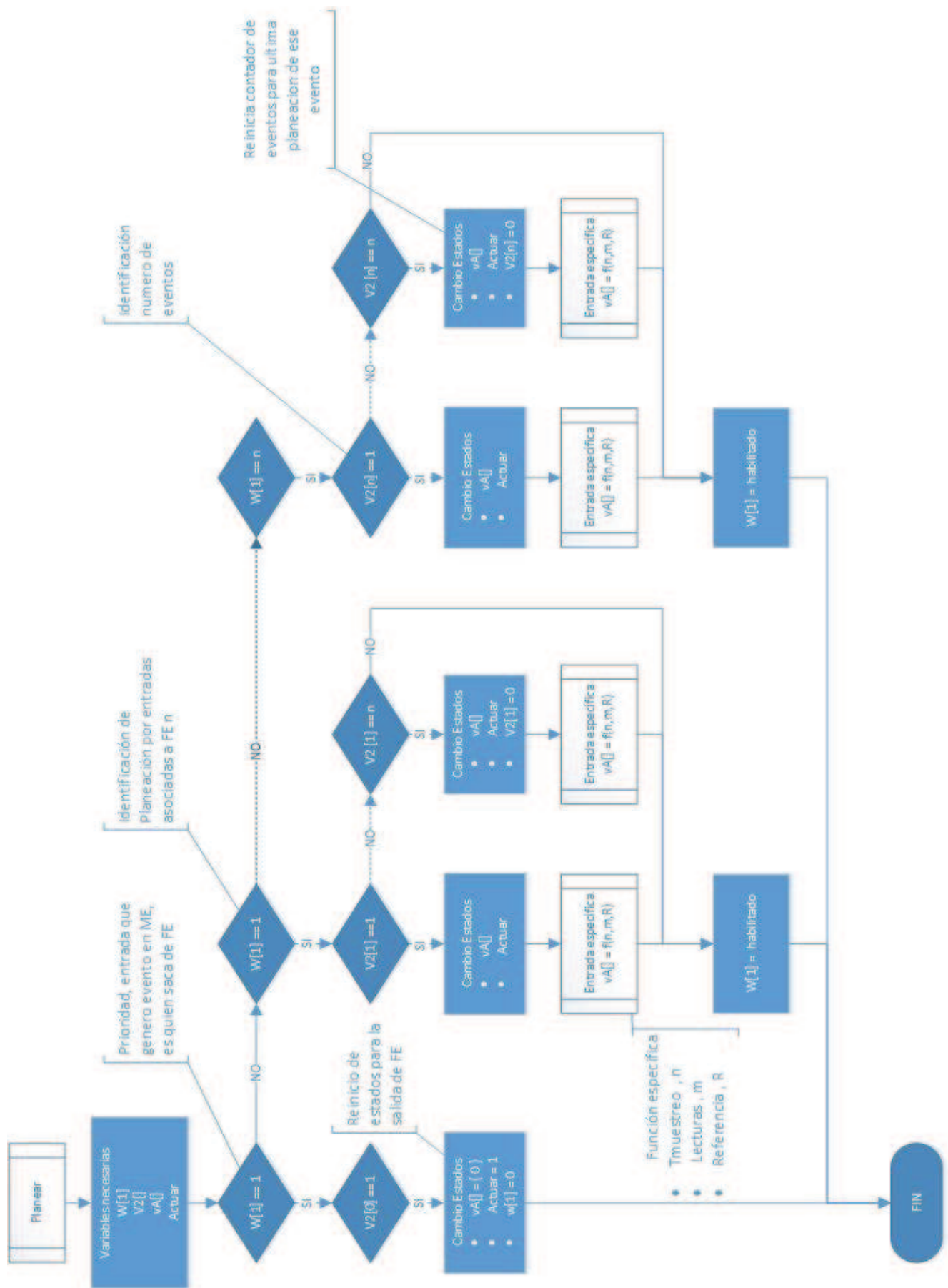


Figura 84. Función de operación en UC: Paradigma deliberativo, planeación

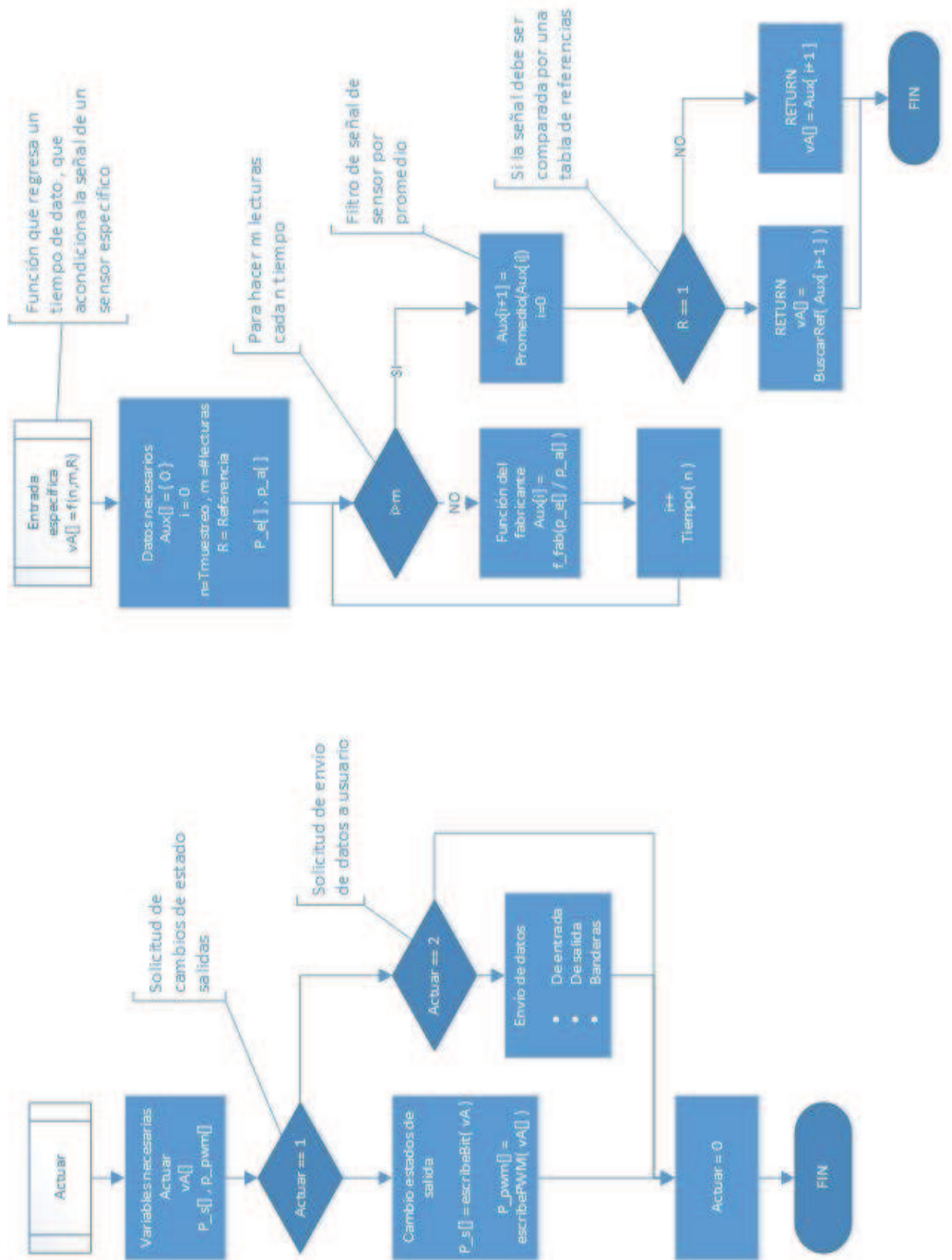


Figura 85. Función de operación en UC: Paradigma deliberativo, actuación

4.3 Fase 1: Control de estructura

Para el desarrollo del módulo de estructura se toman las SFC de las funciones de operación en relación a las entradas y salidas que requieren sus componentes para su implementación en el microcontrolador. Las consideraciones para el establecimiento del módulo se toman a partir del análisis de las funciones de operación, de comunicación y de la unidad de control, estas son:

- Del análisis del proceso, por los a componentes se definió al módulo estructura como un conjunto de sistemas encargados del posicionamiento del contenedor y de la batidora, Figura 36-38 y Tabla 17 y los sistemas carrusel, posicionamiento y orientación representan la mayor jerarquía de componentes, Figura 45.
- Sus funciones de operación del proceso están descritos por los SFC de las Figuras 56-57 junto con sus condiciones y estados de operación.
- En cuanto al módulo a partir de las funciones de operación se describieron el set de instrucciones y su evolución en las operaciones de planeación, Tabla 53.
- De sus componentes, se definieron los puertos de entrada y de salida con su tipo de acondicionamiento y control, Tabla 48.
- Por las funciones de operación en relación a los componentes, los actuadores y sensores no requieren una función específica para la ejecución de las tareas de sus sistemas.

4.3.1 Módulo de estructura

La definición del módulo de estructura a partir de su implantación en microcontrolador para su sistema electrónico y sus funciones de operación se desarrolla en cuatro partes:

1. Se definen la relación de funciones del proceso con los puertos de entrada y salida por componentes en el microcontrolador.
2. Se da estructura al módulo a partir del modelo genérico de la Figura 80, para esclavo.
3. Se establecen los bloques funcionales a partir de las SFC con variables asociadas a la estructura de funciones de la UC.
4. Se desarrolla el sistema electrónico con las características descritas en cap. 4.2.1.

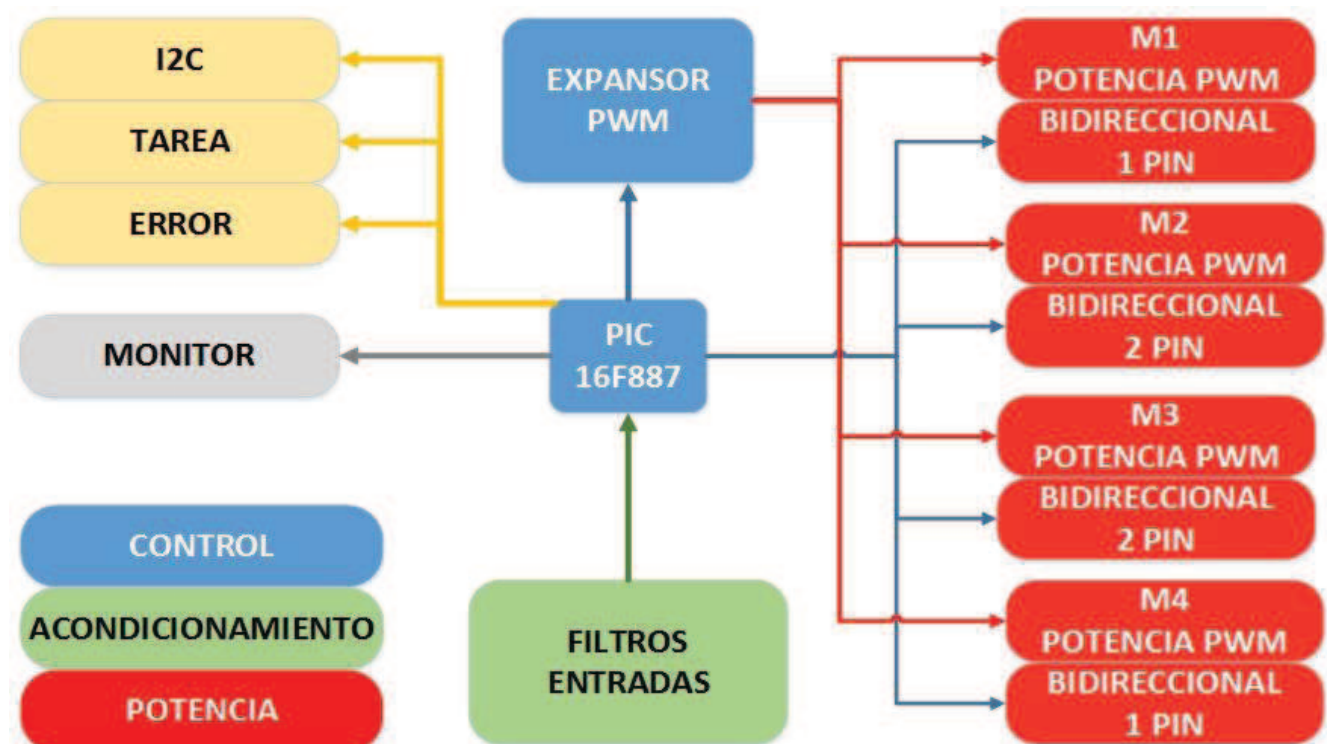
De las consideraciones para el desarrollo del módulo se tiene que la implementación en el microcontrolador se define por los puertos de I/O que requiere los componentes de sus sistemas y las funciones que del micro que permiten el desarrollo de las funciones de operación de las Tablas 46, 47 y 48 y de las Figuras 55, 56 y 57. En la Tabla 57 se agrupa el análisis realizado para este módulo y se determina la implantación y estructura del módulo en el microcontrolador.

En la Tabla 57 se muestra que los puertos del micro para PWM son referenciados por dos pines, esto se debe a que el módulo tiene cuatro actuadores y como especificación del microcontrolador sólo se cuentan con 2 salidas de PWM; la solución para este problema, al igual que en módulo dosificación polímero, es multiplexor la salida del PWM mediante 2 pines de selección E0 y E1, y 4 de habilitación D0-D4, el costo fue de esta solución incremento en 3 salidas de control sin afectar el funcionamiento del microcontrolador. Las consideraciones para esta solución fueron que el movimiento de los actuadores no requiere una función específica, por lo que puede usarse el mismo valor de PWM en caso de habilitar más de uno en tiempo de operación y que el uso del multiplexor no afecta en rendimiento del PWM.

Tabla 57. Módulo estructura

Sistema	SFC	Entrada / Salida	Función micro	Puerto del micro
Carrusel	SFC1: S1a_M1	Encóder principal	Lectura	B0
		Encóder auxiliar	Lectura	B1
		M1	PWM	C2, D0, E0, E1
			Escritura	D4
Orienta contenedor	SFC2: S1bS1c_M2M3	1B: Sensor posición H	Lectura	B4
		1B: Sensor posición L	Lectura	B5
		M2	PWM	C2,D1, E0, E1
			Escritura	D5
		1C: Sensor posición H	Lectura	B6
		1C: Sensor posición L	Lectura	B7
		M3	PWM	C2,D2, E0, E1
			Escritura	D6
Posición batidora	SFC3: S1d_M4	Sensor posición H	Lectura	B2
		Sensor posición L	Lectura	B3
		M4	PWM	C2,D3, E0, E1
			Escritura	D7
Del módulo	Comunicación	Maestro	I2C	C3, C4
		Tarea	Escritura	C1
		Error	Escritura	C0
		Indicador	Escritura	C5
		Monitor	RS232	C6, C7

En la Figura 86 se representa la estructura del módulo en cuanto a sus características físicas de acondicionamiento, puertos de comunicación, indicadores y fuentes.

*Figura 86. Estructura de módulo 1*

4.3.2 Bloques funcionales

Durante el desarrollo del análisis del proceso, se definieron las funciones de operación para los sistemas de este módulo, a partir de esto, Figura 55, 56 y 57 se desarrollan los bloques funcionales, Figura 87 y 88.

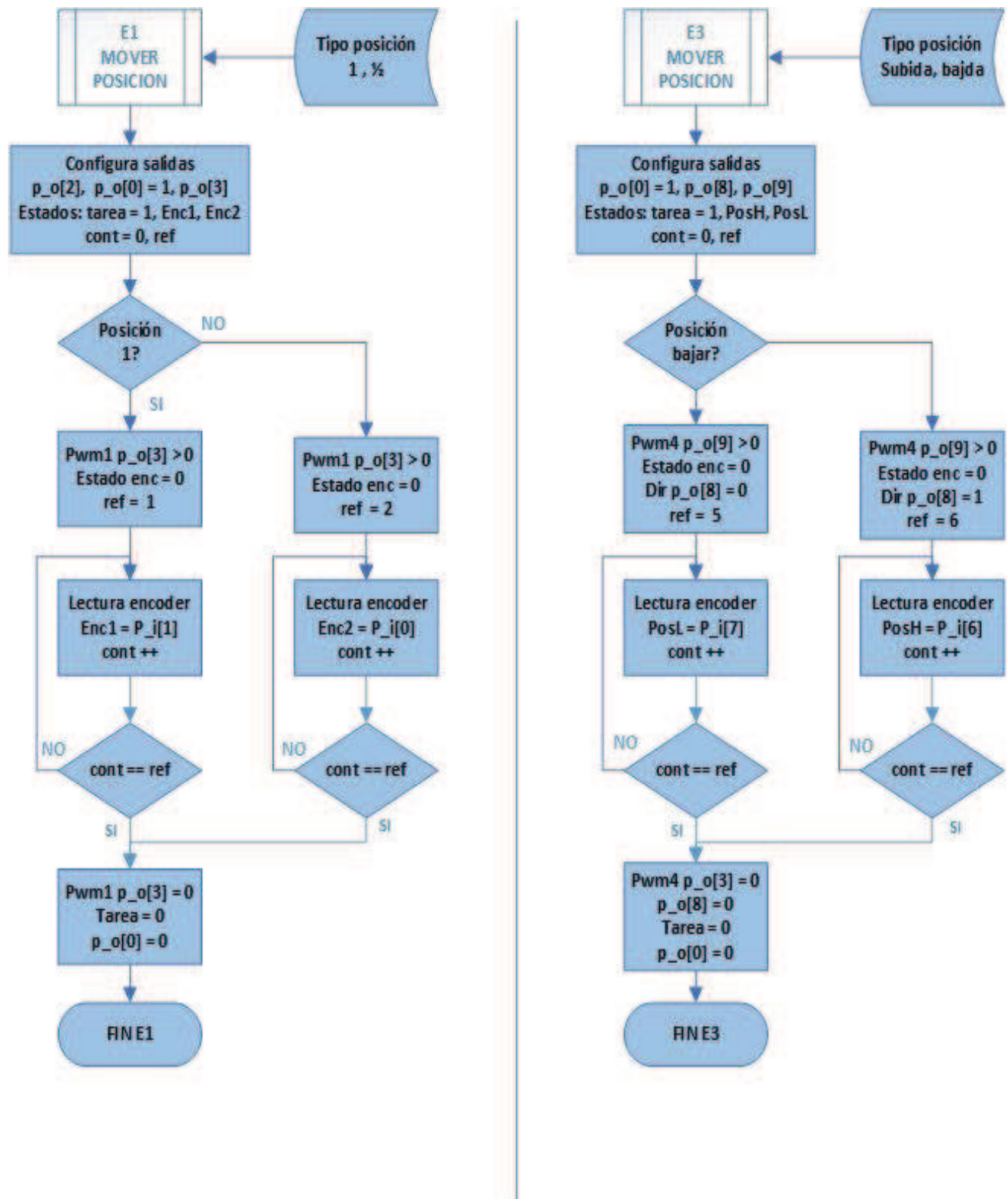


Figura 87. Módulo 1, Bloques funcionales de SFC 1 y SFC 3

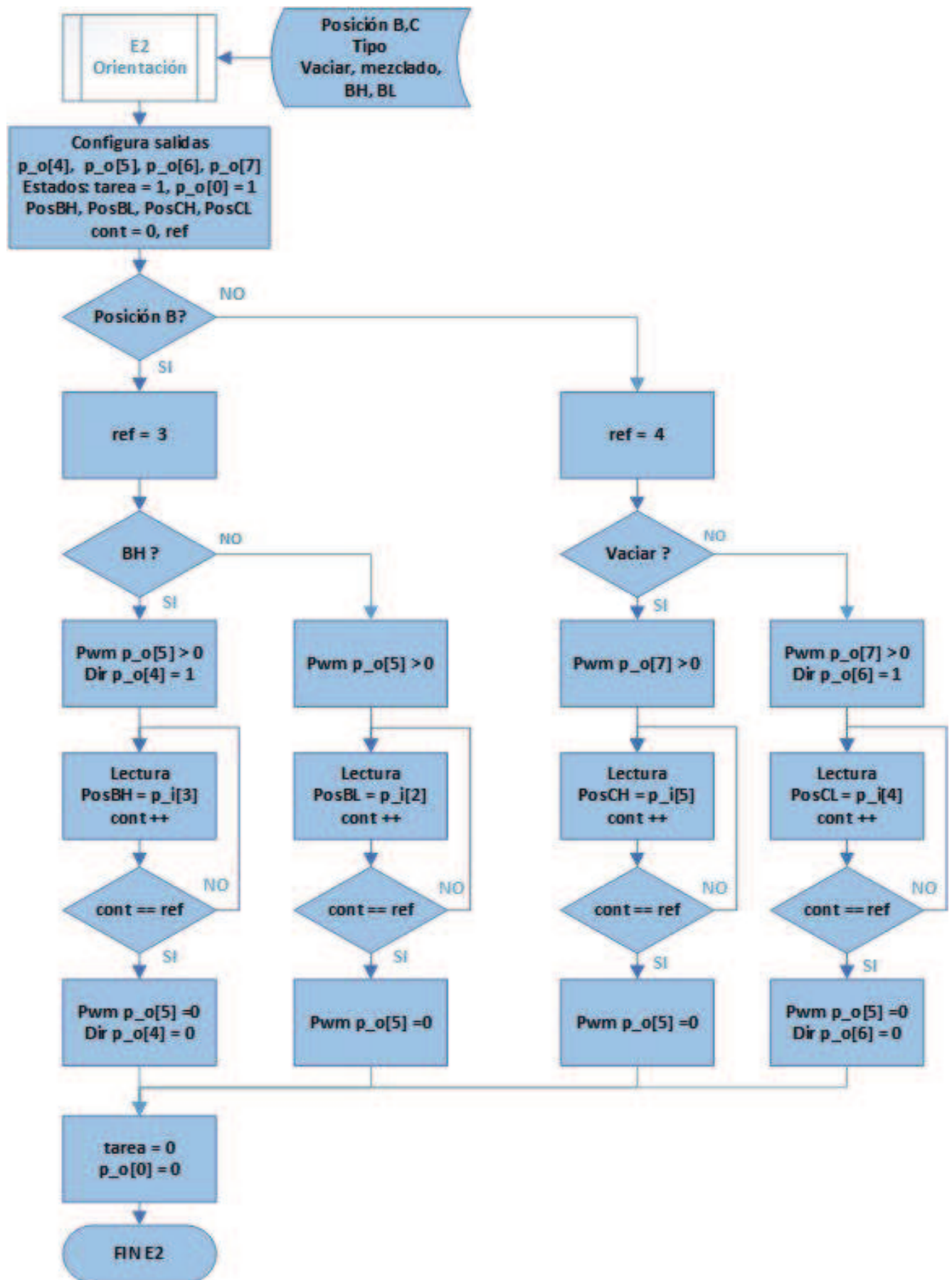


Figura 88. Módulo 1, Bloque funcional de SFC 2

4.3.3 Sistema electrónico

De la Tabla 57, y del diagrama de la estructura de módulo se establece la relación de componentes electrónicos para su implantación, las consideraciones para la selección de componentes sobre las funciones se describen en los siguientes puntos.

Acondicionamiento sensores

Los componentes de sensado del módulo 1 son de dos tipos: encoders y sensor de posición, estos dispositivos deben entregar estados lógicos de 0 y 1. Los encoders son opto eléctricos de barrera de luz, herradura, y de reflexión, sobre objeto; mientras que los sensores de posición son de accionamiento mecánico.

Por otra parte el movimiento de los actuadores en su acoplamiento con los sensores no influye en las velocidades de lectura, los sensores mecánicos están en los extremos de inicio y fin de movimiento y los encóder del sistema carrusel son de 4 y 12 puntos para un tiempo de movimiento no menor a 100 [ns], el límite de lectura para el PIC16F887 con un oscilador de 20 [MHz] es de 2 [ns] y considerando el movimiento de acumulador, instrucciones de lectura y guardado por el uso de compilador en C puede estimarse a no más de 100 [ns].

Bajo estas consideraciones, para el acondicionamiento de los sensores se establece se quiere:

- Comprador para encoders:

Los sensores optoelectrónicas se acordonan por medio de amplificadores operacionales en modo comprador, uno por cada encóder. El OPAM seleccionado es el LM324 por los voltajes de trabajo y el tiempo de cambio. El modo comparador permite asegurar y establecer la afinidad de lectura.

- Filtro paso bajas para sensores mecánicos:

Los sensores por accionamiento mecánico físicamente se encuentran en los extremos del movimiento, por lo que su lectura es única en tiempo de operación, se debe asegurar la lectura inmediata hasta que complete su accionamiento, pues al ser de tipo mecánico tiene un rango de detección hasta que mecánicamente se cierre, por lo que para reducir las variaciones en ese periodo de transición que va del orden de milisegundos, se usa un filtro paso bajas pasivo RC. Los elementos comerciales seleccionados fueron una resistencia de 10 [KΩ] y un capacitor cerámico de 0.22 [uF] para una frecuencia de 72.343 [Hz] considerando el tiempo de transición de una a posición a otra ocurre en un tiempo mayor a 1 segundo.

Acondicionamiento potencia

Por las condiciones físicas del sistema y de operación, los motores usan una corriente menor a 1 [A], por las condiciones mecánicas se usa el dispositivo L298 por las características de corriente máxima de trabajo de 2 [A] y dos puentes H, este dispositivo requiere diodos de protección, seleccionados 1N5400 por su corriente de trabado a 3 [A].

Para el sistema de posicionamiento batidora por condición mecánica de posicionamiento vertical y del sistema carrusel por carga de contenedor se usa la configuración de alta corriente mediante todo el dispositivo L298, mientras que para los actuadores del sistema orientación se usa la configuración normal. De esta forma se requieren tres dispositivos para el acondicionamiento de potencia.

Acondicionamientos en puertos del microcontrolador

Para los elementos de comunicación solo se requieren resistencias pull-up para las líneas de comunicación de I2C y pull-down para las líneas de RS232, de Tarea y de Error con el fin de eliminar los triestados.

Para las entradas de sensores no se requiere elementos adicionales por el acondicionamiento; mientras que para las salidas a los actuadores, debido al número de motores se requiere expandir el número de salida de PWM del microcontrolador, para esto se usa un multiplexor DM74157 por las características de tiempo de cambio estados en 9 [ns]. Además, por las condiciones de operación con el fin de reducir pines de salida, se usa una compuerta NOT 74LS04 en los motores Bidireccionales por un pin dejando su habilitación mediante PWM.

Por otra parte, las consideraciones de frecuencias de PWM en cuanto a la respuesta del motor quedan fuera de los alcances de este trabajo, por convención, por literatura y por experiencia para motores de baja velocidad, menos de 500 [rpm] se usa una frecuencia del PWM a 10 [kHz] para no causar pérdidas de corriente y evitar el ruido auditivo; el microcontrolador PIC16F887 permite una frecuencia de PWM mínima de 1.22 [KHz] bajo el Timer 2. Para un mejor diseño de esta etapa debe recurrirse al estudio de electrónica de potencia.

En la Tabla 58 se pone los materiales para el diseño del sistema eléctrico y en las Figuras 89, 90 y 91 se muestra el esquemático del sistema electrónico por etapas.

Tabla 58. Componentes del sistema electrónico, Módulo 1			
Etapas	componentes	Dispositivo	Valor
Acondicionamiento	Encóder	OPAM	Lm324
		Trimpot	5 [K Ω] x 2
		Resistencia	470 [Ω] x2 10 [K Ω] x 2
	Sensor mecánico	Resistencia	10 [K Ω] x 6
		Capacitor cerámico	.22 [μ F] x 6
Potencia	Motor	Puente H	L298 x 3
		Diodo	1n5400 x 16
Control	Microcontrolador	PIC	16F887
		Cristal oscilador	20 [MHz]
		Resistencia	10 [K Ω] x 1 2.2 [K Ω] x2 470 [K Ω] x 4
		Capacitor cerámico	22 [nF] x 2
		Push botton	4 pines
		LED	Verdes,3 [mm] x 4
		OR	74LS04
		MUX	DM74157
Misceláneos	Conectores	T-Bloc	12
		Header macho	Varios
		Jumper	Varios
	Fuentes	Capacitor electrolítico	100 [μ F] x 3
	pull-down	Resistencias	470 [Ω], varios

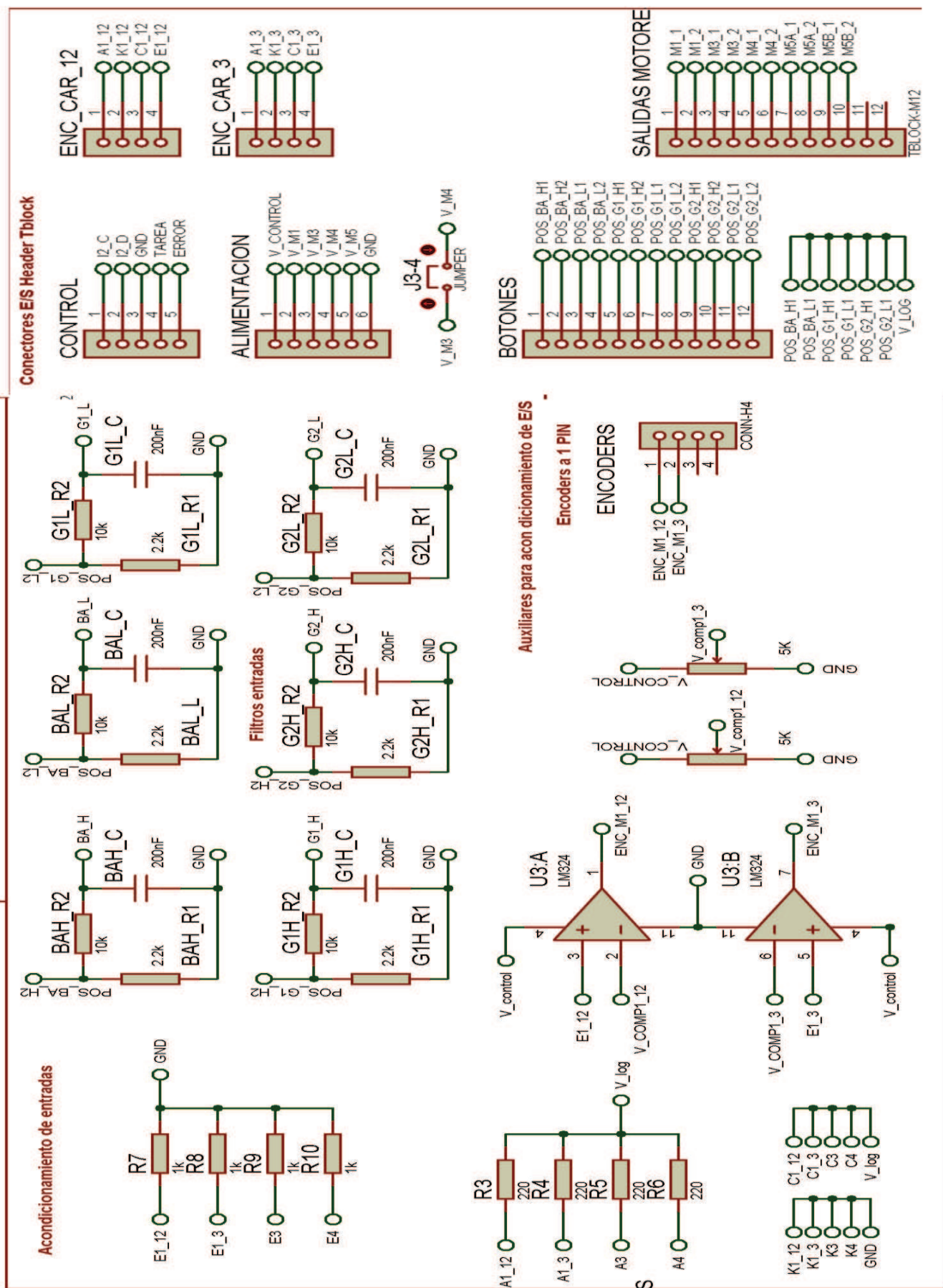


Figura 89. Módulo 1, sistema electrónico, etapa acondicionamiento

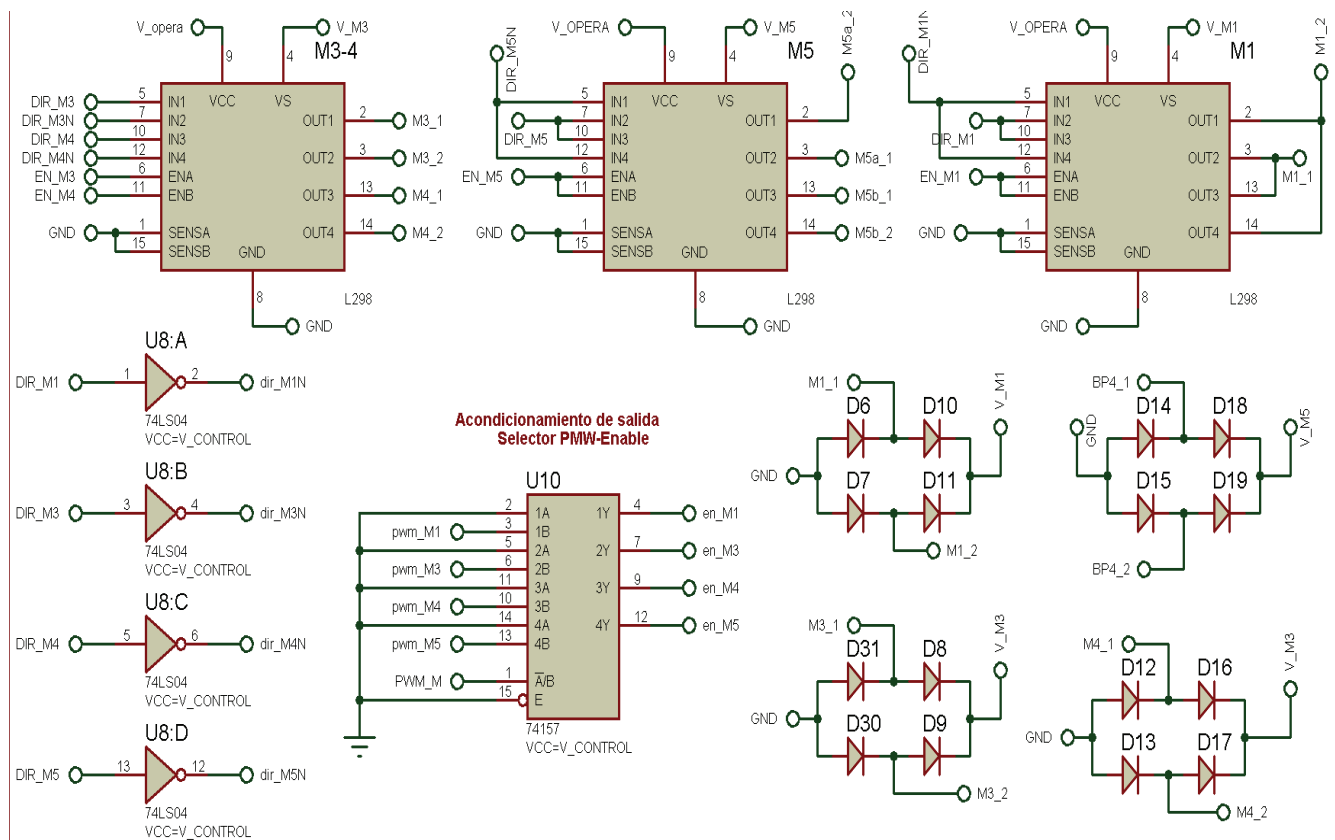


Figura 90. Módulo 1, sistema electrónico, etapa potencia

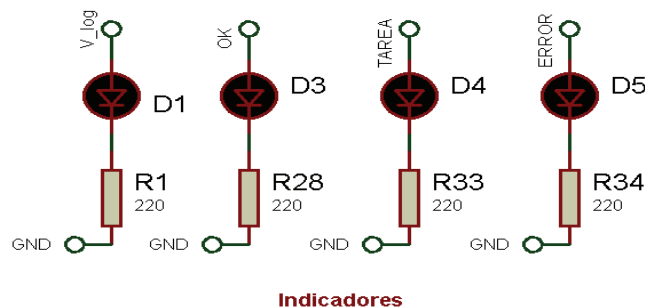
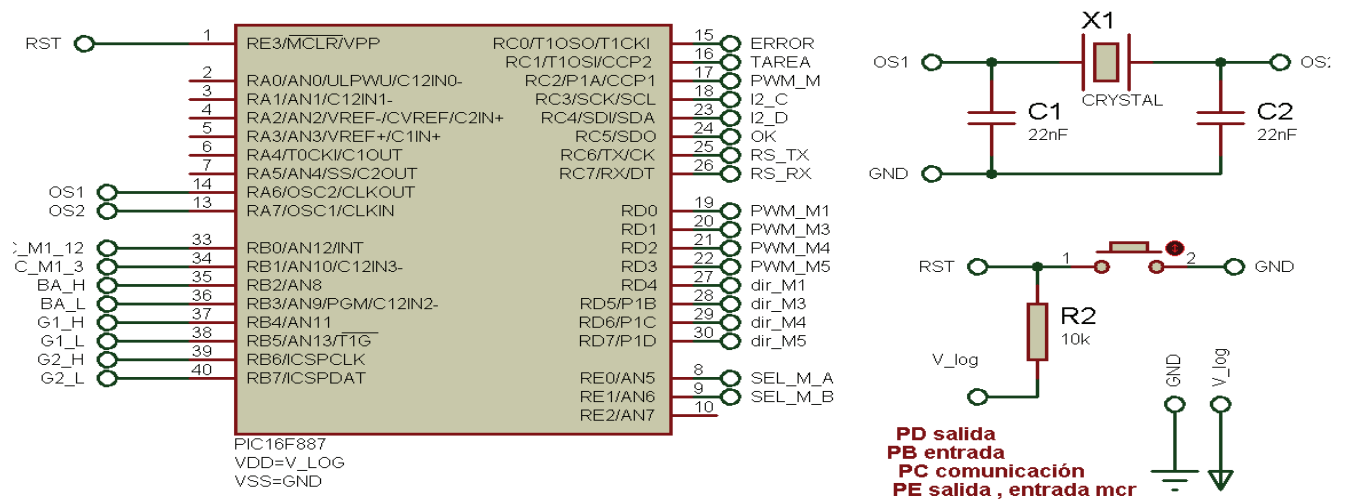


Figura 91. Módulo 1, sistema electrónico, etapa control

4.4 Interfaz de usuario y módulo maestro

El desarrollo de los módulos de las operaciones de configuración y planeación se desarrolló en conjunto al iniciar la segunda etapa de proyecto en un sentido modular y basado en comunicación RS232 e I2C. Los SFC para sus operaciones generaron Grafcet de encapsulamiento mientras que para las operaciones en la unidad de control se desarrolló en cap. 4.2.2 y 4.2.3 en estructura de comunicación y estructura de módulo; para la implementación en microcontroladores se toman las siguientes consideraciones:

- Para la interfaz, se compone de dos dispositivos esclavos, una pantalla y un generador de reportes.
- Los periféricos de entrada y salida se consideran como uno mismo pero no se detallan en cuanto a los pertenecientes a los esclavos, para no particularizar los sistemas.
- Las funciones para los datos no requieren mayor análisis del efectuado en el cap. 4.2.3 así como el tratamiento de la solicitud por paradigmas.
- Los módulos no cuentan con elementos de actuadores del proceso, solo de comunicación.

4.4.1 Módulos

Para el desarrollo de los sistemas electrónicos de los módulos de interfaz y maestro se siguen los siguientes pasos para cada uno de estos:

1. Se establecen los dispositivos intermediarios para acondicionar las señales de datos
2. Se da estructura al módulo a partir del modelo genérico de la Figura 78, para maestro
3. Se desarrolla el sistema electrónico con las características descritas en cap. 4.2.1

De las consideraciones para el desarrollo de estos módulos se tiene el análisis de la estructura los módulos del cap. 4.2.1, el establecimiento de puertos de entrada y salida están descritos en la Tabla 48 para los módulos esclavo, las líneas de comunicación en la Tabla 49 y en la Tabla 50 la representación del modelo Bitbus. Por el análisis de las operaciones del proceso y de la unidad de control no se requiere desarrollar los bloques funcionales, su desarrollo se definió en la estructura de comunicación para solicitudes de envío y lectura, además de las opciones de tratamiento bajo paradigmas, representado los Grafcet encapsulados analizados en cap. 4.1.6.

Los componentes de la interfaz son una pantalla y un generador de reportes, cada uno fue desarrollado en el Apéndice C de forma separada para dejar de forma genérica la implementación, por considera en las etapas avanzadas el uso de otros dispositivos.

Mientras que para el módulo maestro, se debe considerar que cada esclavo tiene sus vías de comunicación, deben ser tratadas por funciones de interrupción, del cual el de mayor prioridad es la línea de tarea. Por las características del microcontrolador solo se tiene un pin de interrupción para cambio de flanco para darle prioridad a un tipo de línea, se propone el uso de compuertas OR para concentrar la entrada de datos y solucionar por software la identificación del dispositivo; para poder tratar estas líneas de comunicación por electrónica se tiene las siguientes consideraciones:

- Las líneas de error tienen mayor prioridad, en cuanto a la estructura de módulo y de datos, se debe determinar el dispositivo de entre los esclavos.

- La línea de tarea indica el tiempo de operación, no importa cual dispositivo termine si no que todas se hayan terminado, en tareas en paralelo y en secuenciales.

De los puntos anteriores, para la lectura del error se establece que las lecturas serán de dos formas, una para detectar el error global y otra para el particular, mientras que para la línea de tarea solo se requiere lectura global. Para el presente trabajo se redujeron los números de módulos por sistemas, de 6 a 4 contando interfaz y el módulo maestro, del cual se definen 4 módulos durante el desarrollo de este trabajo, quedan por definir dos etapas más de las cuales no conoce el número de sistemas; por lo que la implementación de la estructura de comunicación se deja planteado para 8 módulos esclavo.

Para la lectura de tareas y errores en un solo pin, cada uno, se propone el uso de compuertas OR, el dispositivo HCF4078B tiene 8 compuertas OR, tiene un tiempo de respuesta de 75 [ns] y puede funcionar en voltajes de 3 a 15 [V], con poca caída de voltaje (Hoja de especificaciones, fabricante ST Microelectronics <http://www.st.com/> 2016) . Por otra parte, para la identificación del error, las líneas de error se dirigen a su vez a las entradas digitales del módulo maestro, de esta forma, por la Tabla 48 de la distribución y uso recomendado de los puertos del microcontrolador, pueden ser usados hasta 23 pines para detección de errores mediante la lectura de uno a uno de las entradas a partir de la detección global, así se puede identificar más de un error a la vez; puede pensarse en el uso de un DEMUX pero tiene jerarquía por lo que en caso de error n se podría detectar más de un dispositivo.

En la Tabla 59 se muestra que los puertos del micro para PWM son referenciados por dos pines, esto se debe a que el módulo tiene cuatro actuadores y como especificación del microcontrolador solo se cuentan con dos salidas de PWM; la solución para este problema, al igual que en módulo dosificación

Tabla 59. Módulo interfaz y maestro			
Sistema	Entrada / Salida	Función micro	Puerto del micro
Módulo maestro	Comunicación esclavos	Lectura/ escritura	C4
		Reloj	C3
	Error	Lectura por interrupción	B0
		Lectura	D0-D7
	Tarea	Lectura	B1
	Comunicación interfaz	Lectura	C6
		Escritura	C7
	Indicador	Escritura	C5
Interfaz	Comunicación esclavos	Lectura – Pantalla	Rx0
		Escritura – Pantalla	Tx0
		Lectura – Reporte	Rx1
		Escritura – Reporte	Tx1
		Lectura – Maestro	Rx2
		Escritura – Maestro	Tx2
	Indicador	Escritura	53-50
Reportes	Reloj	Lectura/ escritura	C4-C5
	Memoria micro SD	SPI	B2-B5
	Comunicación interfaz	Lectura	D0
		Escritura	D1
	Indicador	Escritura	D5,D7

En la Figura 92 se representa la estructura del módulo interfaz, en cuanto a la estructura de sus componentes, como en la Figura 79.

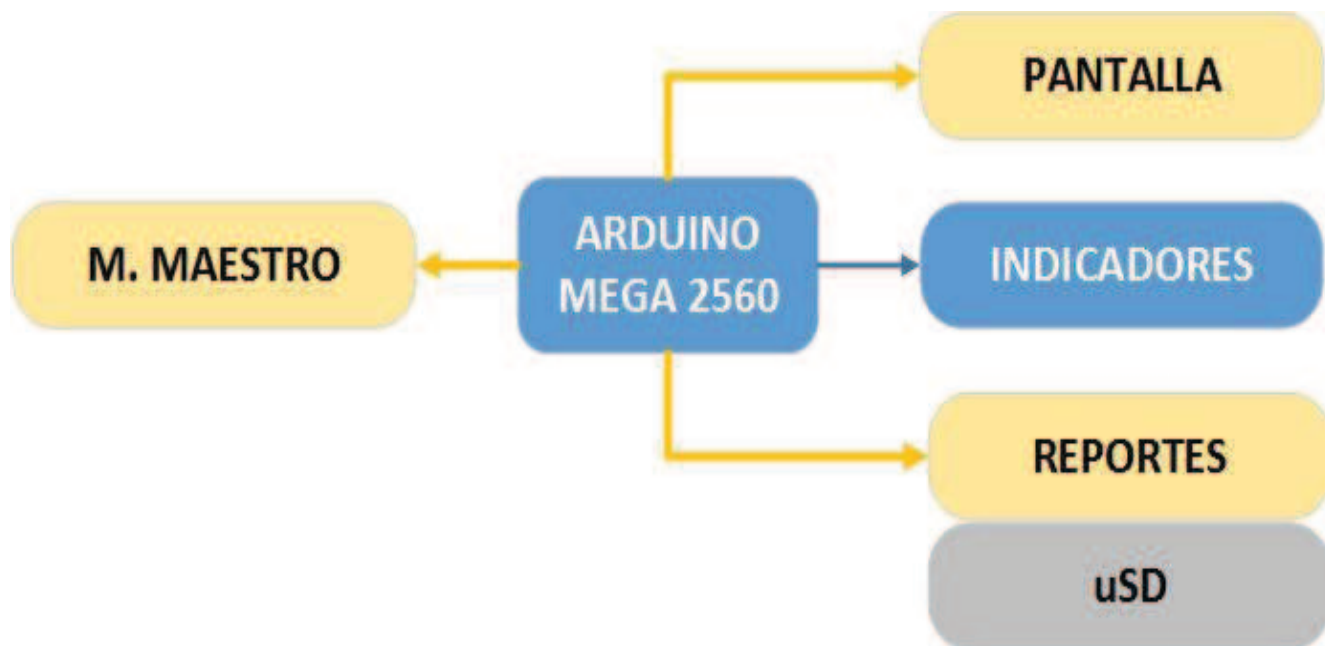


Figura 92. Estructura módulo interfaz

En la Figura 93 se muestra la estructura del módulo maestro, indicando las entradas y salidas de sus componentes. En relaciona la Figura 78, para maestro.

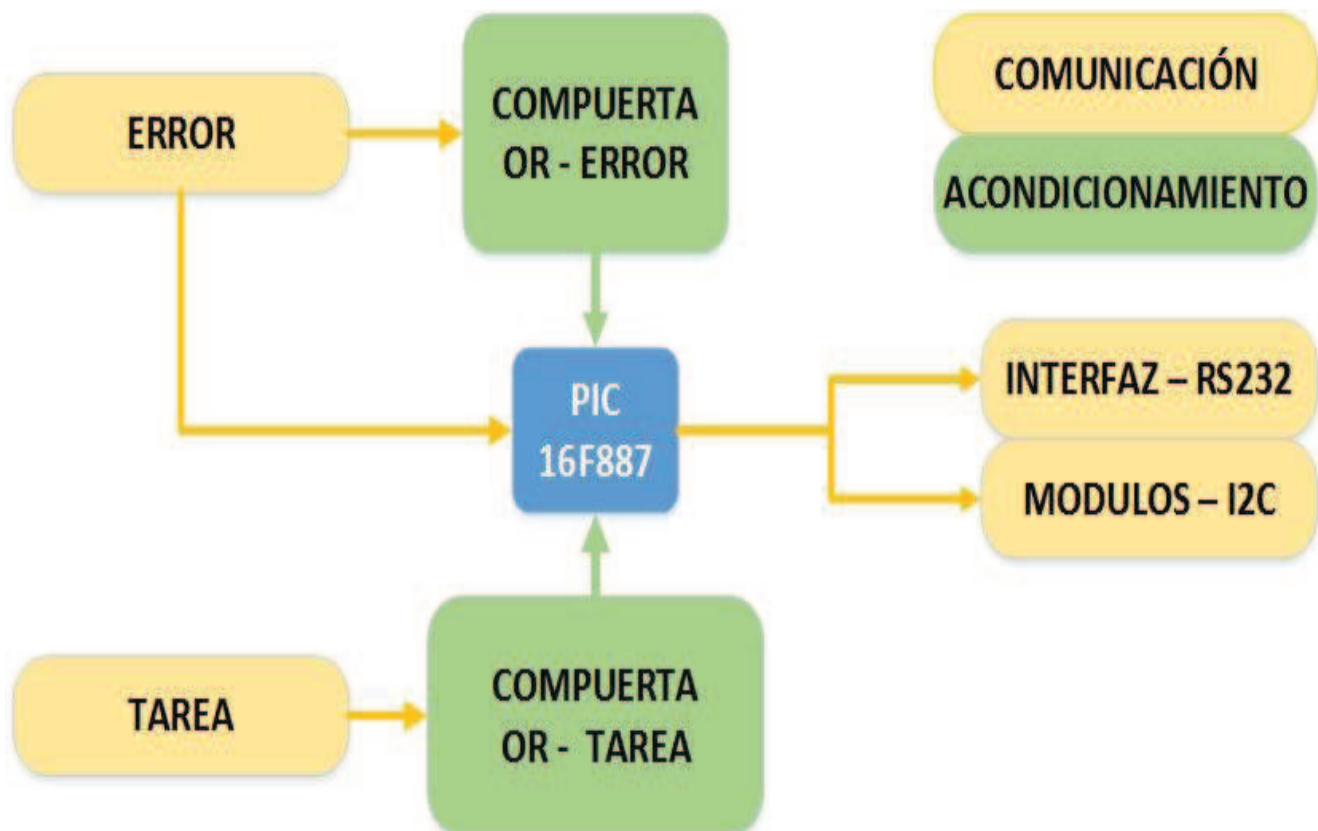


Figura 93. Estructura módulo maestro

4.4.2 Sistemas electrónicos

Los componentes de acondicionamiento para las entradas y salidas están definidas en el desarrollo de cada esclavo, únicamente para el maestro se agregan dispositivos OR HCF4078B para acondicionar las líneas de error y tarea, con resistencias pull-down para evitar triestados.

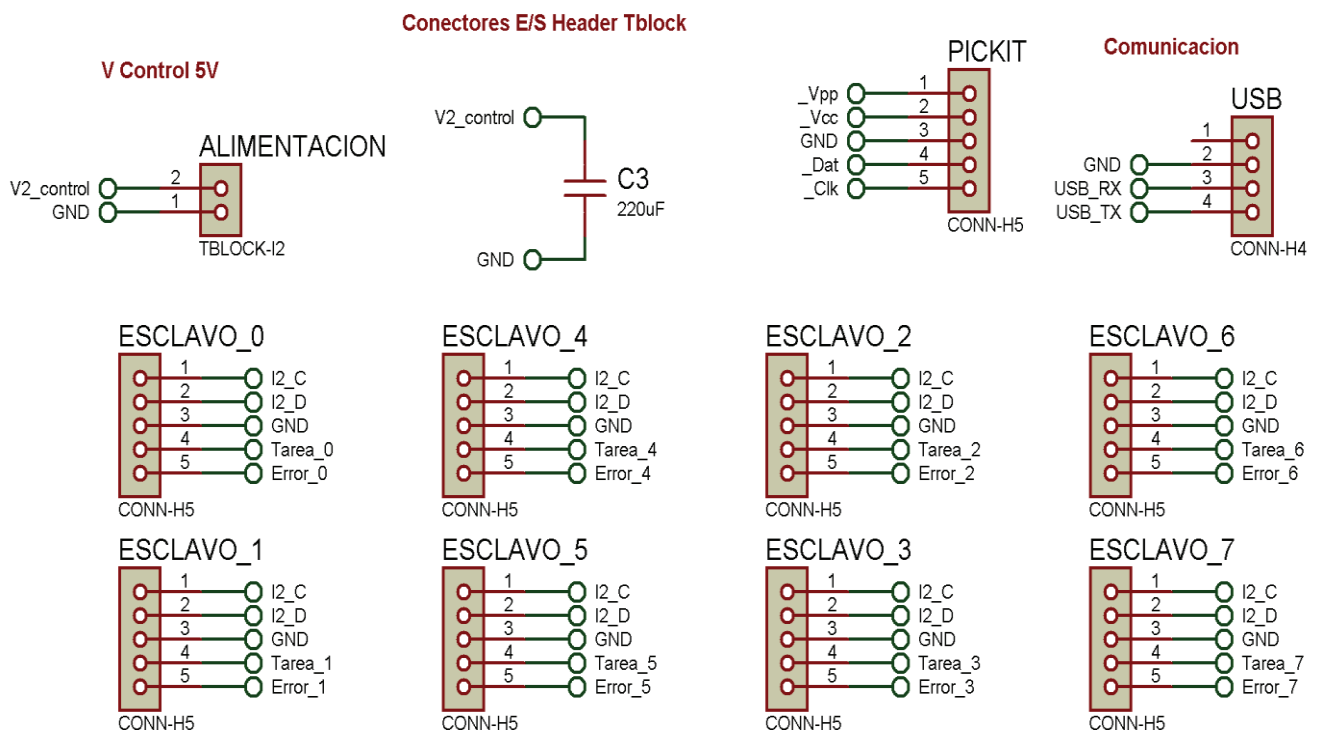
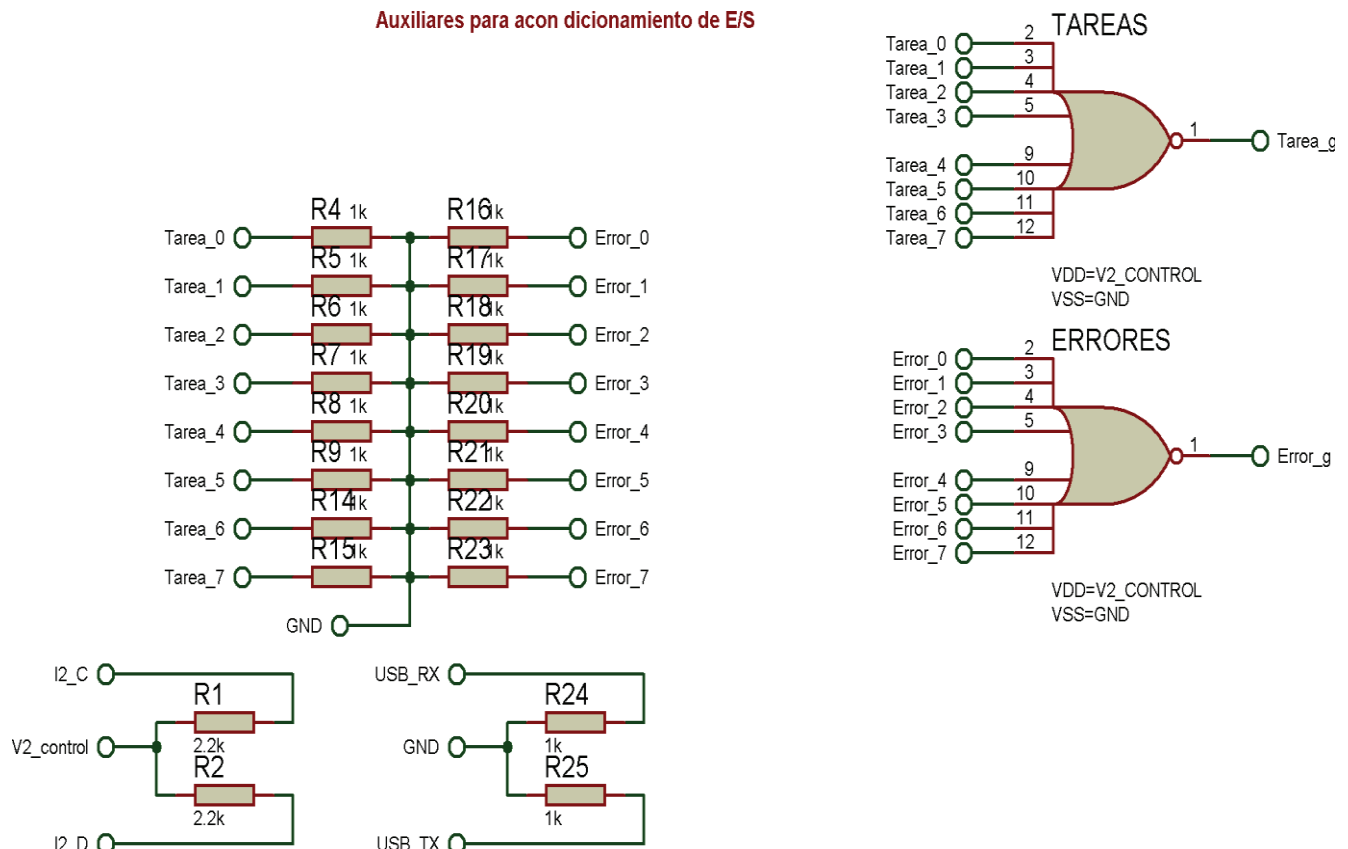
A partir de las estructuras de los módulos y pos sus consideraciones para el establecimiento de la Tabla 60, los componentes para sus sistemas electrónicos son:

Tabla 60. Componentes de módulo de interfaz y de maestro			
Módulo	componentes	Dispositivo	Valor
Maestro	Comunicación	OR	HCF4078B
		Resistencia – tarea	470 [Ω] x 8
		Resistencia – error	470 [Ω] x 8
	Microcontrolador	PIC	16F887
		Cristal oscilador	20 [MHz]
		Resistencia	10 [K Ω] x 1 2.2 [K Ω] x2 470 [Ω] x 4
		Capacitor cerámico	22 [nF] x 2
		Push botton	4 pines
		LED	Verdes,3 [mm] x 4
Interfaz	Arduino mega	LED	Verdes, 3 [mm] x4
		Resistencias pull-down para RS232	470 [Ω] x 6
	Reportes	Memoria micro SD	Lector uSD uSD, máximo 2 GB
	Pantalla		
Misceláneo	Conectores	Header macho	16F887
		Cristal oscilador	20 [MHz]
		Resistencia	10 [K Ω] x 1 2.2 [K Ω] x2 470 [K Ω] x 4
		Capacitor cerámico	22 [nF] x 2
		Push botton	4 pines
		LED	Verdes,3 [mm] x 4
		OR	74LS04
		MUX	DM74157
	Energía	Regulador	7805 , varios
			7803 x 1
			7806 , varios
		Capacitor electrolítico	100 [μ F]

En la Figura 94 se muestra el diagrama electico para el sistema de interfaz, considerando que tanto la pantalla como el Arduino mega son placas pre fabricadas, por lo que se adapta una para integrar las líneas de comunicación.



En la Figura 95 se muestra los elementos de acondicionamiento para las líneas de comunicación; en la Figura 96 los conectores para las terminales de los esclavos y de alimentación.



En la Figura 97 se muestran los componentes para el microcontrolador.

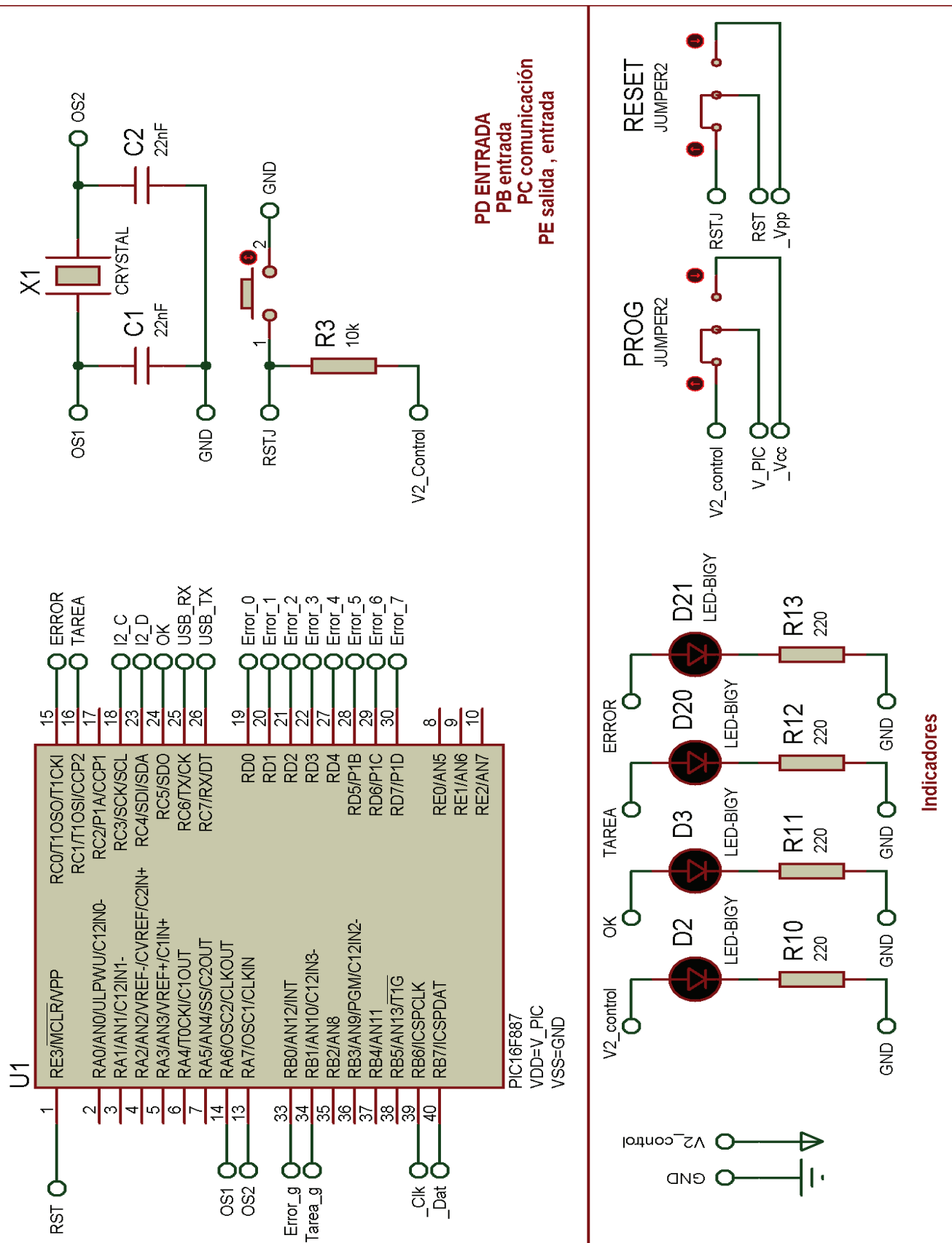


Figura 97. Módulo maestro, componetes de la etapa de control

4.5 Fase 2: Control de batidora

El desarrollo del módulo de mezclado se basa en el control de velocidad de la batidora, se toma las SFC y las especificaciones de diseño para las funciones de operación en relación a las entradas y salidas para su implementación en el microcontrolador. Las consideraciones para el establecimiento del módulo se toman a partir del análisis de las funciones de operación, de comunicación y de la unidad de control, estas son:

- Los componentes del módulo se definen en la Figura 39 y en la Tabla 18.
- La función de operación del proceso está descrito por el SFC de la Figura 58.
- En cuanto a la estructura del módulo, a partir de las funciones de operación se describieron el set de instrucciones y su evolución en las operaciones de planeación, Tabla 53.
- De sus componentes, se definieron los puertos de entrada y de salida con su tipo de acondicionamiento y control, Tabla 48.
- Para la implementación de la función de operación, se requiere el desarrollo de un control de velocidad por método de la curva de reacción de Ziegler-Nichols.

4.5.1 Módulo control batidora

La definición del módulo de mezclado a partir de su implantación en microcontrolador para su sistema electrónico y sus funciones de operación se desarrolla en cinco partes:

1. Se definen la relación de funciones del proceso con los puertos de entrada y salida que requiere por componentes en el microcontrolador.
2. Se da estructura al módulo a partir del modelo genérico de la Figura 78, para esclavo.
3. Se establece el algoritmo de control de velocidad.
4. Se establecen su bloque funcional a partir de su SFC con variables asociadas a la estructura de funciones de la UC.
5. Se desarrolla el sistema electrónico con las características descritas en cap. 4.2.1.

Por las consideraciones para el desarrollo se tiene que el actuador requiere un control de velocidad, esto implica el uso de funciones de interrupción y de Timer para la lectura de la frecuencia angular; Debido a esto se considera un módulo para este sistema siendo una de las especificaciones de diseño con mayor peso, cap. 3.4.

En cuanto al microcontrolador, en la Tabla 46 se define por los puertos de I/O de forma general, en Tabla 47 y 48 se presentan los puertos de entrada, de salida y funciones que requiere el módulo de mezclado. En la Tabla 61 se establece los componentes para la estructura del módulo.

Tabla 61. Módulo batidora				
Sistema	SFC	Entrada / Salida	Función micro	Puerto del micro
Batidora	SFC4: S2a_M5	Encóder	Lectura Interrupción Timer 1	B0
		M5	PWM	C2
		Dirección	Escritura	D0

Del módulo	Comunicación	Maestro	I2C	C3,C4
		Tarea	Escritura	C1
		Error	Escritura	C0
		Indicador	Escritura	C5
		Monitor	RS232	C6, C7

En la Figura 98 se representa la estructura del módulo en cuanto a sus características físicas de acondicionamiento, puertos de comunicación, indicadores y fuentes.

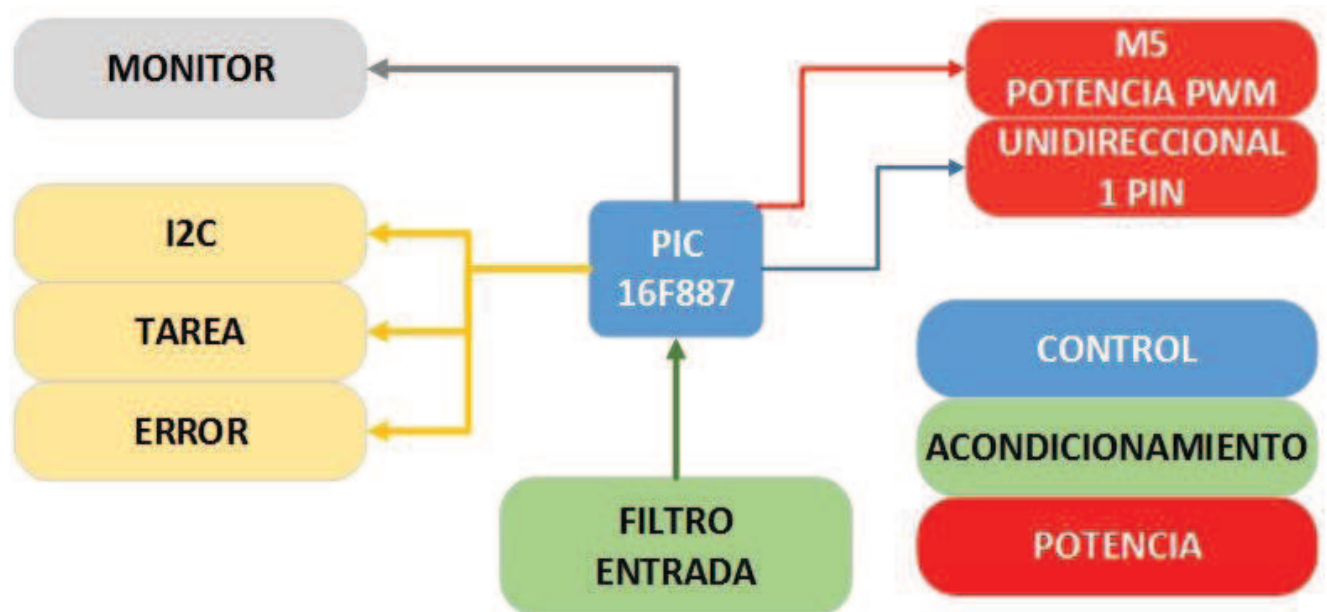


Figura 98. Estructura de Módulo 2

Para el control de velocidad, se recurrió al método de la curva de reacción de ZN debido a las siguientes consideraciones:

- Las características del motor no se relacionan a la hoja de especificaciones.
- Del punto anterior, la velocidad máxima registrada fue de 300 [rpm].
- El tiempo para su desarrollo es limitado.
- El tiempo de asentamiento no afecta en los resultados del proceso debido a que se analiza la mezcla final.
- La variación de rpm durante ejecución no afectan al resultado de las pruebas si estas son menos de 10%, Apéndice A y B.

A manera de resumen, se desarrolló el análisis de la respuesta de la planta de acuerdo al método de ZN, las pruebas determinaron tanto las ganancias del control como la velocidad máxima del motor, a su vez esto permitió establecer la factibilidad de implantación del control en el microcontrolador PIC16F887; En la Tabla 62 y en la Figura 99 se muestra las pruebas de ZN para la obtención ganancias de control.

Tabla 62. Parámetros y constantes de control PID por ZN		
Parámetros de ZN	Parámetros de PID	Constantes de algoritmo
$T_0 = 12 - -8 = 20 \text{ ms}$	$K_p = (1.2 * Y_0) / (K_0 * T_0) = 14.328358$	$a = K_p = 14.328358$
$Y_0 = 76 - 12 = 64 \text{ ms}$	$T_i = 2 * T_0 = .04 \text{ s}$	$b = K_p * 0.1 / T_i = 35,8208955$
$K_0 = 2.68 / 10 = 0.268$	$T_d = 0.5 * T_0 = 0.01 \text{ s}$	$c = K_p * T_d / 0.1 = 1,432835$

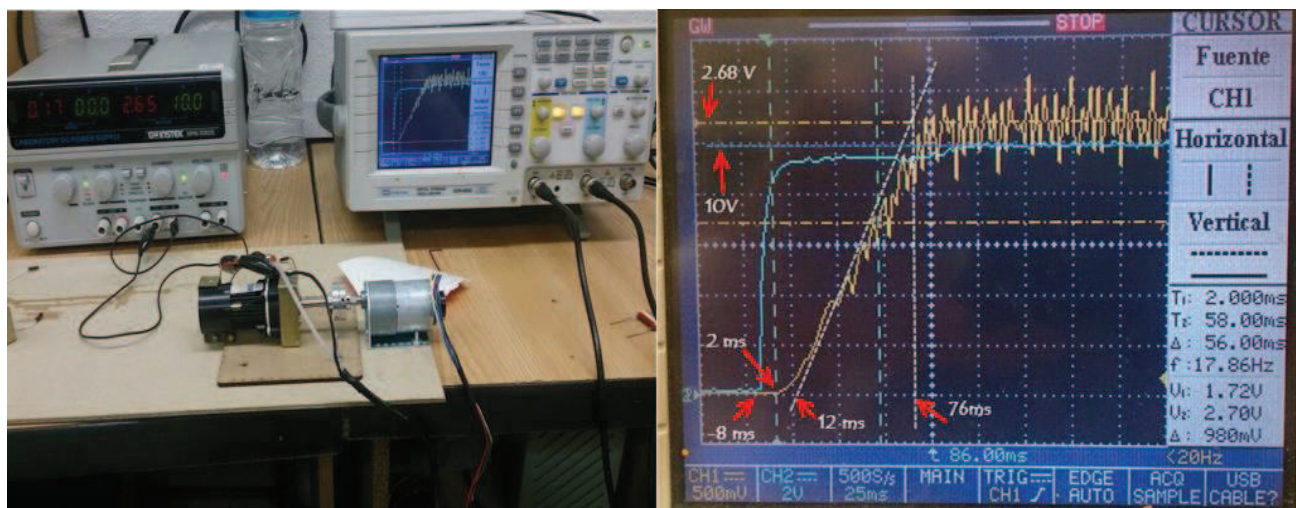


Figura 99. Módulo 2, prueba de respuesta al escalón, ZN

Con las ganancias calculadas se implementó el algoritmo de control de la Figura 100.

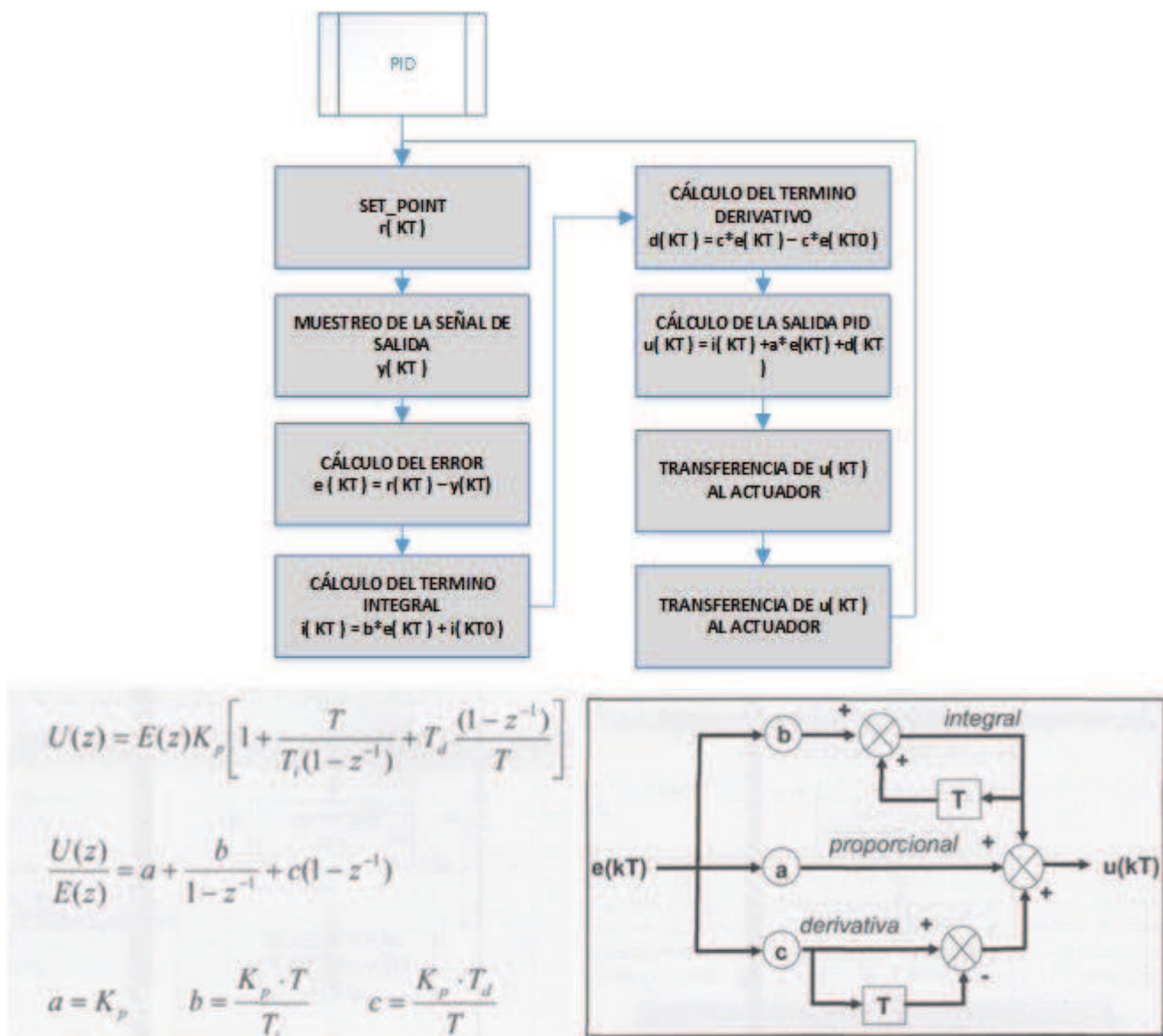


Figura 100. Algoritmo para control PID en microcontroladores

Las primeras pruebas del controlador definieron las siguientes consideraciones:

- Se ajustan las ganancias, considerando que durante las pruebas hubo imperfecciones mecánicas del banco de pruebas.
- El tiempo de muestreo se establece en función de la velocidad del encóder por función de interrupción de la siguiente forma: a partir del flanco de subida se mide el tiempo de duración hasta el flanco de bajada, se deshabilita la interrupción para calcular la ganancia y se vuelve a habilitar la interrupción para flanco de subida.
- Para velocidades bajas al algoritmo le toma más iteraciones que en altas velocidades.
- A pesar de que el motor fue arrancado previamente para romper la inercia, la ejecución del control inicia con una caída de la ganancia por iniciar en ceros sus variables de operación. Esto afecta al motor debido a tiene un mínimo de rpm para no frenarse.
- El sobrepaso y las oscilaciones que presenta el control son normales, pues el método de ZN tiene este efecto por obtener las ganancias sin conocer el modelo de la planta.

En la Figura 101 se muestra las lecturas realizadas durante la ejecución del control, cada iteración representa las operaciones de lectura y cálculo de ganancia. A la izquierda el control con cambio de referencia, se aplicó perturbación en el estado estacionario; a la derecha el control a una velocidad baja. Para las pruebas, el motor inicia con velocidad, en referencia a las funciones de operación, pues por condición durante su posicionamiento por parte del módulo 1, se ejecuta la función batir para romper inercia y luego se solicita la de mezclado al estar en posición, cap. 4.1.4.

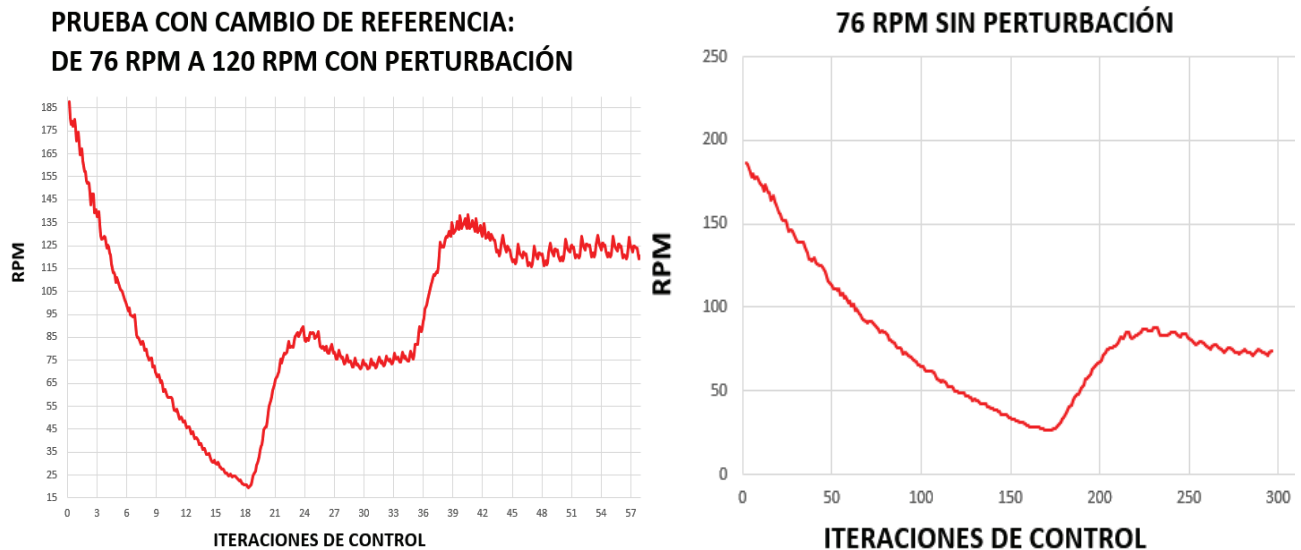


Figura 101. Pruebas de control

Al dejar el muestreo en función de la velocidad del encóder tiene una desventaja, en caso de que una perturbación frene la flecha al instante no se podrá ejecutar los cálculos por espera del cambio de flanco, para la solución de este error se usa la función de la estructura de operaciones del módulo de detección de error por la interrupción del Timer.

Para reducir los efectos de arranque por los cálculos, se agregó al algoritmo de control una tabulación de ganancias para reducir los efectos, para esto se realizaron pruebas en diferentes valores de velocidad y se registraron los valores de las variables en estado estable, de tal forma que al inicio del algoritmo se compara la referencia más cercana al tabulado y se pre cargan los valores de las variables.

En la Figura 102 se muestra la comparación del control con las ganancias tabuladas, de lado izquierdo se inicia con una velocidad cercana a la de referencia, a la derecha se inicia con la velocidad más cercana a la tabulada; obsérvese que el efecto del control es el mismo pero el número de iteraciones que le toma con tabulación es menor y en el inicio se suprime la caída de la ganancia de control.

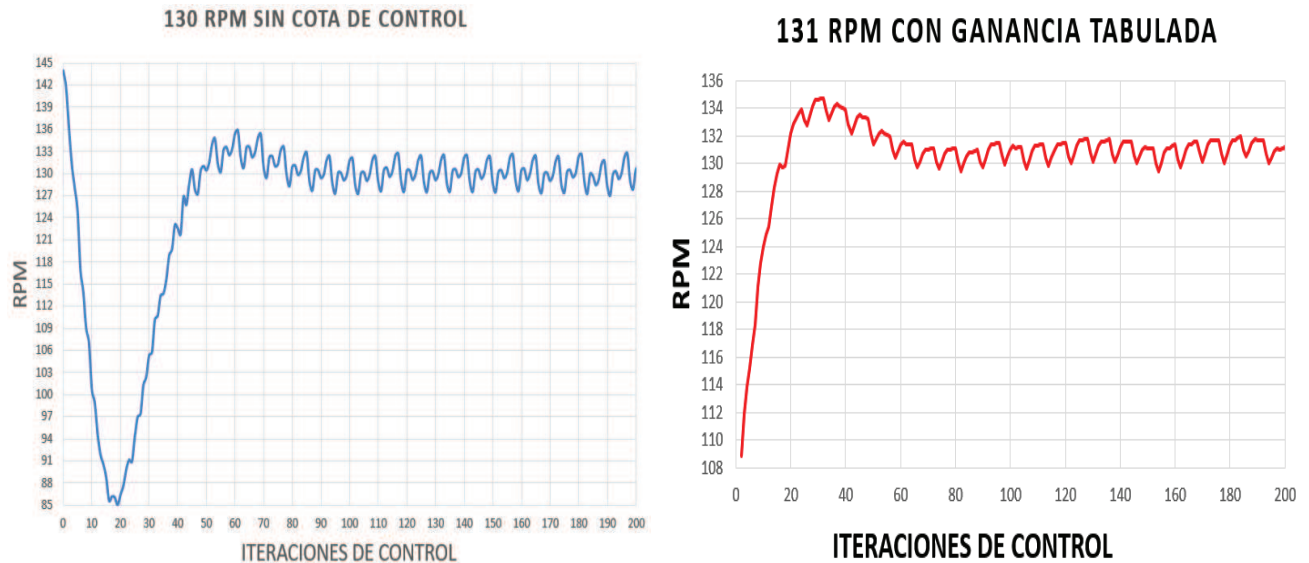


Figura 102 . Comparación del arranque del control por ganancias tabuladas

Por otra parte, en cuanto al error en estado permanente, en la Figura 103 se muestra una prueba con referencia cercana a la tabulada, la variación de la velocidad es de 2 [rpm], en la Figura 102 derecha se aprecia el mismo rango de variación.



Figura 103. Error en estado permanente

Es importante aclarar que esta tabulación solo afecta al inicio del control pero no altera el efecto de sobrepaso propio del método de ZN, además resulta beneficioso pues durante la ejecución de operaciones del módulo no se vuelve a cambiar la referencia, además por tiempos de operación se solicita la ejecución del algoritmo cuando se encuentra en posición y en contacto con el fluido, por lo que la caída de la ganancia por los cálculos puede provocar el paro del motor

4.5.2 Bloques funcionales

Durante el desarrollo del análisis del proceso, se definieron las funciones de operación para el sistema de este módulo, a partir de esto, Figura 58, se desarrolla su bloque funcional, Figura 104.

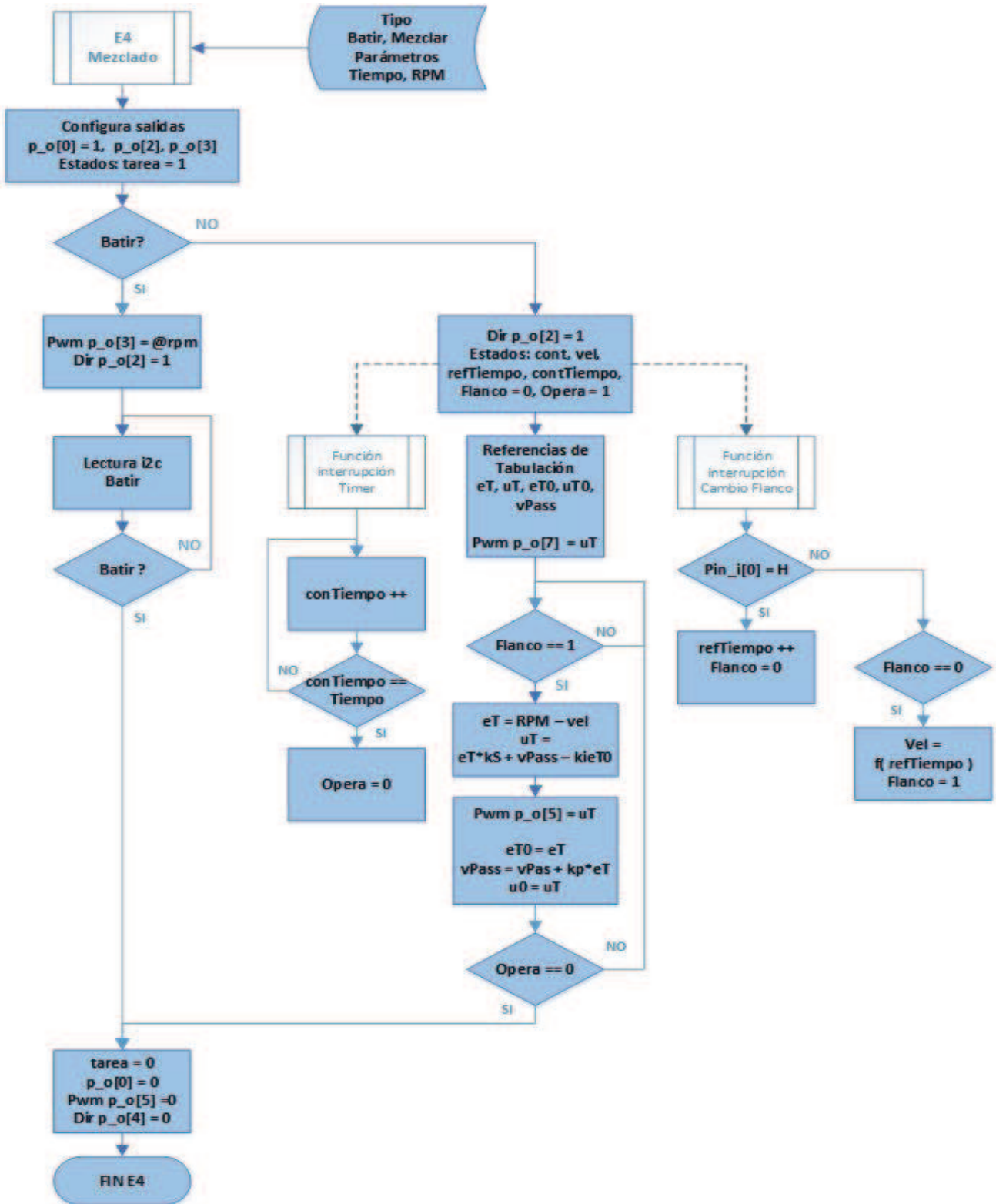


Figura 104. Módulo 2, Bloque funcional, SFC 4

4.5.3 Sistema electrónico

De la Tabla 62, y del diagrama de la estructura de módulo se establece la relación de componentes electrónicos para su implantación, las consideraciones para la selección de componentes sobre las funciones se describen en los siguientes puntos.

Acondicionamiento sensores

El componente de sensado del módulo 2 es un encóder de efecto hall incorporando con el mismo actuador, tiene dos salidas de sensor con 16 puntos por revolución, **ppr**, en la flecha del motor con una reducción de 43.7:1, lo que representa 699 ppr en la flecha de reducción. La velocidad del motor es de 300 [rpm], mediante la medición en laboratorio se comprobó las capacidades el microcontrolador para realizar la medición, en la Figura 105 se muestra la lectura. El motivo de corroborar la medición fue para probar las funciones de interrupción del microcontrolador para la implementación del control, a su vez para su uso en el sistema modular.

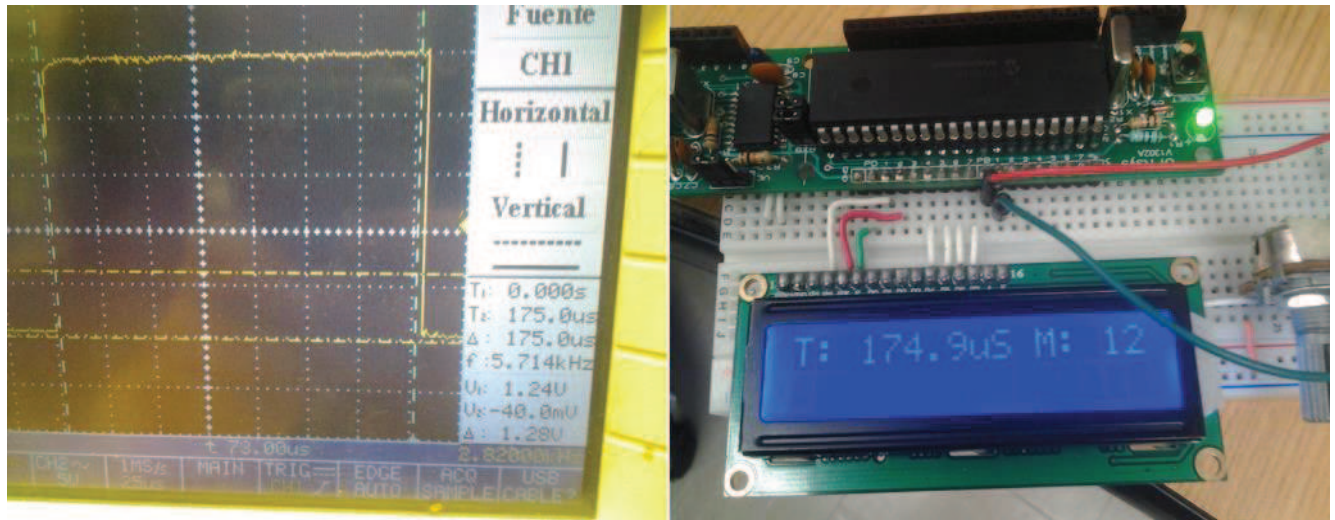


Figura 105. Lectura de encóder a 300rpm

Se acondiciona ambas salidas del encóder mediante amplificador operacional LM324 en comprador, con Trimptot de 5 [K Ω] y resistencias pull-down a la salida, para asegurar la lectura pues aunque la señal que entrega es cuadrada, tiene un voltaje de 1.24 [V].

Acondicionamiento potencia

Las especificaciones del motor en consumo de corriente es de 0.4 [A], a 12 [V], por lo que puede usarse un puente H L293D, pero por condiciones de diseño se cuenta con dispositivos L298 y diodos 1N5400. A demás, al ser el único actuador del módulo, el puente H se usará en modo de demanda de corriente pensando en que para futuras aplicaciones se cambie el modelo del actuador.

Acondicionamientos en puertos del microcontrolador

Para los elementos de comunicación solo se requieren resistencias pull-up para las líneas de comunicación de I2C y pull-down para las líneas de RS232, de Tarea y de Error con el fin de eliminar los triestados. Las entradas y salidas del senado y la potencia no requieren elementos adicionales. Se mantiene las consideraciones de la frecuencia del PWM mencionadas en el cap.

4.4.3. En la Tabla 63 se pone los materiales para el diseño del sistema eléctrico y en las Figuras 106, 107 y 108 se muestra el esquemático del sistema electrónico por etapas.

Tabla 63. Componentes del sistema electrónico, Módulo 2			
Etapas	componentes	Dispositivo	Valor
Acondicionamiento	Encóder hall	OPAM	LM324
		Trimpot	5 [K Ω] x 2
		Resistencia	470 [Ω] x2
			10 [K Ω] x 2
Potencia	Motor	Puente H	L298
		Diodo	1N5400 x 4
Control	Microcontrolador	PIC	16F887
		Cristal oscilador	20 [MHz]
		Resistencia	10 [K Ω] x 1 2.2 [K Ω] x2 470 [Ω] x 4
		Capacitor cerámico	22 [nF] x 2
		Push button	4 pines
		LED	Verdes,3 [mm] x 4
Misceláneos	Conectores	T-Bloc	12
		Header macho	Varios
		Jumper	Varios
	Fuentes	Capacitor electrolítico	100 [μ F] x 3
	pull-down	Resistencias	470 [Ω], varios

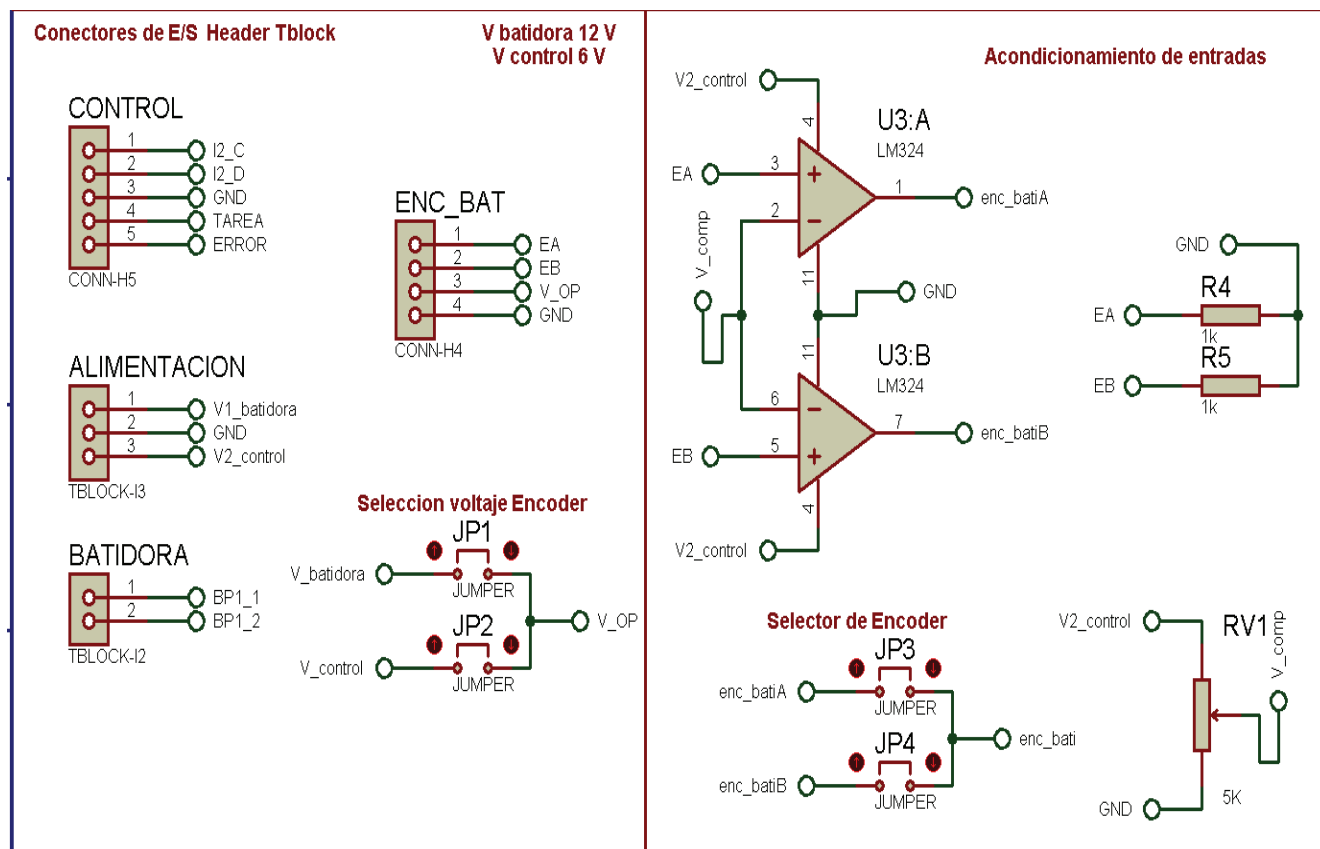


Figura 106. Módulo 2, sistema electrónico, etapa acondicionamiento

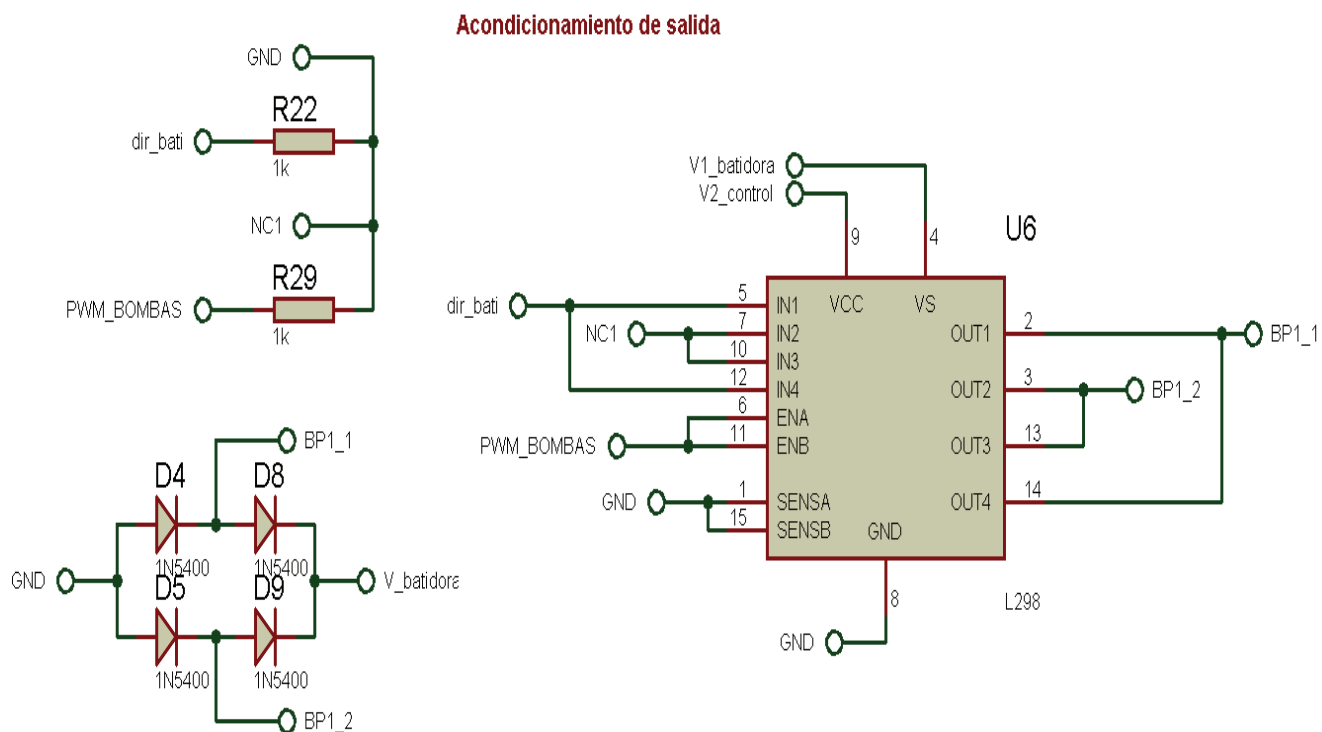


Figura 107. Módulo 2, sistema electrónico, etapa potencia

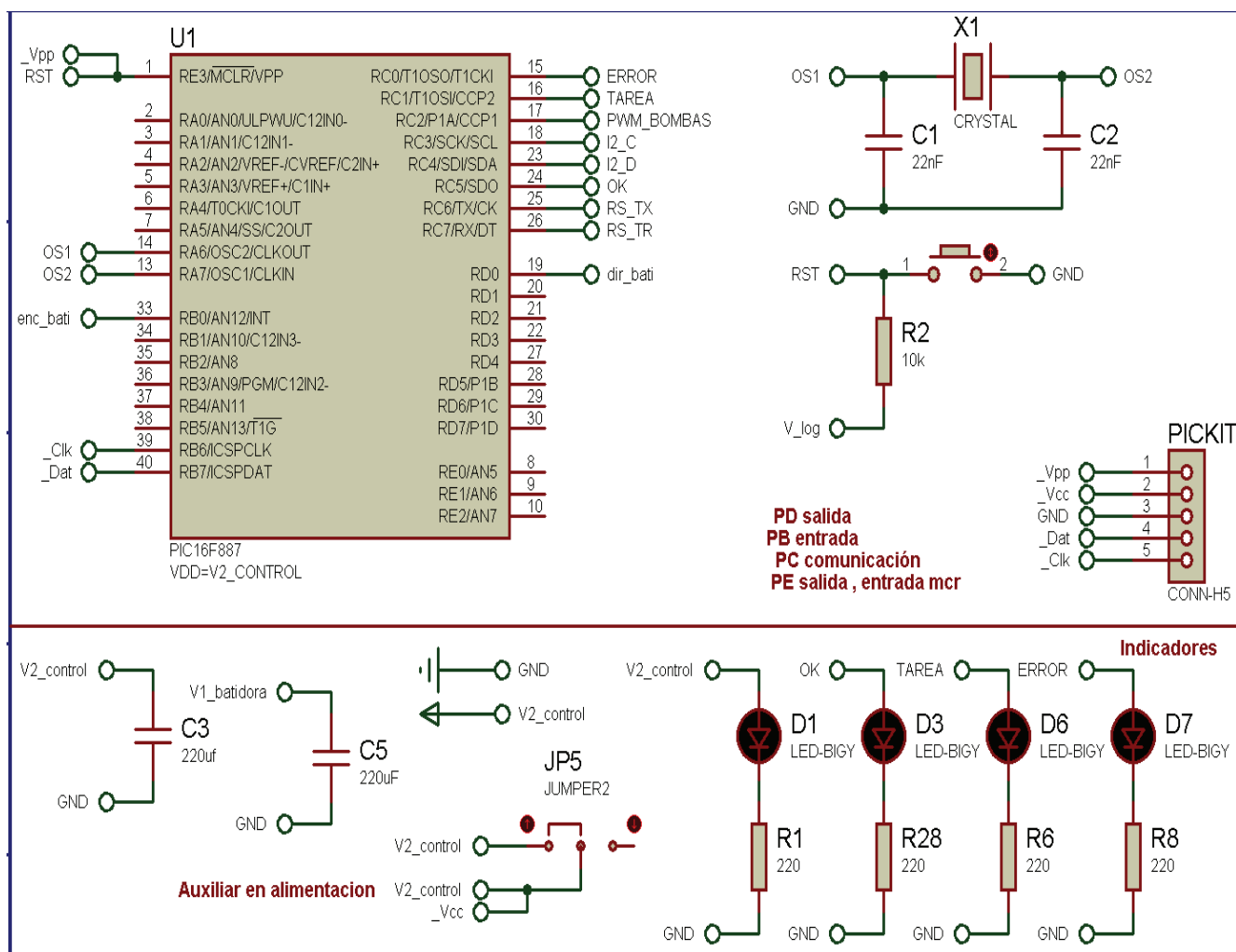


Figura 108. Módulo 2, sistema electrónico, etapa de control

4.6 Fase 3: Dosificación floclante y efluente

Para el desarrollo de los módulos de dosificación se consideran las condiciones físicas y de operaciones de proceso para su ejecución, su diseño fue en diferentes etapas del proyecto; mientras que para el módulo de efluente las condiciones de diseño fueron sobre su bomba de dosificación, sobre el sensado de volumen y control ON-OFF; para el módulo de floclante se diseñó a partir de las especificaciones del proyecto para selección de su bomba, tipo encóder y tipo de control; se toma las SFC, la jerarquía de componentes y operaciones del proceso para relacionar las funciones de módulo para su implementación en el microcontrolador. Las consideraciones para el establecimiento de los módulos son:

Efluentes

- Los componentes se definen en la Figura 40 y en la Tabla 19.
- La función de operación del proceso está descrito en el SFC, Figura 59.
- De la estructura de módulo, su set de instrucciones está en la Tabla 53 y sus componentes en la Tabla 48.
- La bomba fue seleccionada por el equipo mecánico, en función de su posición física y potencia de bombeo, por lo que sólo se hace control ON-OFF por sus características.
- El sensor de volumen será un ultrasónico, para no afectar las condiciones mecánicas.
- Por condiciones del proceso, la cantidad de dosificación está en un rango de 200-400 [ml], Tabla 52.

Floclantes

- Los componentes se definen en la Figura 41 y en la Tabla 19.
- Las funciones de operación de sus procesos están descritos en las SFC de las Figuras 60 y 61.
- De la estructura de módulo, su set de instrucciones está en la Tabla 53 y sus componentes en la Tabla 48.
- El diseño requiere selección de tipo de lectura y control para dosificación.

4.6.1 Módulo de dosificación del efluente

El establecimiento del módulo de efluente en el microcontrolador se desarrolla en cinco partes:

1. Se define la relación de funciones del proceso con los puertos de entrada que requiere sus componentes.
2. Se da estructura de módulo a partir del modelo genérico de la Figura 78, para esclavo.
3. Se define el tipo de función para la medición del volumen de fluido a partir de sensor ultrasónico.
4. Se establece su bloque funcional a partir de su SFC.
5. Se desarrolla el sistema electrónico.

Por las especificaciones del actuador, solo se requiere un control ON-OFF, por otra parte ese necesita un procesamiento del sensor ultrasónico para determinar el nivel de altura, Debido a que se esperan modificaciones en el tipo de bomba y tipo de sensor, se desarrolla como un módulo. Las funciones del microcontrolador serán de interrupción por cambio flanco y uso de Timer para mejorar la lectura del sensor; de tal forma que tendrá una adquisición similar al del módulo 2.

En cuanto al microcontrolador, en la Tabla 46 se define por los puertos de I/O de forma general, en Tabla 47 y 48 se presentan los puertos de entrada, de salida y funciones que requiere el módulo de lodo. En la Tabla 64 se establece los componentes para la estructura del módulo.

Tabla 64. Módulo efluente				
Sistema	SFC	Entrada / Salida	Función micro	Puerto del micro
Lodo	SFC5: S3a_M6	Señal sensor ultrasónico	Lectura Interrupción Timer 1	B0, D0
		M6	Escritura	D1
Del módulo	Comunicación	Maestro	I2C	C3,C4
		Tarea	Escritura	C1
		Error	Escritura	C0
		Indicador	Escritura	C5
		Monitor	RS232	C6, C7

En la Figura 109 se representa la estructura del módulo en cuanto a sus características físicas de acondicionamiento, puertos de comunicación, indicadores y fuentes.

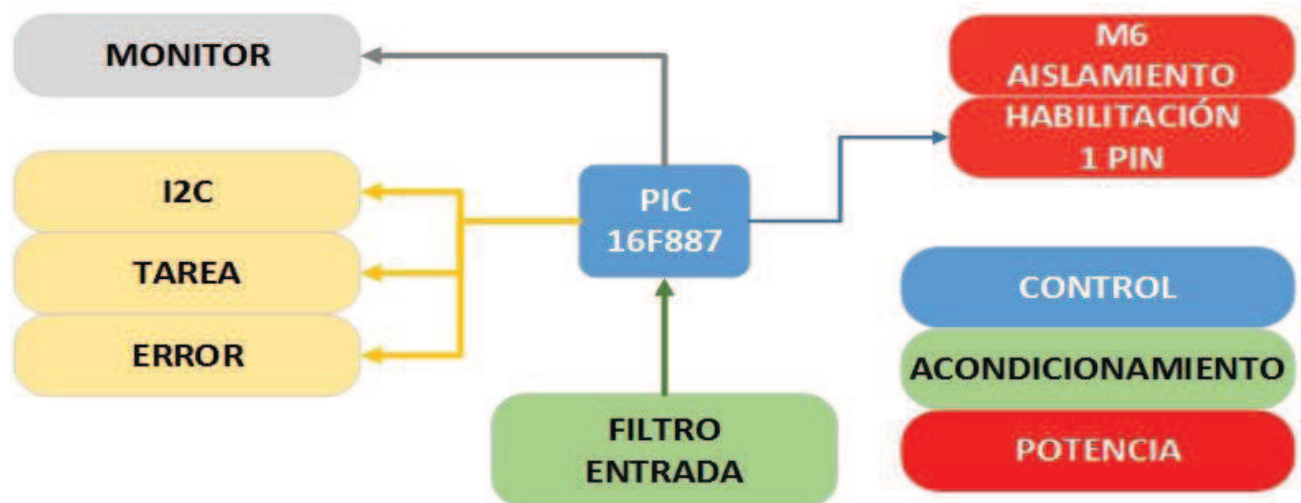


Figura 109. Estructura de Módulo 3

Para el procesamiento de la entrada, se requiere establecer un algoritmo de acondicionamiento; el funcionamiento del sensor es el siguiente, Figura 110.

A partir de la señal de control Trigger, se emite una señal sónica, luego mientras es reflejado por un objeto se levanta un pulso en la señal salida, Eco. Una vez la caída del Eco puede enviarse una nueva señal de Trigger, se recomienda esperar al menos 50 [ms]; el medio de propagación es el aire del ambiente por lo que la medición puede variar en función de la humedad y temperatura. Para la conversión para de distancia en función del tiempo se debe multiplicar el tiempo del Eco por la mitad de la velocidad del sonido en el ambiente. Las consideraciones para la lectura son:

- La distancia de lectura no sobrepasa los 50 [cm], por la posición física.
- Se realizaran cuatro lecturas, dando un tiempo máximo de adquisición de 212 [ms], a 50 [cm] con tiempo entre lectura de 50 [ms].

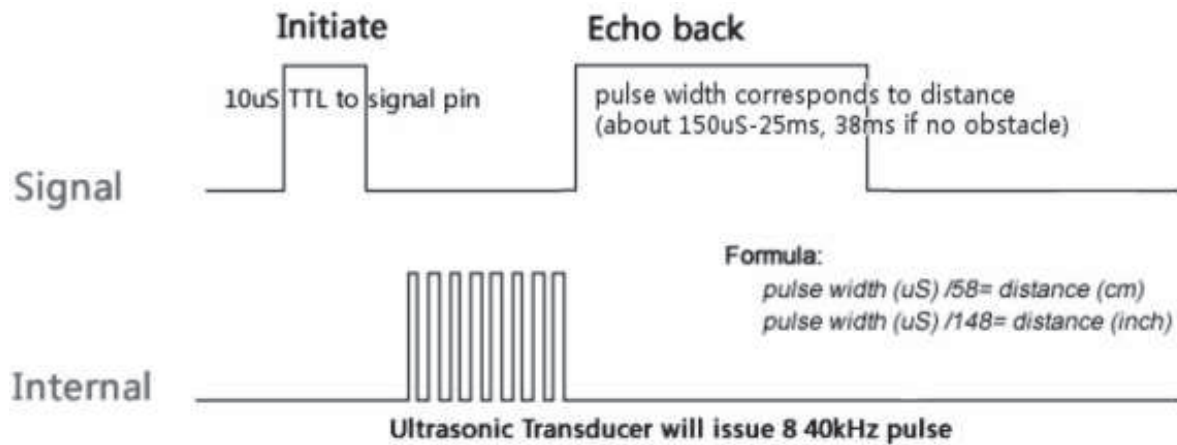


Figura 110. Diagrama de evolución sensor ultrasónico HCSR04, datasheet fabricante Cytron technologies, 2016

Para la lectura se seguirá un algoritmo de selección de picos y se hará el promedio de las dos restantes, de la función sigue el siguiente diagrama de la Figura 111.

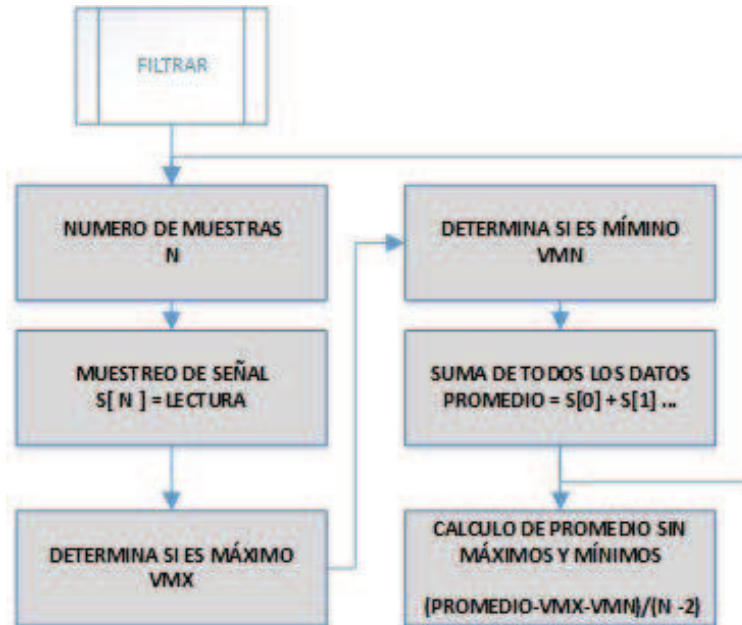


Figura 111. Pseudo código para filtrado por promedio, sin picos

4.6.2 Módulo de dosificación del floculante

La definición del módulo de mezclado a partir de su implantación en microcontrolador para su sistema electrónico y sus funciones de operación se desarrolla en cinco partes:

1. Se definen la relación de funciones del proceso con los puertos de entrada y salida que requiere por componentes en el microcontrolador.
2. Se da estructura al módulo a partir del modelo genérico de la Figura 78, para esclavo.
3. Se establece el tipo de lectura de la bomba y su control.
4. Se establecen su bloque funcional a partir de su SFC con variables asociadas a la estructura de funciones de la UC.
5. Se desarrolla el sistema electrónico con las características descritas en cap. 4.2.1.

Por las especificaciones, la bomba no cuenta con un sensor, además, la bomba dosifica una cantidad continua por cada giro al ser periferia estática por lo que se adaptó un encóder en su eje con un sensor de herradura; para determinar su función se realizaron distintas pruebas para determinar el comportamiento de dosificado. En la Figura 112 se muestra el modelo propuesto y en la Tabla 65 se muestra las pruebas realizadas.



Figura 112. Sensor propuesto para bomba de dosificación efluente

Tabla 65. Pruebas de dosificación para caracterización de encóder	
Referencia de ml	Pulsos de encóder
0.2	13
1	69
2	139
3	209
5	351
10	701
20	1401

De las pruebas se obtuvo por regresión dos curvas que se adaptara al comportamiento para hacer las pruebas con las demás bombas, en la Figura 113 se muestra las curvas con sus ecuaciones.

Pruebas de lectura de dosificación con sensor propuesto

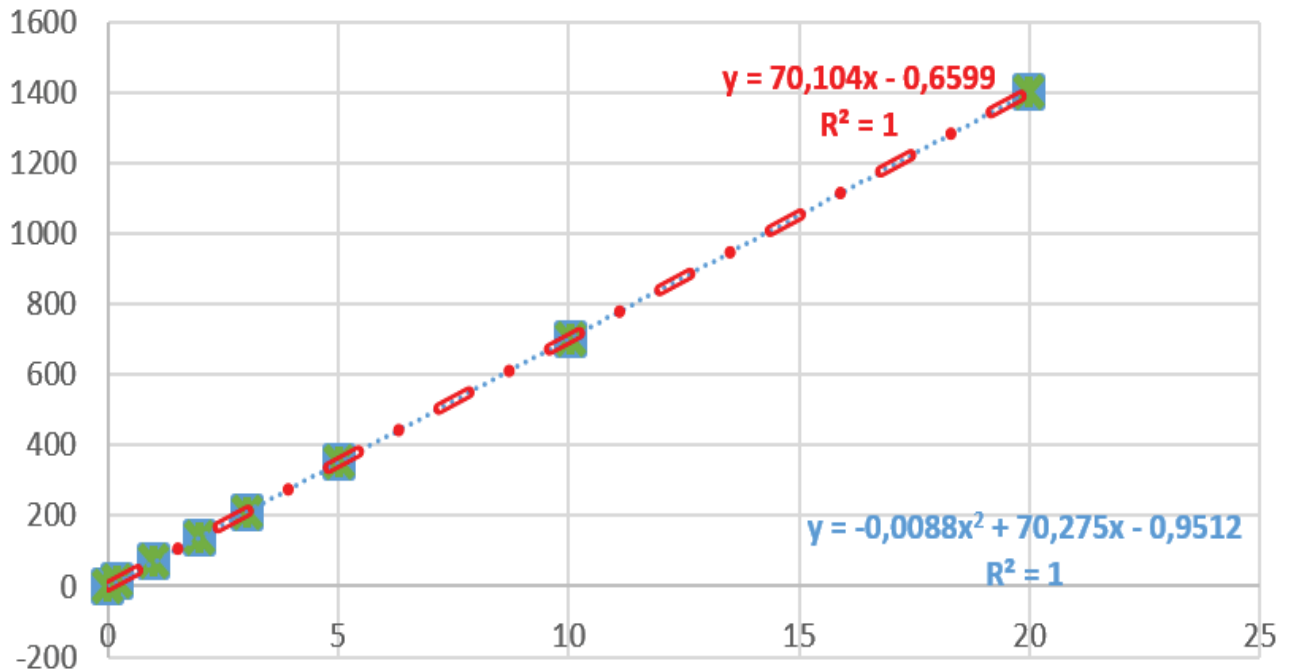


Figura 113. Ecuaciones por regresión para caracterización de sensor

En la Tabla 66 se muestran las comparaciones entre los cálculos por las ecuaciones, se realizaron pruebas con la ecuación polinómica por tener un valor más cercano al esperado, para la pruebas se programa la ecuación y se solicitó una dosificación de referencia, los resultados fueron medidos por jeringas de 5 y 10 [ml]. En cuanto a la lectura, se tubo mediciones con error al mantener un ON-OFF provocado por la inercia del motor y del fluido en las vías; para reducir este efecto se usó una ley de control basado en el error por la raíz del error, el comportamiento fue un decremento de velocidad para el extremo, teniendo como resultado un error fue de ± 0.3 [ml], equivalentes a dos gotas.

Tabla 66. Pruebas de dosificación por ecuaciones de regresión

Referencia [ml]	Cálculo lineal	Cálculo polinómica	Esperado encóder	Medido [ml]
0.2	13.36	13.1	13	0.3
3	209.65	209.79	209	3.1
5	349.86	350.2	351	4.9
10	700.38	700.91	701	10.2

Los resultados fueron presentados al equipo de trabajo el cual aprobó el uso del sensor y tipo de lectura para las demás bombas, el equipo mecánico se encargó de diseñar el soporte del sensor. De esta forma se determinó el número de pines de entrada y salida que requiere el microcontrolador, Tabla 46. Se define por los puertos de I/O de forma general, en Tabla 47 y 48 se presentan los puertos de entrada, de salida y funciones que requiere el módulo de polímero.

Por las operaciones del proceso, cada bomba es activada una a la vez, por lo que esta condición permite usar el microcontrolador mediante expansión de PWM como en el módulo 1, mientras que para la lectura, se usa un selector de entradas, para mantener la lectura por interrupción de cambio de flanco. En la Tabla 67 se establece los componentes para la estructura del módulo.

Tabla 67. Módulo floculante				
Sistema	SFC	Entrada / Salida	Función micro	Puerto del micro
Polímero	SFC6: S3b1_M7-M10 SFC7: S3b2_M7-M10	Encóder	Lectura Interrupción Timer 1	B0, E0,E1
		M6	PWM	C2,D0-D3
		Dirección	Escritura	D4-D7
Del módulo	Comunicación	Maestro	I2C	C3,C4
		Tarea	Escritura	C1
		Error	Escritura	C0
		Indicador	Escritura	C5
		Monitor	RS232	C6, C7

En la Figura 114 se representa la estructura del módulo en cuanto a sus características físicas de acondicionamiento, puertos de comunicación, indicadores y fuentes.

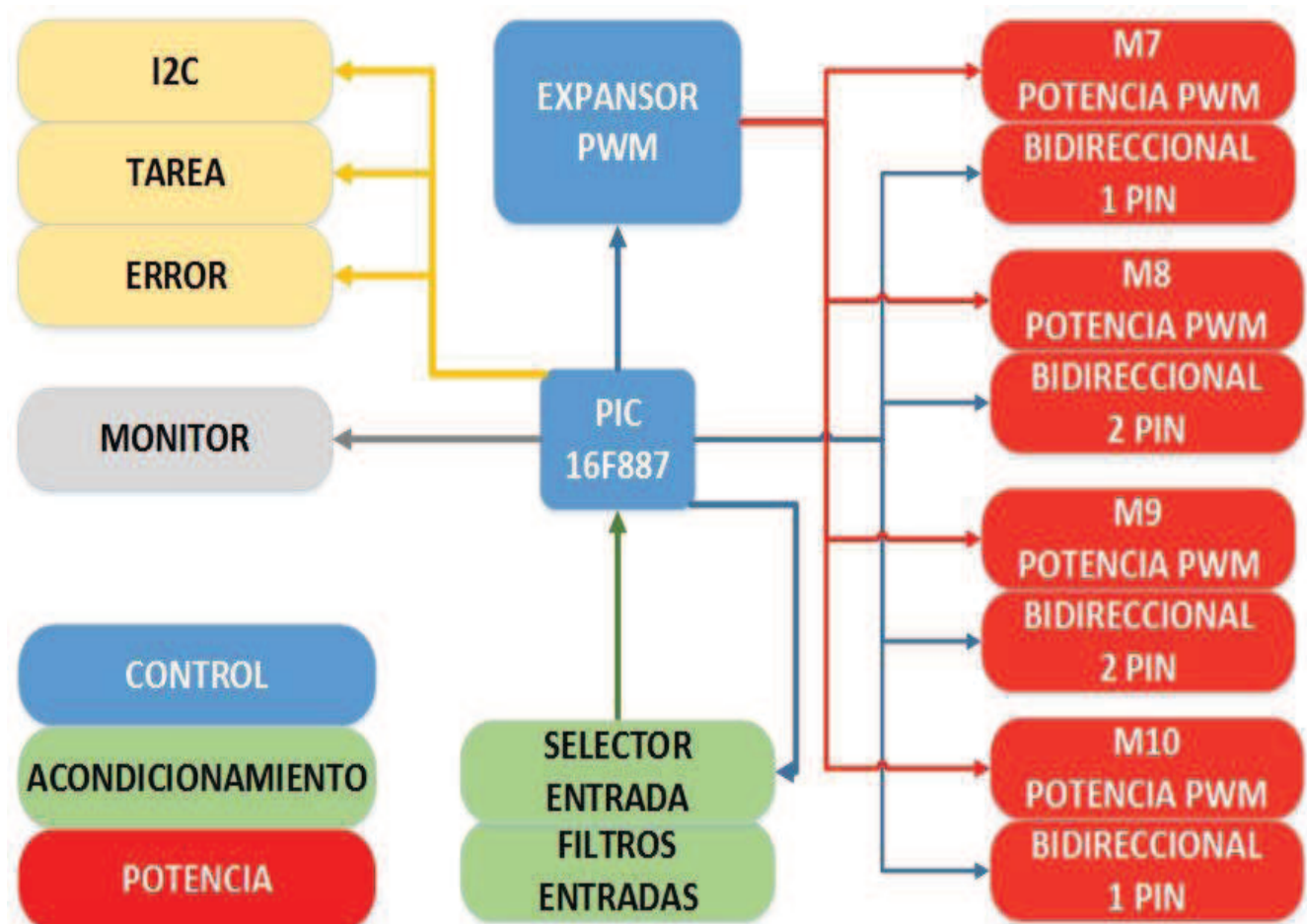


Figura 114. Estructura módulo 4

4.6.3 Bloques funcionales

Durante el desarrollo del análisis del proceso se definieron las funciones de operación para los módulos 3 y 4, Figura 58; se desarrollan sus bloques funcionales en las Figuras 115-117.

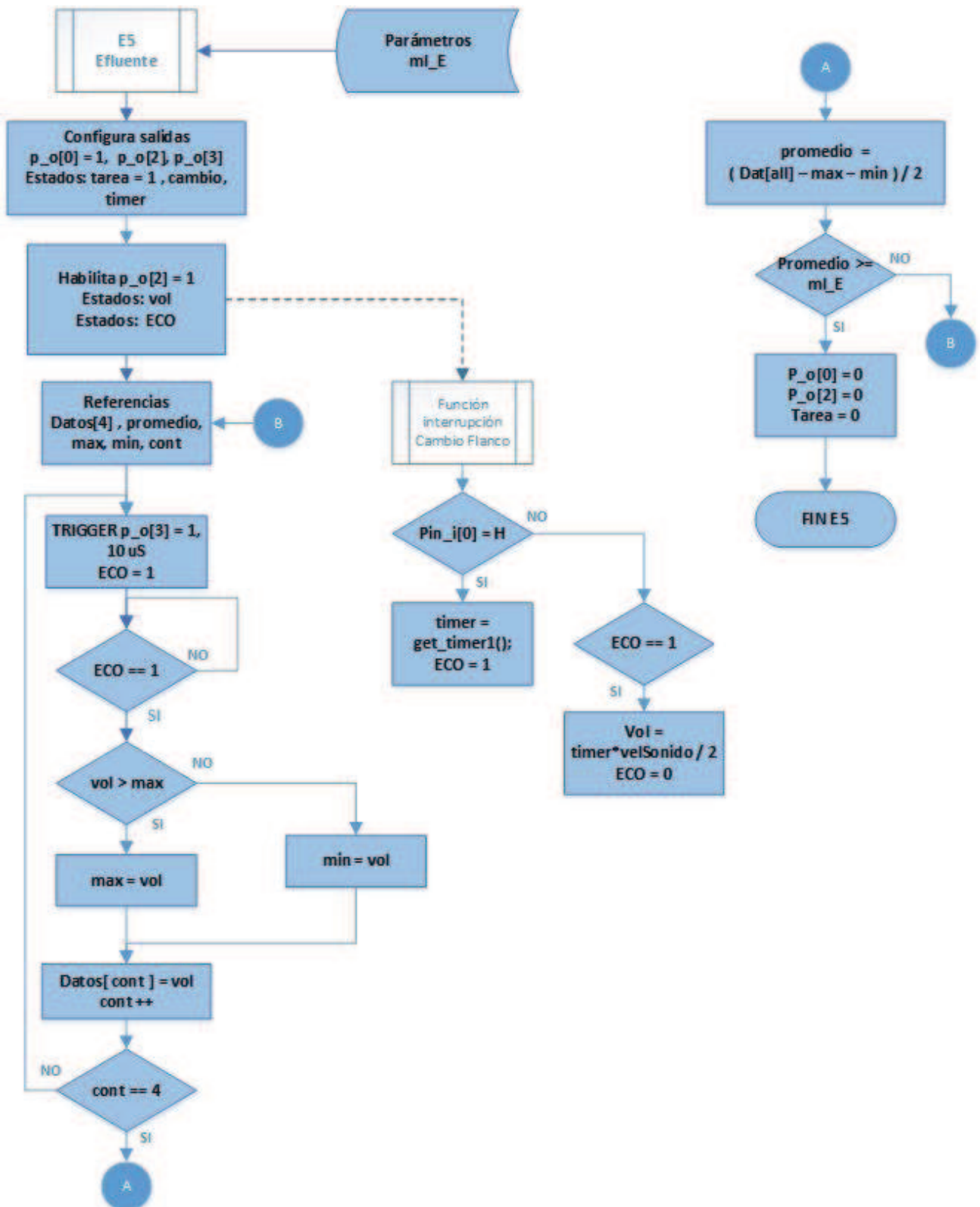


Figura 115. Módulo 3, Bloque funcional, SFC 5

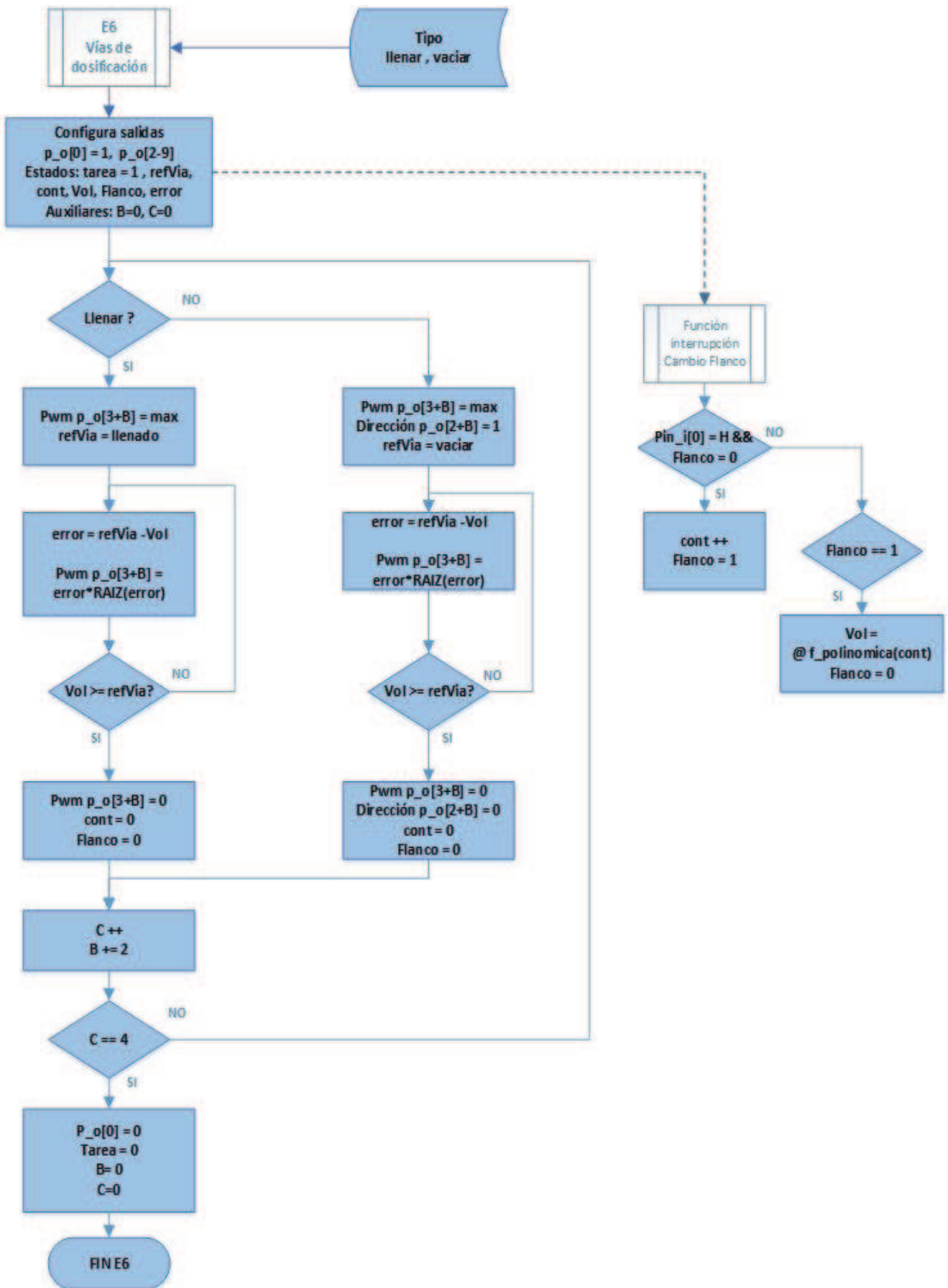


Figura 116. Módulo 4, Bloque funcional, SFC 6

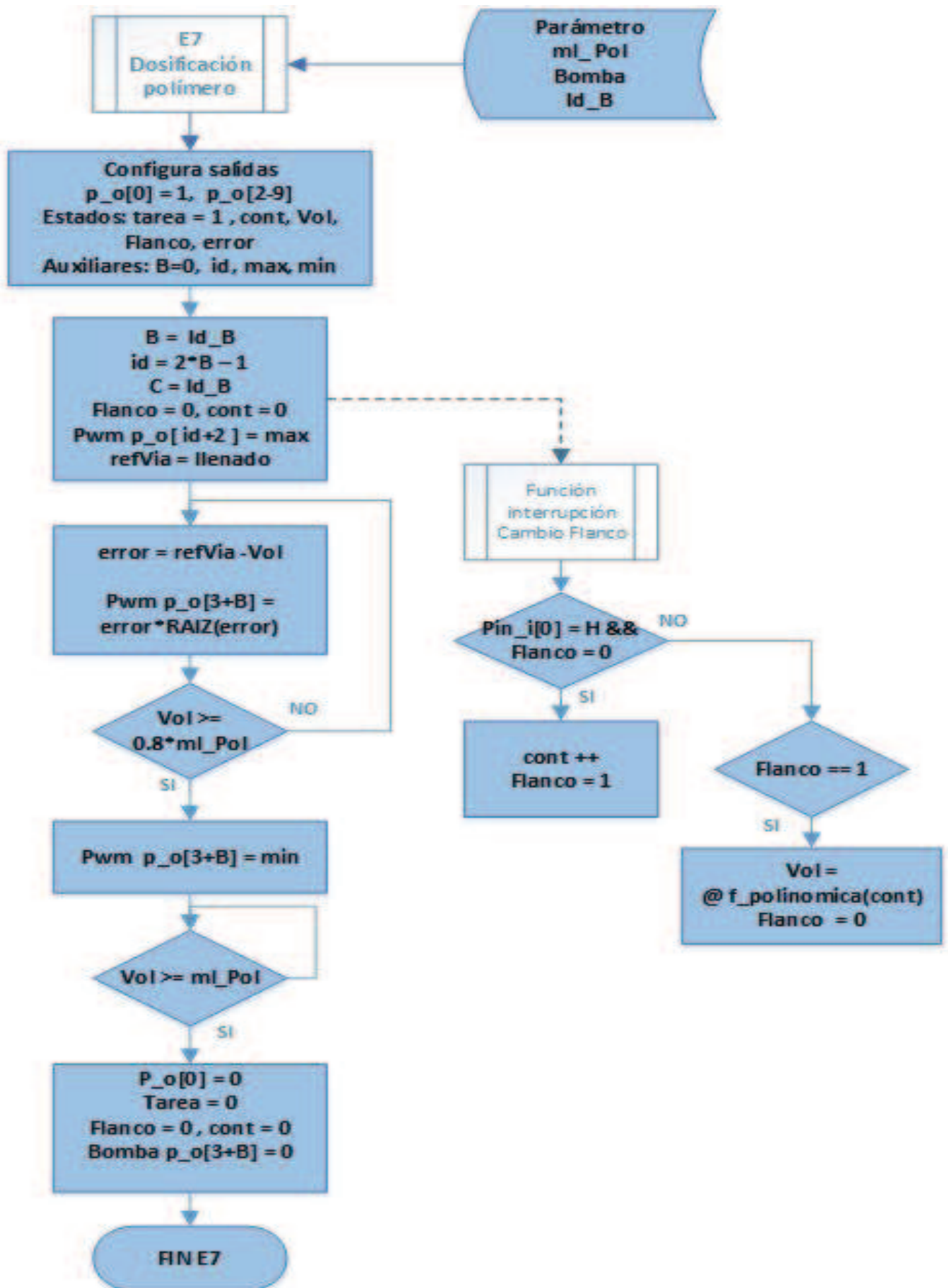


Figura 117. Módulo 4, Bloque funcional, SFC 7

4.6.4 Sistemas electrónicos

De las Tablas 64 y 67 se establecieron los puertos de los microcontroladores para los módulos, en relación a los componentes, ambos requieren acondicionamiento para sensores y potencia, mientras que el módulo de efluente tiene menos componentes por estar dedicado para un solo actuador, como el módulo 2, el módulo 4 requiere una gran cantidad de componentes.

Acondicionamiento sensores

Para el módulo de lodo, el dispositivo ultrasónico tiene implementado en su placa el acondicionamiento. Para el módulo de polímero, se tienen sensores de herradura para encóder, por lo que se usara comparadores mediante OPAM LM324, como en los otros módulos, además se usará un DEMUX para seleccionar el sensor en la entrada del microcontrolador. En el caso del módulo maestro, las entradas de error eran unidas mediante OR, la diferencia en este caso es que se requiere leer uno sin que los otros tres interfieran. El dispositivo a usar es el 74LS171, solo requiere resistencias pull-down para eliminar triestados

Acondicionamiento potencia

Para el módulo de lodo, solo se requiere controla por ON-OFF la activación de la bomba; por las especificaciones de consumo voltaje de operación, se aislara su activación mediante relevador pues no estará encendido en periodos largos de tiempo, y para su habilitación se usará un puente h a manera de reducir efectos por ser una bobina, el dispositivo a usar será un puente H L298 por condiciones de material.

Para el módulo de polímero, se requiere 4 señales de PWM, como en el caso del módulo 1, se usará un MUX 74LS157 para expandir las señales, además bajo las condiciones de operación no se activa más de una bomba a la vez. Se requiere resistencias pull-down para eliminar triestados. Para la etapa de potencia se usa dos puentes H L298. Para las direcciones, se usa compuertas NOT 74LS04 para reducir nmero de pines y queda en función del PWM para movimiento.

Acondicionamientos en puertos del microcontrolador

Para los elementos de comunicación solo se requieren resistencias pull-up para las líneas de comunicación de I2C y pull-down para las líneas de RS232, de Tarea y de Error con el fin de eliminar los triestados. Las entradas y salidas del senado y la potencia no requieren elementos adicionales.

En la Tabla 68 se pone los materiales para las etapas de acondicionamiento, potencia y de control para ambos módulos.

Tabla 68. Componentes del sistema electrónico, Módulo 3 y 4			
Etapas	componentes	Dispositivo	Valor
Acondicionamiento	Encóder – módulo 4	OPAM	LL324 x 2
		Trimpot	5 [K Ω] x 4
		Resistencia	470 [Ω] x 4 10 [K Ω] x 4
		DEMUX	74LS171
		Resistencia	1200 [Ω] x 4

Potencia	Bomba – efluente	Puente H	L298
		Diodo	1N5400 x 3
		Relevador	5 [V]
		Capacitor electrolítico	100 [uF]
	Bomba – floculante	Puente H	L298 x 2
		Diodo	1n5400 x 16
		MUX	74LS157
		NOT	74LS04
Control	Microcontrolador	PIC	16F887
		Cristal oscilador	20 [MHz]
		Resistencia	10 [KΩ] x 1 2.2 [KΩ] x2 470 [KΩ] x 4
		Capacitor cerámico	22 [nF] x 2
		Push botton	4 pines
		LED	Verdes,3 [mm] x 4
Misceláneos	Conectores	T-Bloc	12
		Header macho	Varios
		Jumper	Varios
	Fuentes	Capacitor electrolítico	100 [uF] x 3
	pull-down	Resistencias	470 [Ω], varios

En las Figuras 118, 119 y 120 se muestran el esquemático del sistema electrónico del módulo 4; en las Figuras 121 y 122 se muestra los esquemáticos para el sistema de control del módulo 3.

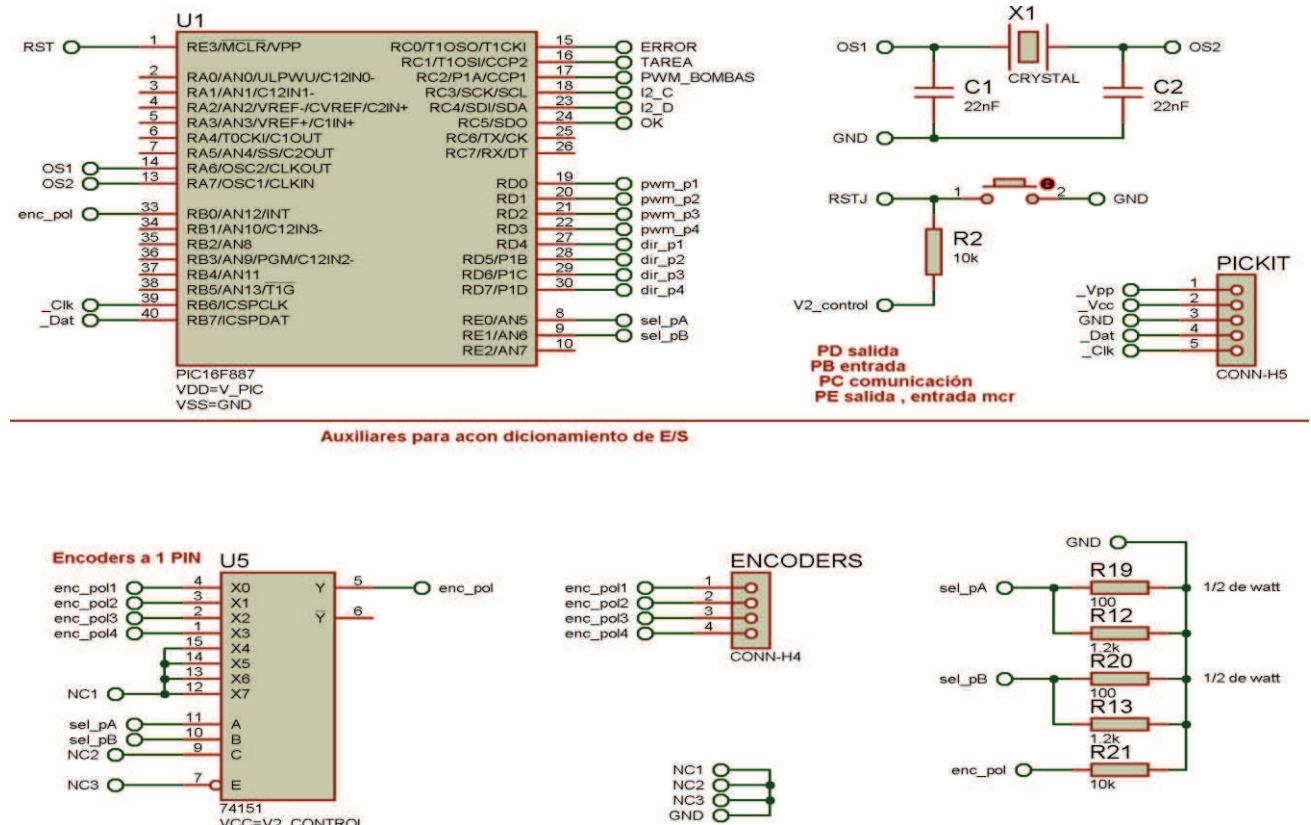


Figura 118. Módulo 4, sistema electrónico, etapa de control

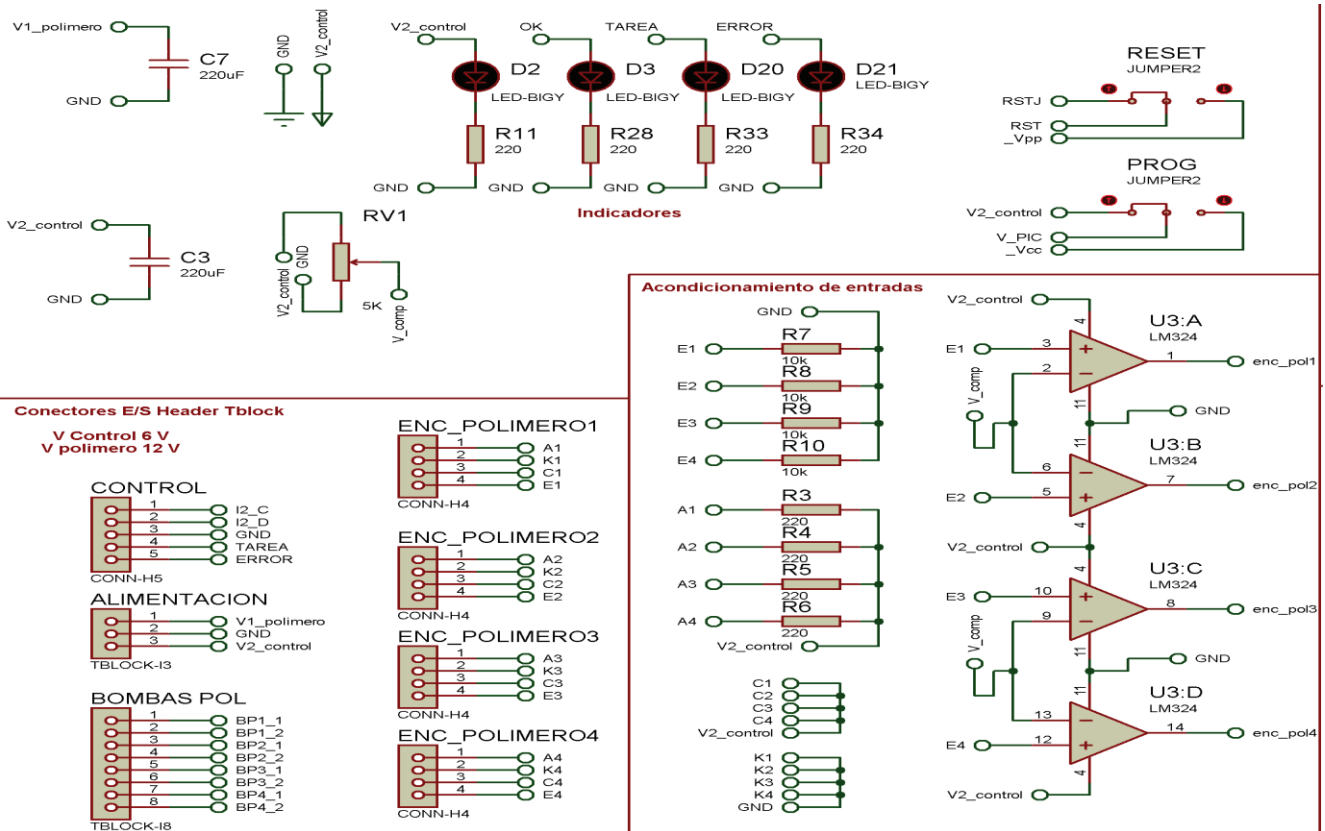


Figura 119. Módulo 4, sistema electrónico, etapa de acondicionamiento

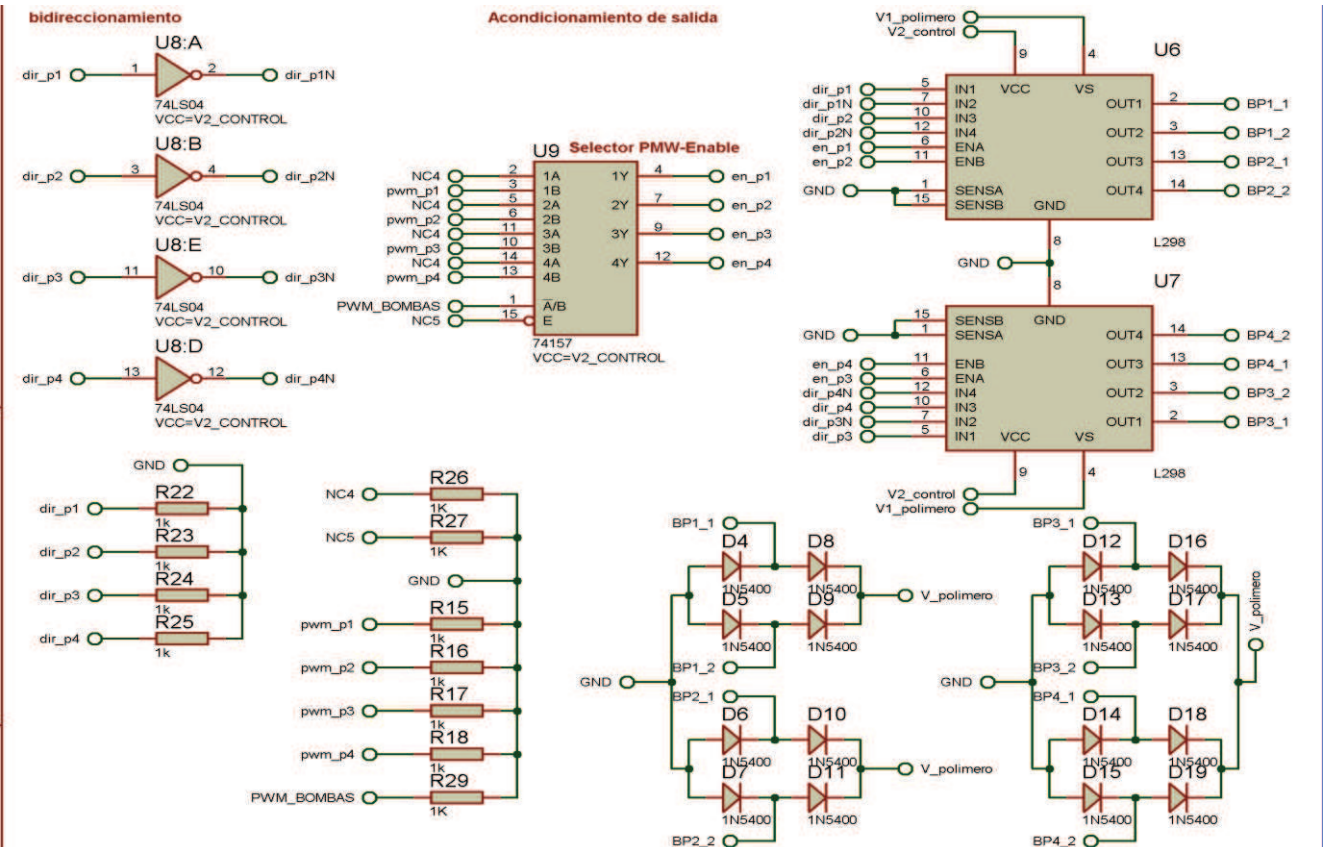


Figura 120. Módulo 4, sistema electrónico, etapa de potencia

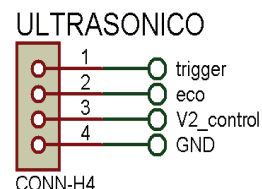
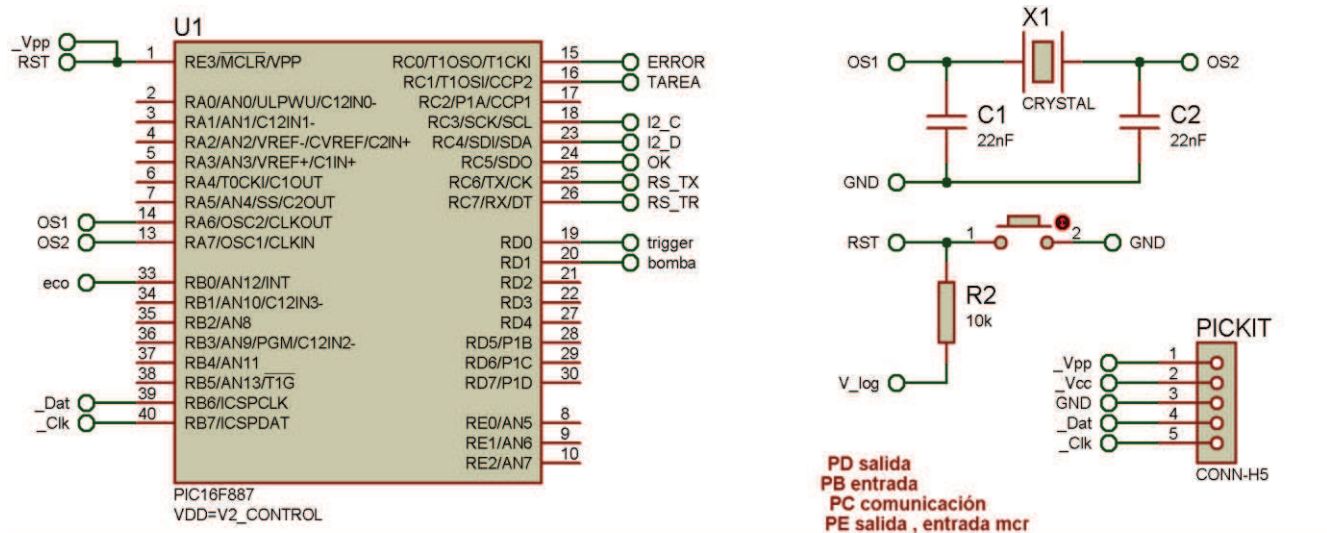


Figura 121. Módulo 3, sistema electrónico, etapa de potencia



CAP. 5 PRUEBA DE MODELO FUNCIONAL

El presente trabajo desarrolla una unidad de control bajo el concepto de modular para automatización de procesos, cubre el desarrollo de tres etapas del proyecto **LODOS**, auxiliar en laboratorio para el tratamiento de aguas residuales. A partir de la combinación de dos metodologías, una orientada al producto y otra orientada a componentes del proceso se estructuró una metodología de diseño para automatización basada en microcontroladores, con la separación y distinción de operaciones del proceso, de operaciones de la unidad de control y de comunicación, lo que da independencia a la implantación del sistema y permite generar un modelo genérico para sistemas modulares.

Las pruebas del modelo funcional están distribuidas en dos partes, la prueba de módulos de forma independiente en su sistema electrónico y la ejecución de sus tareas, y la prueba de las operaciones del proceso de la unidad de control. Cabe aclarar que se realizan pruebas en PCB donde la unión de los sistemas electrónicos de control y su separación de sus elementos de acondicionamiento fue para realizar pruebas y validar el modelo funcional. Se espera que para las siguientes etapas de diseño se particularicen en módulos individuales para formar el prototipo. Por otra parte, las pruebas fueron sobre el modelo mecánico para la fecha de fin de Fase de trabajo, julio 2015 y algunas fueron realizadas en etapas anteriores. Además, se limita el desarrollo a las funciones principales por condiciones de privacidad del proyecto.

5.1 Prueba de módulos

Para la primera parte solo se corrobora el funcionamiento de los sistemas electrónicos, pues en las Fases de trabajo se establecieron las condiciones de acondicionamiento para su funcionamiento. Cada módulo es probado de forma individual en sus etapas de acondicionamiento, mientras que para la parte de control se realiza la ejecución de tareas sin la estructura de comunicación, esto mediante los comandos de tarea por el monitor serial; este modo permite mantenimiento a nivel programador y su implementación fue independiente al desarrollado por ser una característica particular de programación. Las pruebas con la estructura de comunicación serán la segunda parte de este capítulo.

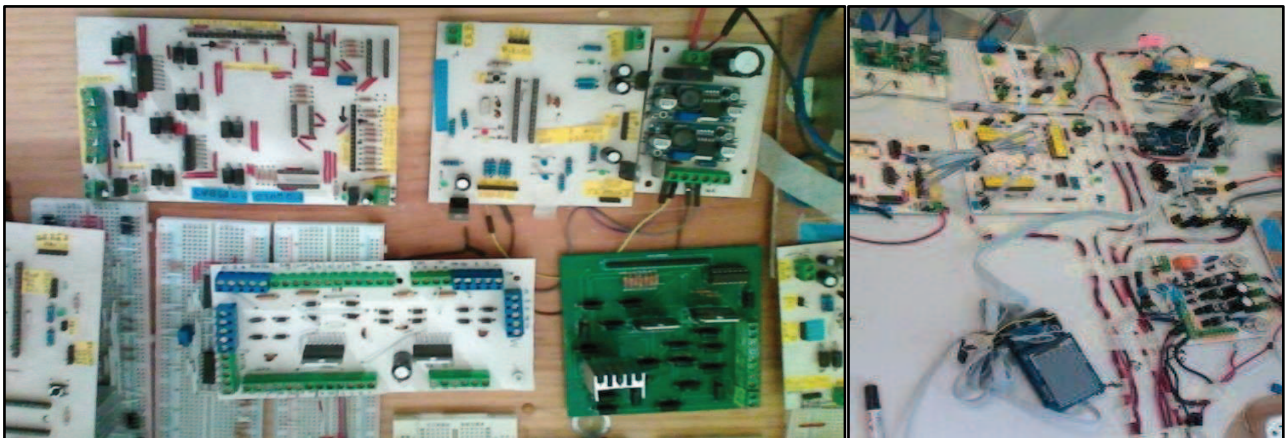


Figura 123. Sistema electrónico de la unidad de control, general

En la Figura 123, a la derecha se muestra el modelo funcional de los PCB de los sistemas electrónicos, su desarrollo fue a partir de su prueba en protoboard siguiendo sus estructuras de los sistemas electrónicos; de lado izquierdo se muestran los PCB por separado en etapa de acondicionamiento y control, formado por las placas de cada módulo, líneas de comunicación para el

monitoreo individual y las fuentes de voltaje. Como se mencionó, estas fuentes no fueron desarrolladas en este trabajo, se sigue un modelo genérico de rectificación por transformador y capacitor. Se espera que para las siguientes etapas el equipo de trabajo de electrónica se centre en diseño por fuentes conmutadas.

Las pruebas realizadas para los módulos tienen las siguientes consideraciones:

- La fuente usada fue separada en tres partes: para potencia, para operaciones de control, para acondicionado. Se desarrolló por parte de otros miembros de equipo de electrónica.
- Antes de la fabricación de los PCB, fueron probados en protoboard, estas pruebas fueron realizadas durante las Fases de trabajo en los sistemas electrónicos, las pruebas se menciona en este punto para probar el modelo funcional en conjunto.
- Para cada módulo se usa el monitor serial a manera de interpretar las operaciones de la estructura de comunicación y mostrar las operaciones de cada módulo. Para las pruebas individuales se usa este monitor para ejecutar lecturas o movimientos. Este modo es particular y no se describe en este trabajo por condiciones del proyecto.
- Algunos detalles no son presentados por las condiciones de privacidad de la empresa.

5.1.1 Sensores

Se comprueba las etapas de acondicionado para los módulos, los valores esperados para esta prueba es la de posicionamiento y caídas de voltaje. Para el módulo de batidora no hace falta verificar el funcionamiento pues su circuito de acondicionamiento está integrado al motor y para las pruebas del desarrollo del controlador fue verificado.

Módulo estructura. Cuenta con 2 encóder y 6 botones para posicionamientos. Cada uno es acondicionado para entregar valores lógicos entre 0 y 5 [V]. En la Figura 124 se muestra de lado izquierdo la placa de sensores. Se La palca fue desarrollada de forma independiente para realizar las pruebas. En la misma figura, la palca verde es de acondicionamiento de potencia para el módulo.

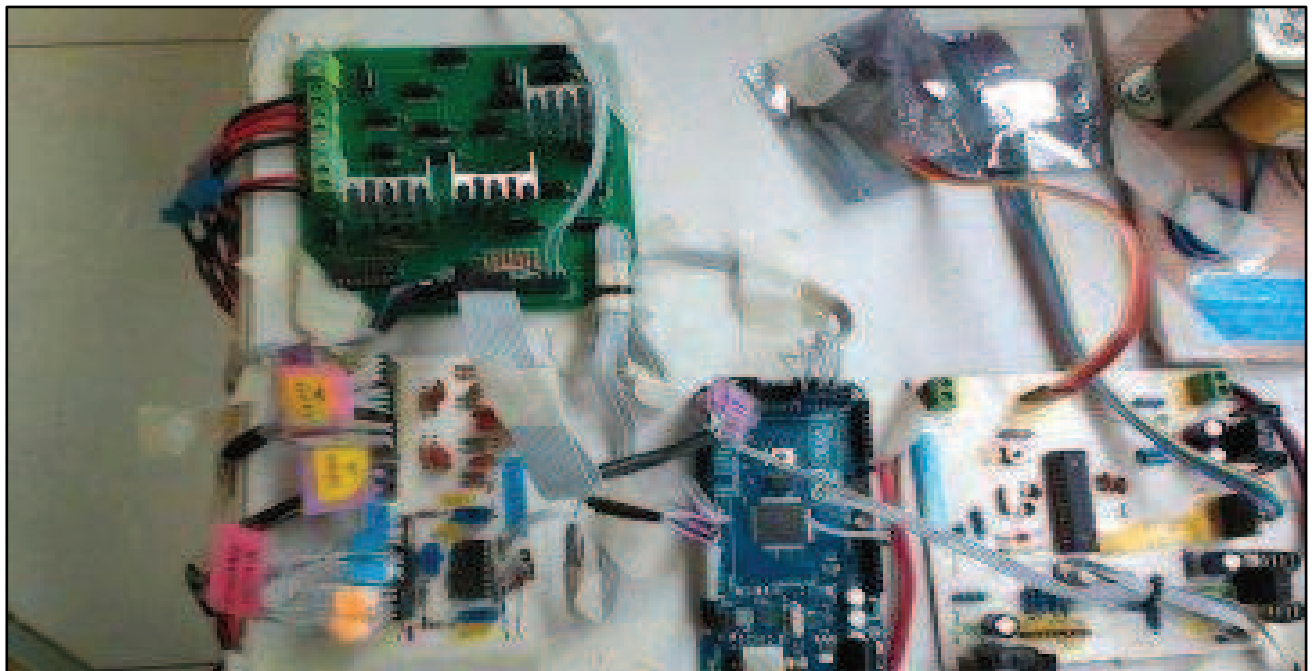


Figura 124. Módulo 1, placas de acondicionamientos

De las funciones de monitoreo, se leyeron los estados mientras se probó la palca, Figura 125. De lado izquierdo se muestra los componentes de la placa y de lado derecho las instrucciones en el monitoreo, donde se cambian las activaciones de las entradas y se recibe los estados de lectura.

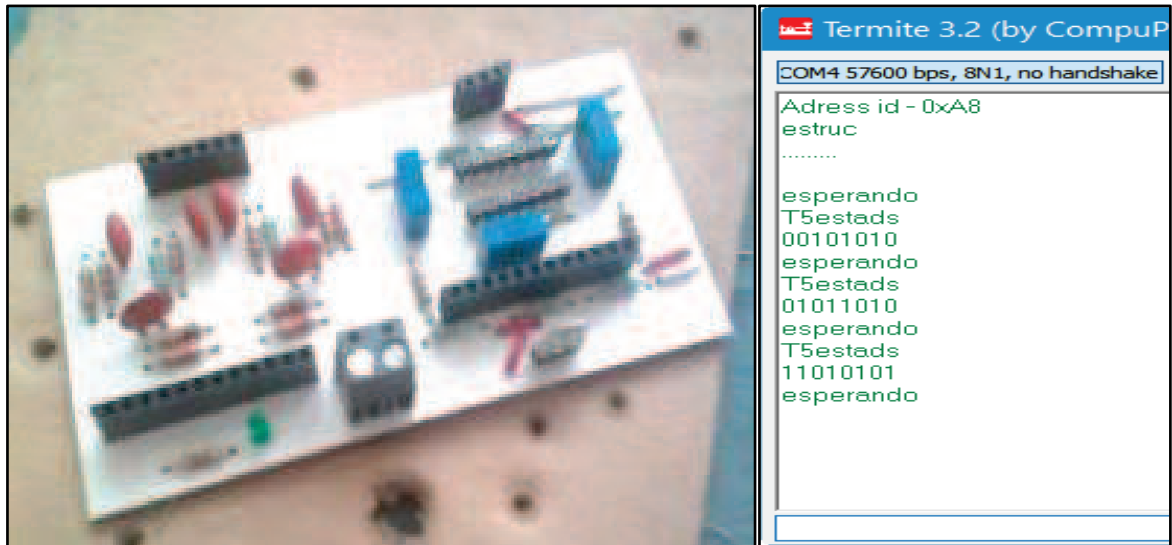


Figura 125. Módulo 1, pruebas de sensado

Los resultados de las pruebas mostraron una caída de voltaje para los encoders, de la hoja de especificaciones del OPAM LM324, indican que la salida tiene una caída de voltaje de 1.5 [V] para una entrada de 5 [V], considerando el inicio de la estructura, por otra parte las caídas de voltaje en los señores de posición mecánicos es despreciable.

Módulo 3. Cuenta con un sensor ultrasónico para el nivel de volumen. El acondicionamiento se auxilia por software mediante promedios. En la Figura 127 se muestra pruebas mediante el monitor, de lado izquierdo el sensor y el indicador son habilitados para lectura de distancia; en el centro se pone un objeto y el indicador es apagado; de lado derecho se muestran los comandos del monitor, la distancia probada fue de 37 [cm], el objeto fue puesto en posición y termina hasta que la distancia alcanzada.

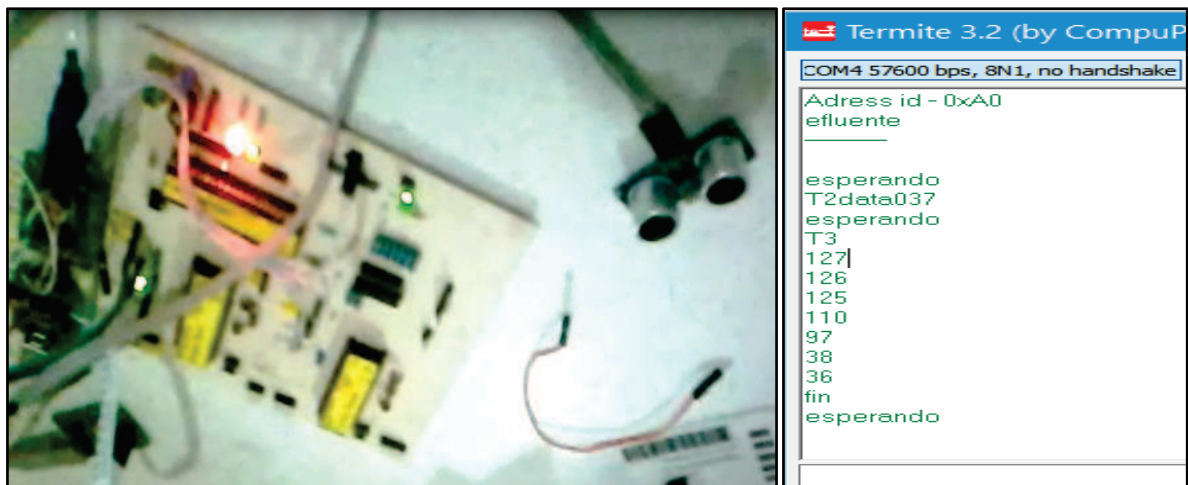


Figura 126. Módulo 3, pruebas de sensado

En la Figura 127 se muestra as pruebas para determinar la ventaja del acondicionado por promedios, de lado izquierdo se muestra la configuración de sensado de forma directa, se posiciona

un objeto a 30 [cm], tendiendo variaciones en la distancia medida; de lado derecho se muestra las lecturas de 24 [cm] mediante el acondicionamiento auxiliar, la variación es más estable, las consideraciones para este acondicionado están descrito en la Fase de desarrollo, cap. 4.6.1. Las caídas de voltaje son depreciables debido a que la palca del sensor esta acondicionado para lectura digital.

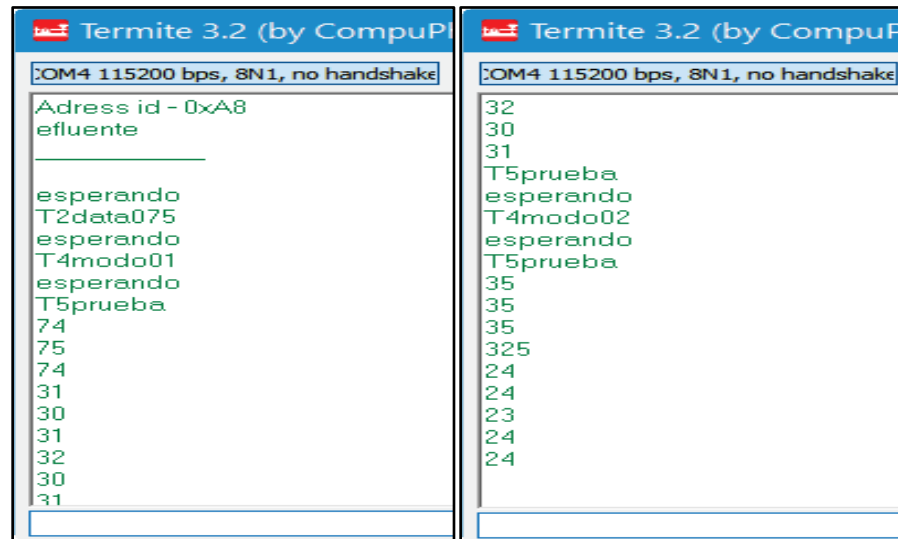


Figura 127. Módulo 3, pruebas de lectura del sensor ultrasónico

Módulo 4. Los sensores del módulo son 4 encoders que fueron propuestos y probados para la dosificación de efluente, las pruebas se realizaron para determinar la exactitud de dosificación, en el cap. 4.6.2 se describen la propuesta del uso de encóder, con un error de ± 2 gotas equivalente a 0.2 [ml]. En la Figura 127 se muestra las pruebas de lectura, de lado izquierdo se muestra al sensor de herradura encendido; de lado derecho se muestra el monitor con el set de comandos para la lectura de cada sensor, como se ha usado un DEMUX sólo se puede leer 1 encóder a la vez.

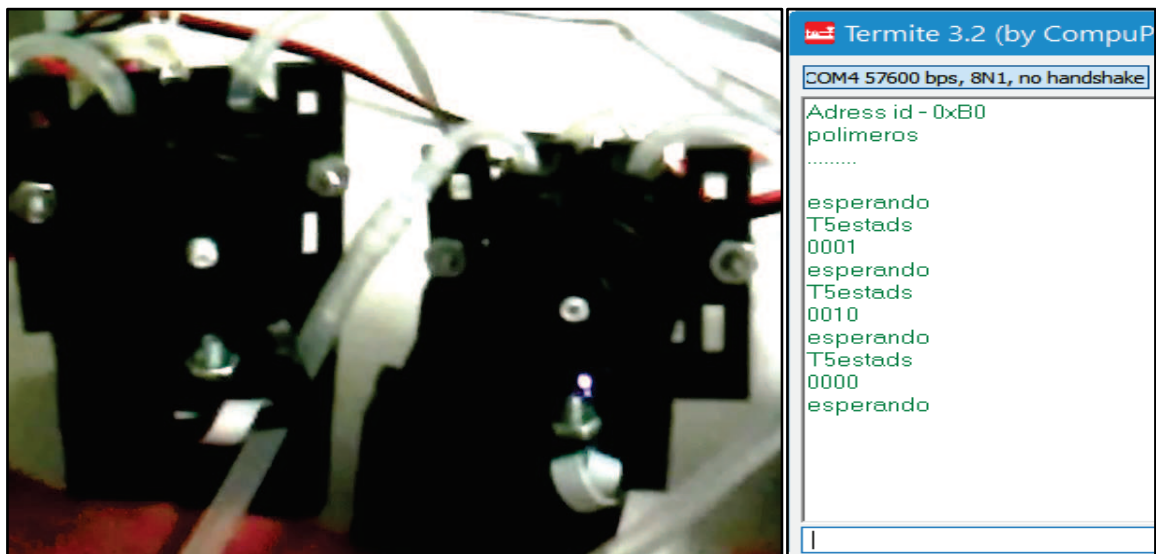


Figura 128. Módulo 4, pruebas de sensado

Las caídas de voltaje fueron proporcionales al módulo 1 pues de igual forma usa un OPAM LM324 y el DEMUX provoca de igual forma caída de voltaje con salida de hasta 3.5 [V] cuando los demás módulos son encendidos.

5.1.2 Actuadores

Para las pruebas de los actuadores, se habilitan de forma manual e individual sobre sus palcas de desarrollo, en la Figura 129 se muestra de lado izquierdo la distribución de las líneas de voltaje de la fuente, de lado derecho las placas de los actuadores con la etapa de potencia. El objetivo de estas pruebas fue de verificar el correcto funcionamiento de los circuitos planteados para los puentes H con los actuadores, cada acondicionado es descrito en su Fase de trabajo.

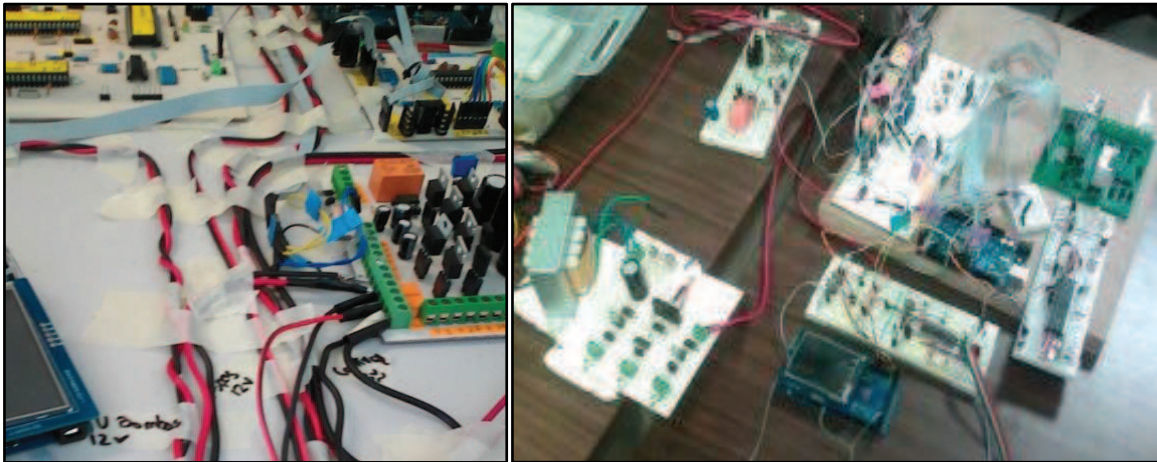


Figura 129. Distribución de fuentes y palcas de acondicionamiento de potencia

Módulo 1. Cuenta con cuatro actuadores para tres sistemas, en la Figura 130 se muestra la prueba de su placa de acondicionamiento de potencia, fue desarrollada de forma independiente para las pruebas. No se presentó errores pero hubo caídas de voltaje considerables, de 12 [V] a 9 [V]; aunque las fuentes no son definitivas, se muestra la importancia del desarrollo específico de las etapas de potencia, como se mencionó en su Fase de desarrollo, cap. 4.3.3.

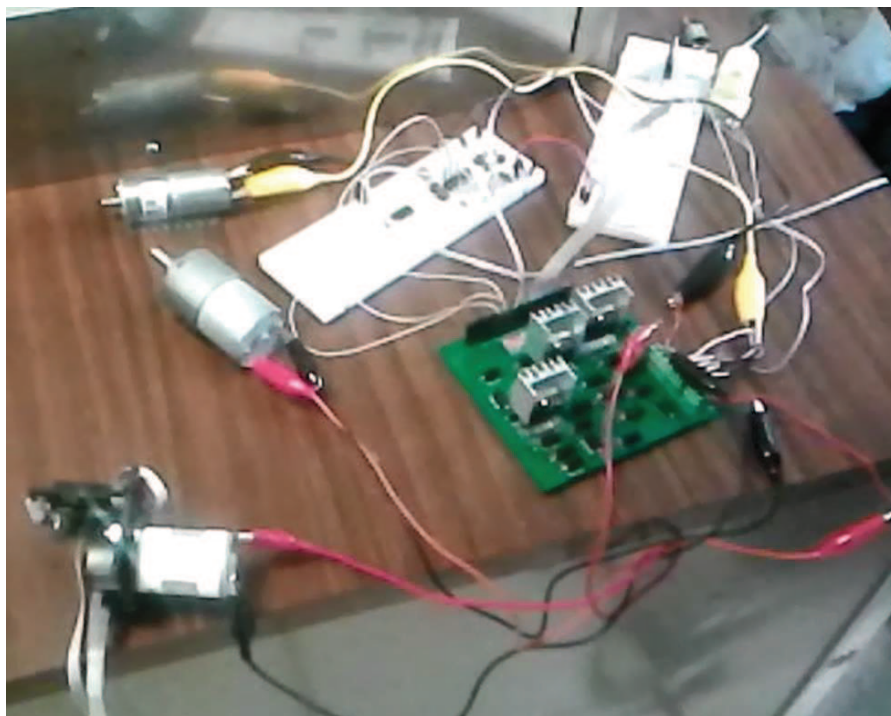


Figura 130. Módulo 1, pruebas de actuadores

Módulo 2. Las pruebas del acotador fueron realizadas en su Fase de trabajo, en la Figura 131 se muestra la habilitación por ON-OFF para el actuador. La caída de voltaje es proporcional al del módulo 1. Su componente mecánico fue terminado de desarrollar por el equipo mecánico al finalizar la etapa 3 del proyecto, por lo que las pruebas fueron sin oposición en el eje del motor.

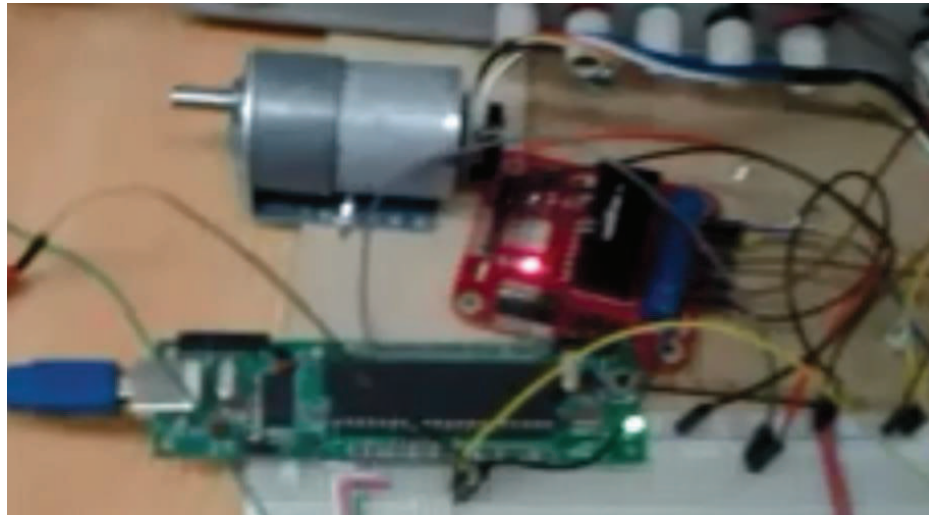


Figura 131. Módulo 2, prueba de actuador

Módulo 3. La salida para su actuador requiere un ON-OFF por relevador, fue probado por el equipo mecánico de forma independiente por las pruebas que realizaron para las línea de dosificación en la estructura mecánica, por lo que para las pruebas del sistema electrónico se aseguró la habilitación; el sistema electrónico propuesto incluyó un puente H para para aislar el relevador por ser una bobina, las caídas de voltaje no fueron significativas. En la Figura 132 se muestra su placa de lado izquierdo.

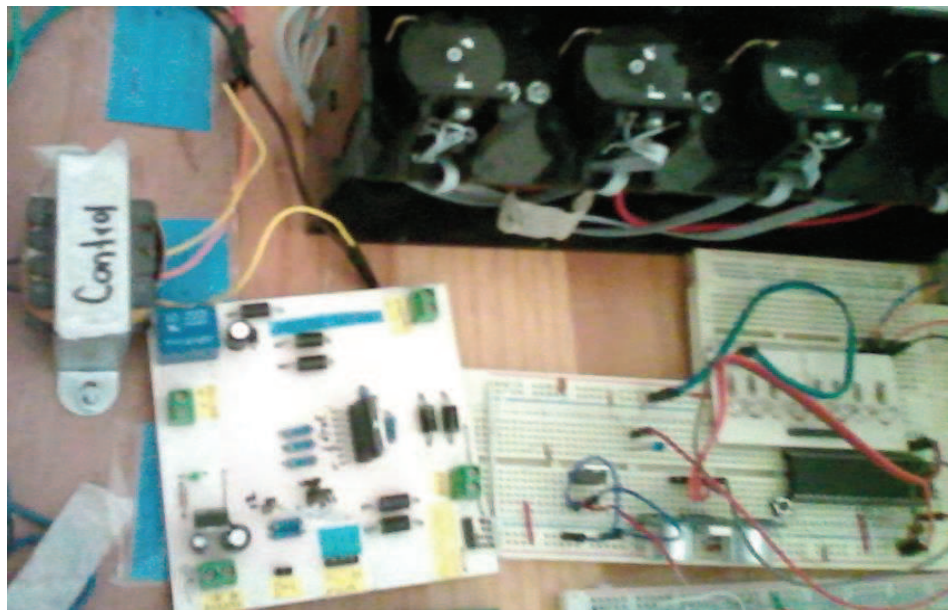


Figura 132. Módulo 3, prueba de actuadores

Módulo 4. Las pruebas para los actuadores fueron realizadas en su Fase de trabajo, en la Figura 133 se muestra la habilitación de las bombas, por condiciones de sus dispositivos electrónicos son habilitadas de una en una, las caídas de voltaje fueron proporcionales al del módulo 1.

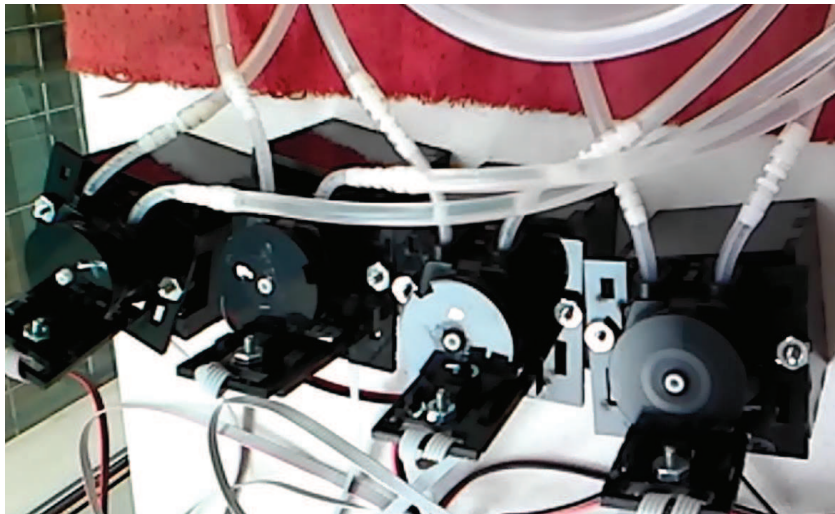


Figura 133. Módulo 4, prueba de actuadores

5.1.3 Ejecución de tareas

Después de comprobada el funcionamiento de los acondicionamientos en los sistemas electrónicos, para la ejecución de tareas de cada módulo fue mediante el monitor y de forma general para mostrar el funcionamiento en la etapa de control; en la Figura 134 se muestra en la izquierda los dispositivos convertidores seriales a USB, de lado derecho se muestra la estructura mecánica. Las consideraciones para la ejecución de las tareas son:

- Las pruebas se realizan por módulo.
- No se detalla el registro de las pruebas por condiciones de privacidad del proyecto.
- Los sets de instrucciones no se muestran para las pruebas.

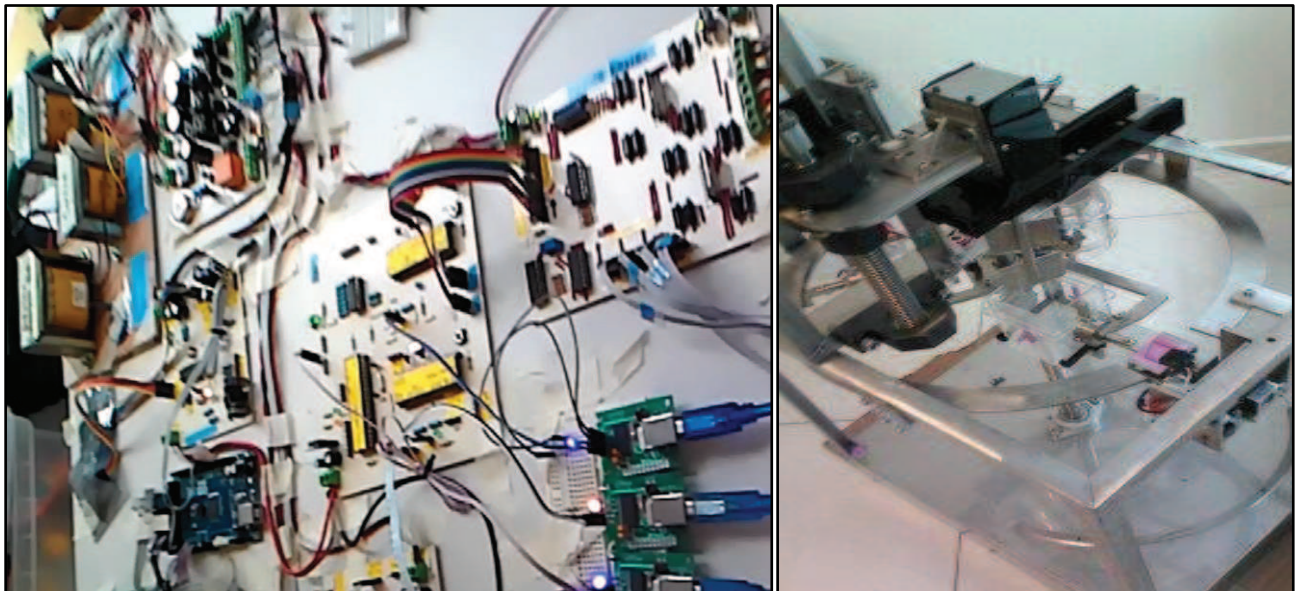


Figura 134. Unidad de control y estructura mecánica

En la Figura 135 se muestra una parte del movimiento de la estructura, para el módulo 1; equivalente a las inducciones de prueba de secuencia, donde se mueve mecánicamente para comprobar la libertad del movimiento; de lado izquierdo se orienta la posición 1C para permitir

mecánicamente el avance del sistema carrusel a la posición $\frac{1}{2}$; a la derecha se muestra el cambio de orientación en 1C para recibir mecánicamente el siguiente contenedor.

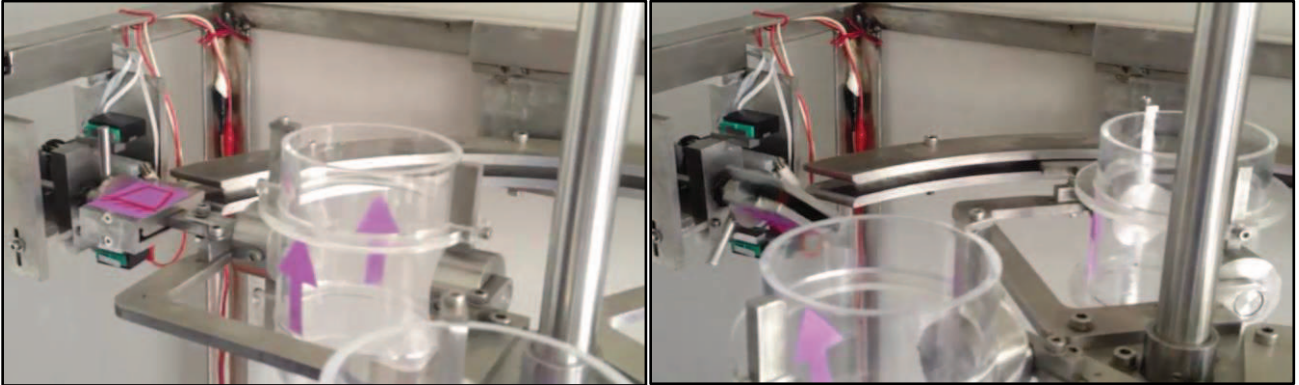


Figura 135. Módulo 1, ejecución de tarea en estructura

En la Figura 136 se muestra las pruebas de los actuadores con el sistema de control, de lado izquierdo se muestra el movimiento del actuador representativo de posicionamiento de batidora “aspas”; de lado derecho se muestra el actuador de batidora ejecutado una vez se ha posicionado. Los elementos negros en los ejes de los actuadores sirven de referencia visual para los movimientos.

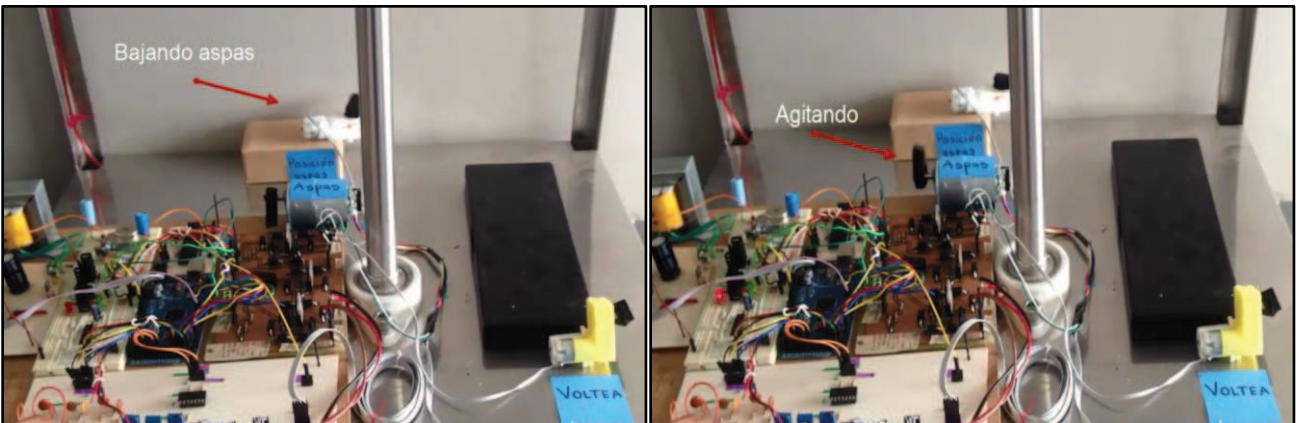


Figura 136. Ejecución de tarea, posicionamiento y mezclado

En la Figura 137, de lado izquierdo se muestra el cambio de posición de la batidora y de lado derecho el cambio de orientación del contenedor, en referencia a la Figura 136.

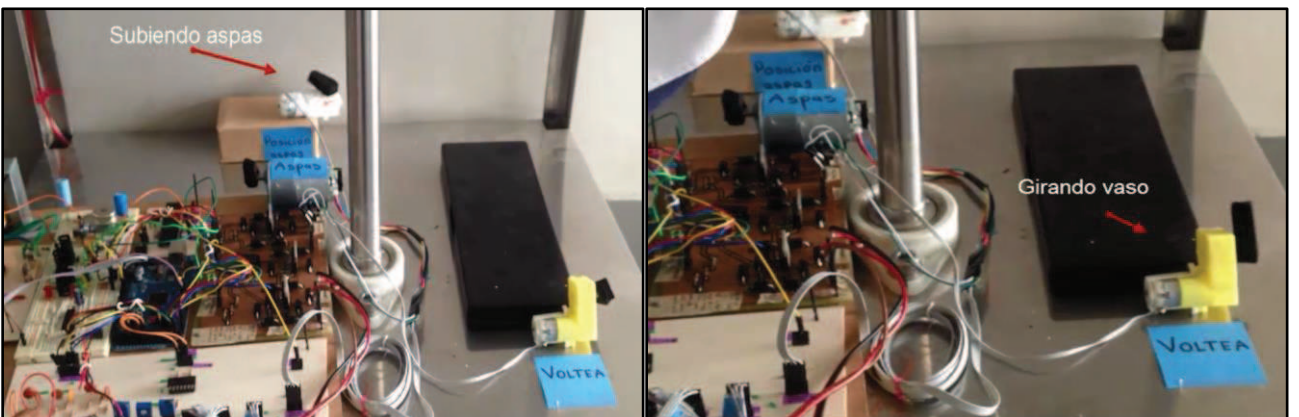


Figura 137. Ejecución de tarea, posicionamiento batidora y orientación contenedor

Al tener retroalimentación por los encoders y por los sensores mecánicos, los tiempos para la realización de la tarea depende del sistema físico y por condiciones de tiempo de operación para ejecución de proceso.

Para el módulo 3 de dosificación efluente no se realizan pruebas de tarea pues solamente es activación por ON-OFF. Para el módulo 4, las pruebas de tarea llenado/vaciado de vías son omitidas pues cumplen la misma función que dosificación, con diferente referencia y sentido. En la Figura 138 se muestra las pruebas para el dosificado de floculante, de lado izquierdo se muestran las bombas con sus encoders en una sola estructura mecánica diseñada por equipo mecánico; en la derecha se muestra las pruebas de dosificado. Los tiempos de ejecución fueron proporcionales, de 10 [s] para dosificación de 20 [ml].

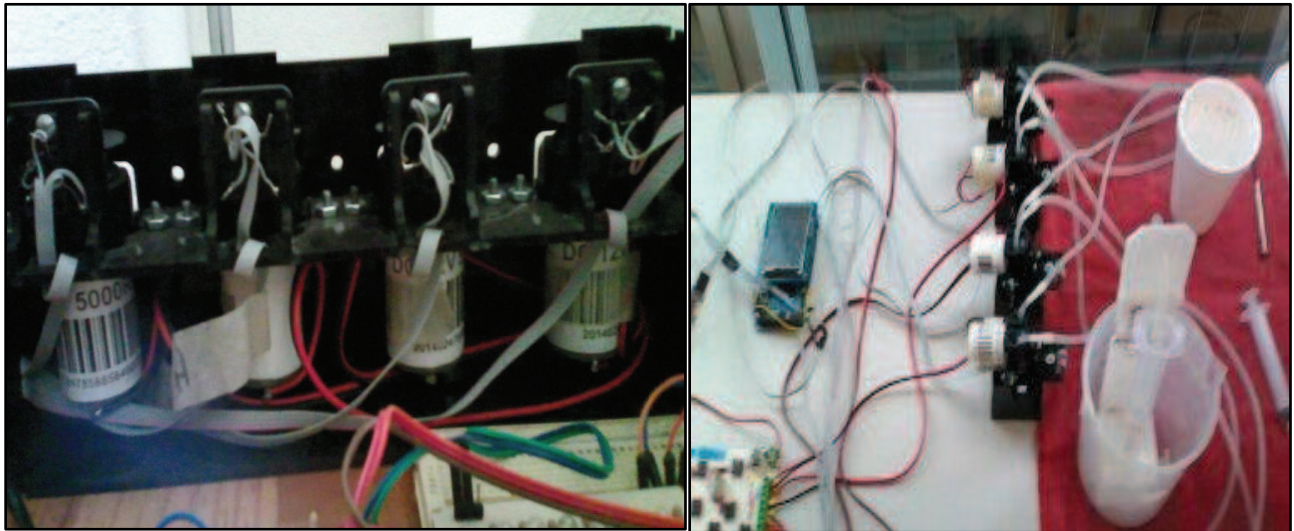


Figura 138. Ejecución de tarea, dosificación de floculante

Para la prueba de mezclado se describe en su Fase de trabajo cap. 4.5. Se comprueba la ejecución de las tareas por comando. En la Figura 139, a la izquierda y centro se muestra la lectura de la velocidad para el microcontrolador y en osciloscopio; en la derecha se muestra su ejecución. El tiempo de ejecución es por parámetro de usuario; en la transición de la función revolver a mezclado, se redujeron los efectos por los cálculos de las ganancias por tabulación para el control de velocidad.

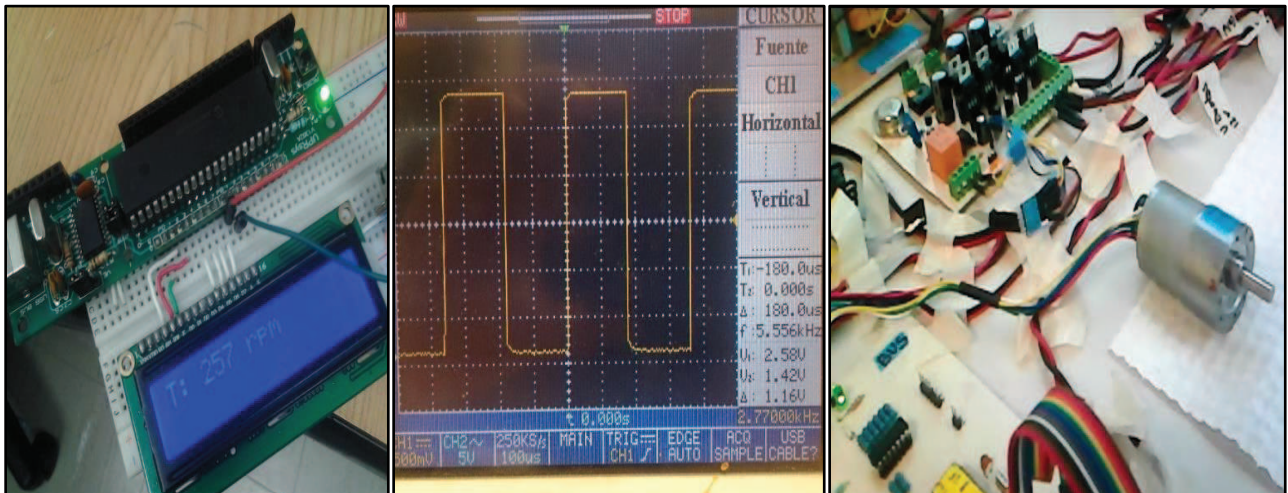


Figura 139. Ejecución de tarea, mezclado

5.2 Prueba de comunicación y de operaciones

Para las pruebas de comunicación y de operaciones se muestran las básicas que muestran el funcionamiento correcto de la unidad de control por las condiciones de privacidad del proyecto, además se toma en cuenta los siguientes puntos:

- Las pruebas de comunicación se realizan de forma general centradas en selección envío de datos, prueba de tareas y detección de errores
- Se compara parte de la simulación para selección de tarea y evento de error
- Las operaciones de configuración en la interfaz no se detallan y solo se muestran operaciones básicas, como preparación de secuencia y parámetros
- No se desarrollan tablas específicas sobre la evolución de operaciones

En la Figura 140 se muestra la distribución del sistema mecánico y de la unidad de control al finalizar las Fases de trabajo, las pruebas finales se realizan en este módulo funcional. Cabe mencionar que algunos sistemas mecánicos fueron probados hasta este punto por el tiempo de máquina, de tal forma que el desarrollo de la unidad de control termino antes que el sistema mecánico. El desarrollo de las pruebas del sistema se divide en tres partes: prueba de comunicación simulada y por monitor serial, y prueba de solicitud de tarea con error.

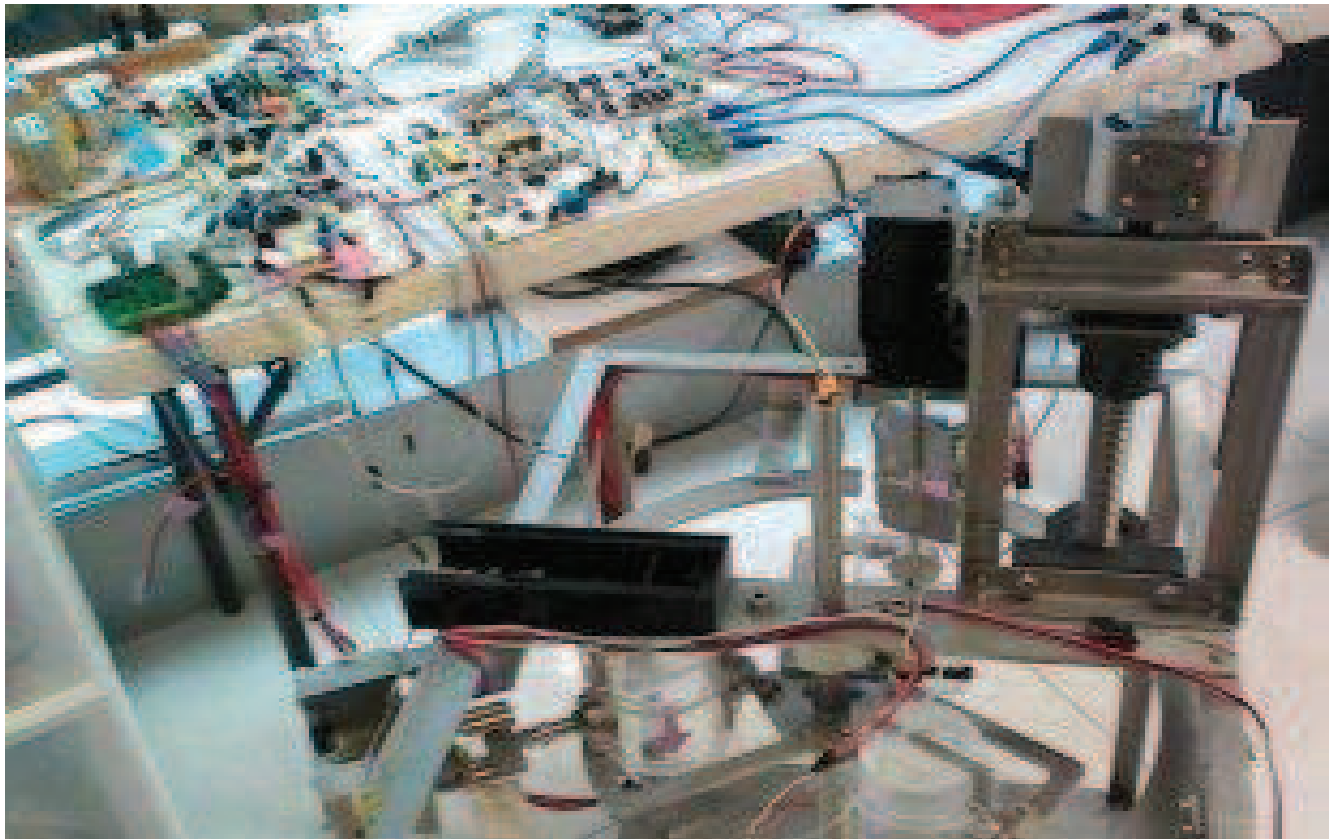


Figura 140. Unidad de control y sistema mecánico en la última Fase de trabajo

Prueba de comunicación simulada. Para las pruebas se inicia con la puesta en simulación, a manera de mostrar las conexiones y respuestas de los módulos esclavo, para andar el set de instrucciones se usa un monitor serial en el maestro, la muestra de las pruebas de interfaz quedan limitadas por las condiciones de la empresa.

En la Figura 141 se muestra la detección de los elementos en la estructura, retomando las operaciones de comunicación, el maestro envía solicitud de id de cada esclavo, si no responde se regresa valor nulo. En la parte superior se encuentra el monitor del maestro seguido de los esclavos; el uso de las funciones de monitor, como se ha mencionado, no son desarrolladas en este trabajo por ser de carácter particular, y su uso es para el programador.

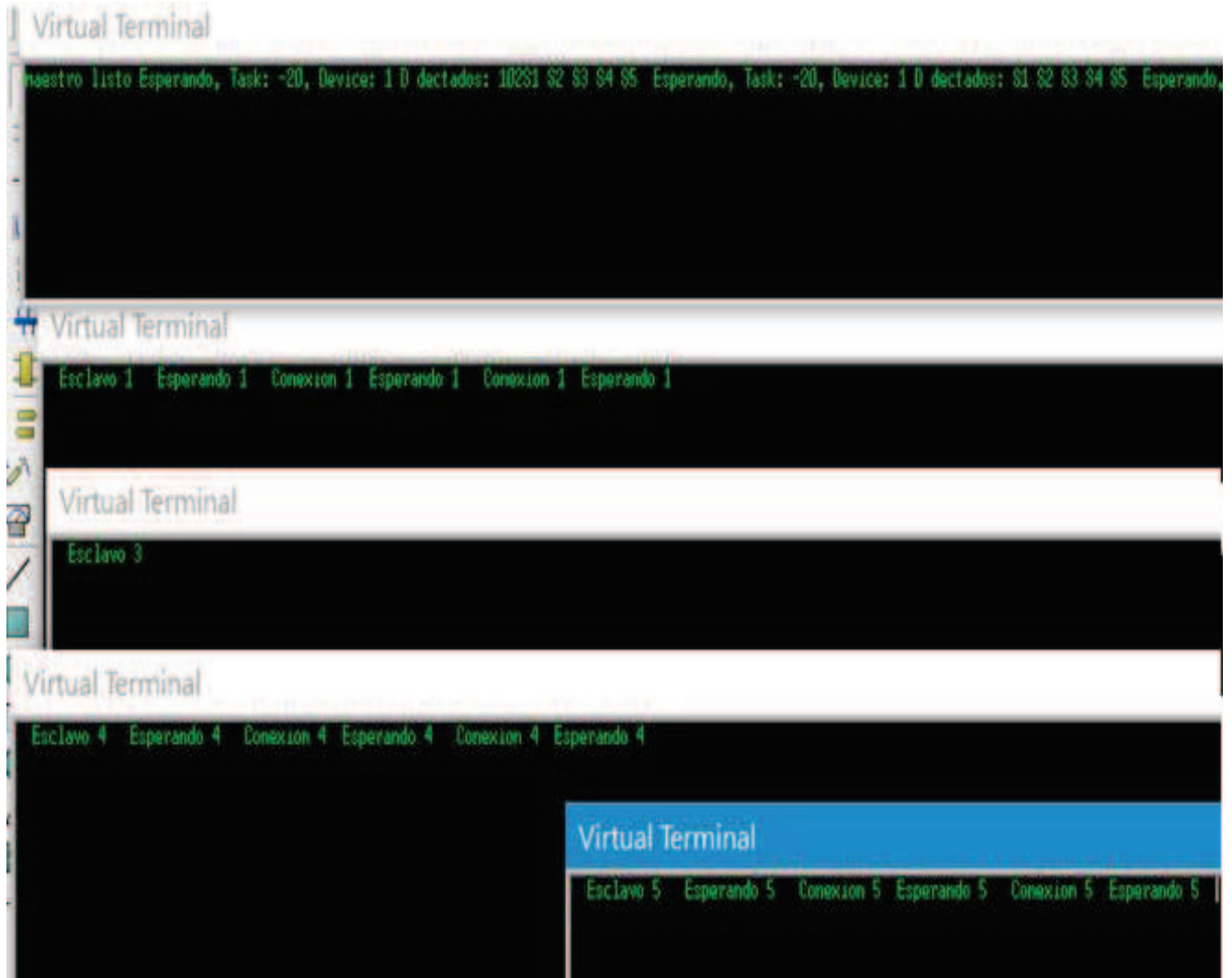


Figura 141. Estructura de comunicación, simulación de solicitud id de esclavos

Prueba de comunicación física. En la Figura 142 se muestra de lado izquierdo el sistema de control conectado con los monitores, convertidores de serial a USB, y las placas de los módulos; de lado derecho se muestra la información del monitor serial físico de la unidad de control. Se envía solicitud de envío de datos a los esclavos, cada esclavo responde únicamente a su solicitud y convierte los datos a su tipo, mientras que el maestro envía datos de forma general.

La comparación entre el sistema simulado y el sistema físico demuestra la adaptación de la estructura de comunicación en el microcontrolador. A pesar del problema con el dispositivo PIC16F887 en la lectura por I2C, al forzar los reinicios de comunicación se mantuvo estable en todo momento, permitiendo lecturas y escrituras sin pérdida de datos, aunque el tiempo que le toma al maestro n enviar datos a los esclavos, aun cuando no estén presentes, es de 9 [ms] por la simulación, se considera aceptable para las operaciones del proceso pues no se requiere en todo momento la solicitud de datos.

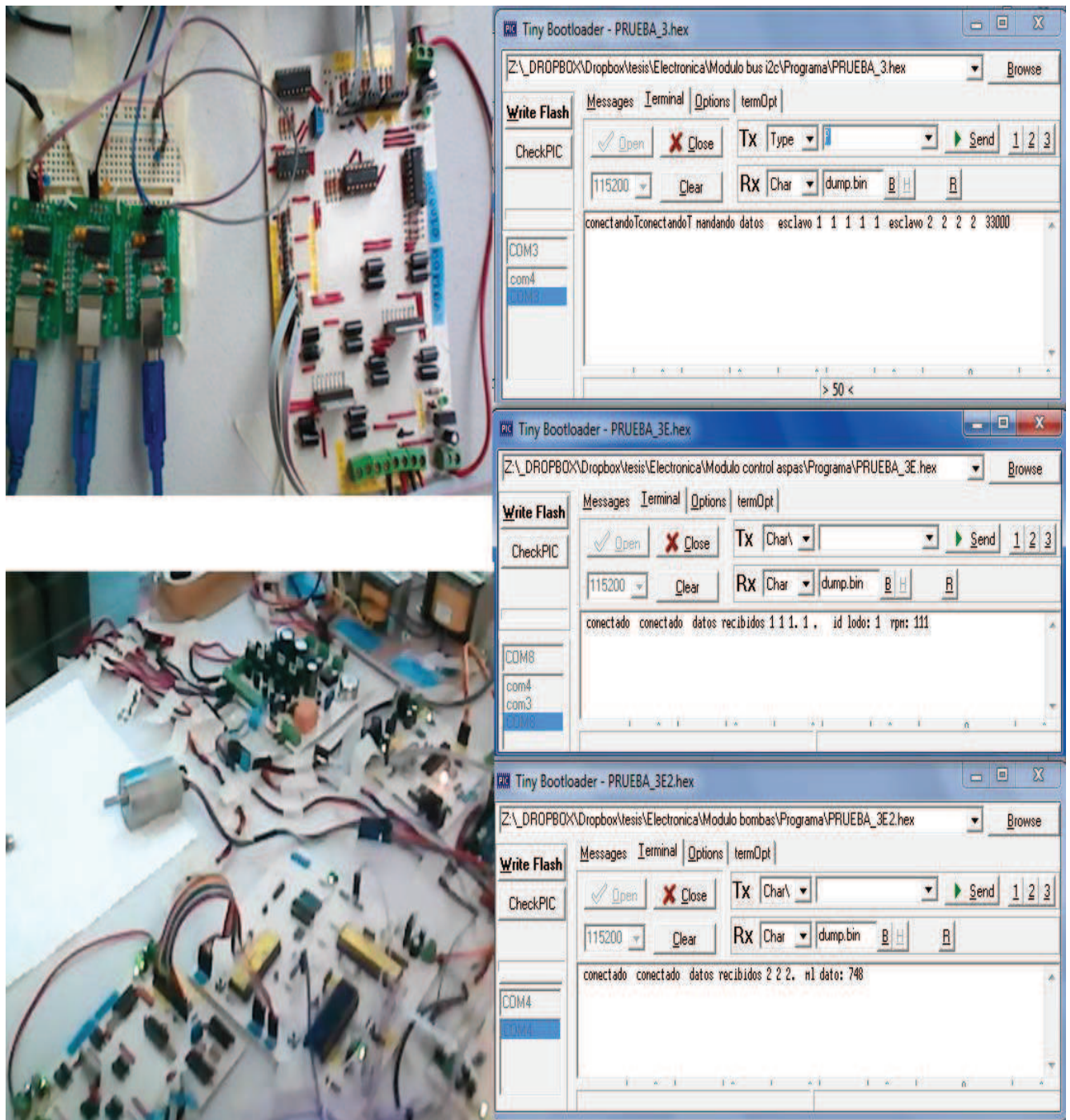


Figura 142. Estructura de comunicación, prueba de solicitud envío de datos

Prueba de ejecución de tareas y detección de errores. En la Figura 143 se muestra una solicitud de tareas para dos esclavos por parte del maestro, de lado izquierdo se muestra los puertos COM de cada monitor: el 3 para el maestro, el 4 para esclavo de polímero y el 8 para lodo. Se ejecuta envío de datos y una solicitud para el módulo de lodos; por otra parte, de lado derecho se muestra al sistema de control conectado mientras recibe las instrucciones por los monitores.

Las pruebas para solicitud de tareas no tuvieron problemas; por las condiciones de operación y por la estructura de la planeación se asegura su funcionamiento al tener todos los elementos conectados y probados, además las pruebas se realizaron en un ambiente controlado.

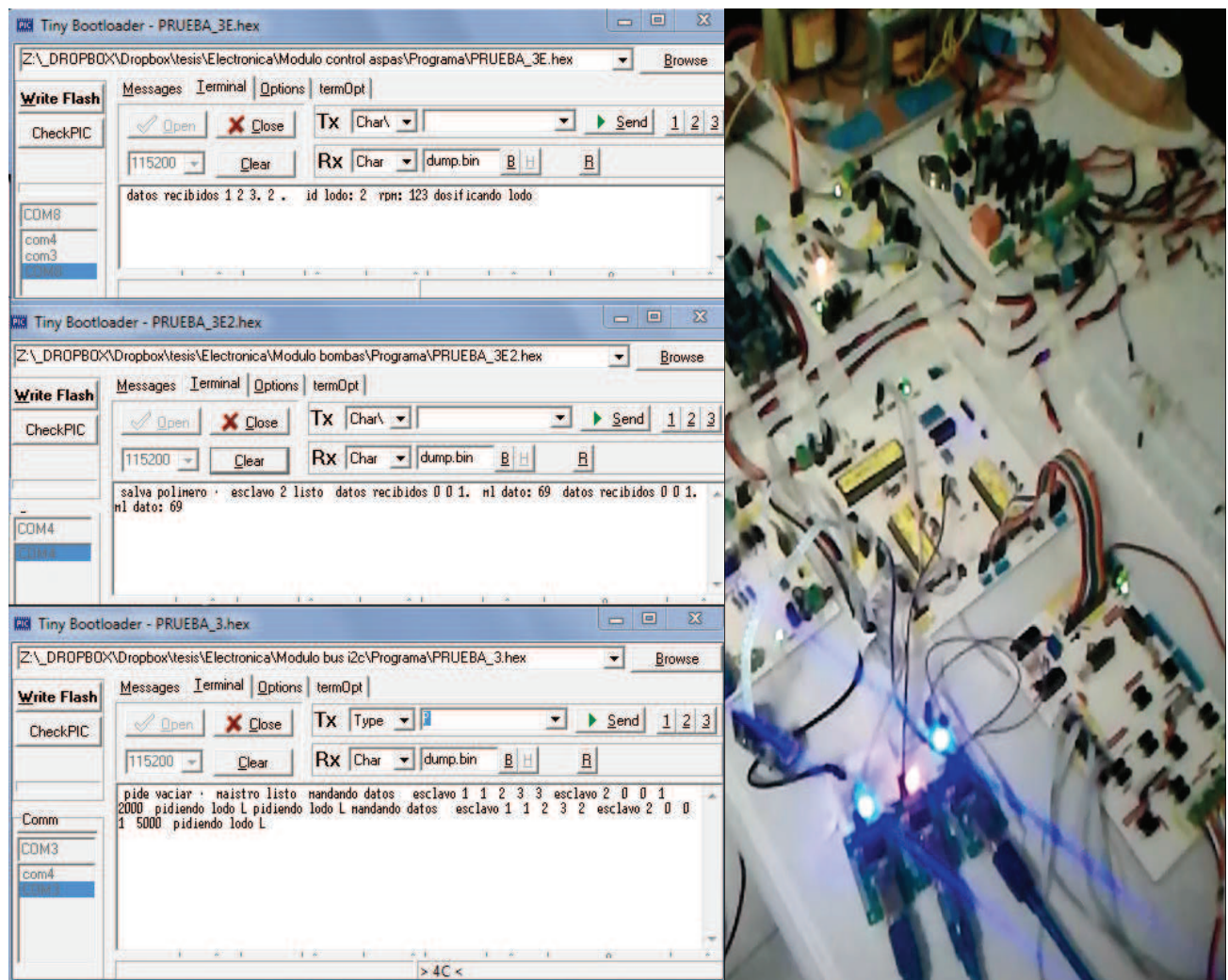


Figura 143. Estructura de comunicación, prueba solicitud tareas

Detección de errores en tarea. Para la detección de errores en tarea, se retoma el cap. 4.2.2 en la sección de detección de errores, le microcontrolador cuenta con dos maneras para la detección por parte del maestro y por parte del esclavo, a partir de la configuración de Timer para la duración de operación y por la misma detección del esclavo para la línea de error. Para las pruebas se establece una secuencia de tarea entre el módulo maestro y el módulo de efluente. El sensor del esclavo es desconectado y se espera la respuesta en tiempo de ejecución. La evolución para la detección es:

- Se envía solicitud de tarea.
- Se ejecuta la tarea y a su vez la configuración por Timer.
- El maestro espera la finalización de la tarea.
- El esclavo detecta fallo en sensor por tiempo de respuesta y detiene proceso, levantando línea de error.
- El maestro detecta el error, determina la línea y solicita el paro a todos los módulos. Finalmente verifica la conexión de los dispositivos.

En la Figura 144 se muestra una ejecución normal de una tarea, el maestro es el de la izquierda y el esclavo es el superior derecho. De lado izquierdo se muestra el inicio de solicitud de tarea, el maestro espera a que termine. De lado derecho se muestra el fin sin error, el maestro levanta un indicador de fin de tarea, mientras que el esclavo permanece apagado por fin normal.

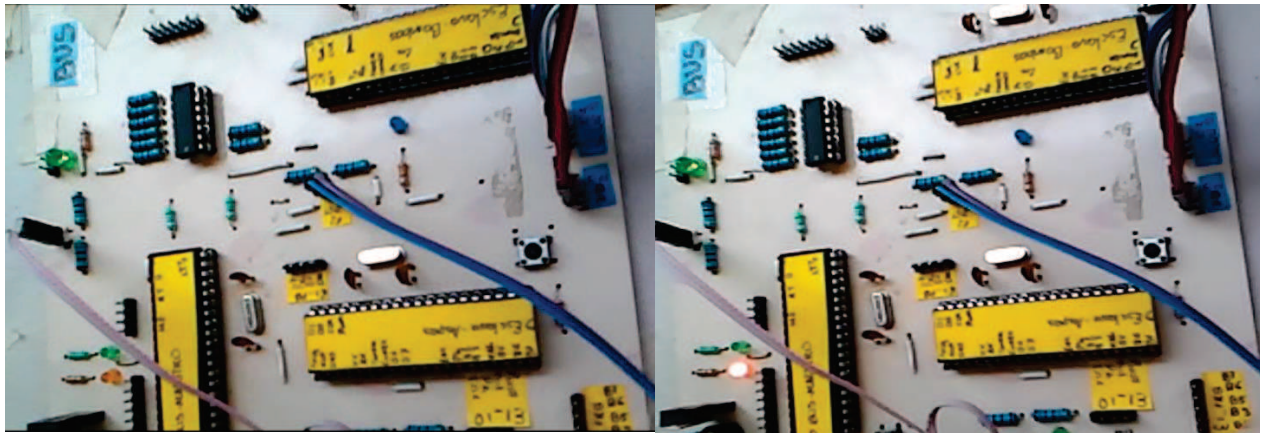


Figura 144. Estructura de comunicación, prueba de detección de errores, fin normal

La evolución de la detección de errores se muestra a continuación en forma de serie, se ponen en marcha tres de cinco esclavos, y se toma impresión de pantalla de los monitores. En la Figura 145, el maestro inicia la rutina de detección de dispositivos, de lado izquierdo el monitor del maestro, luego de detectar los dispositivos disponibles, dejando en 0 los faltantes. De lado derecho el esclavo de efluente responde solicitud de conexión, indicando que es el dispositivo 1, el cual será puesto a error.

<pre> Adress id - 0xFF maestro esperando task: -20, Device: 1 0 detectados: S1 S2 0 S4 0 esperando </pre>	<pre> Adress id - 0xA0 efluente ----- esperando Conexion 1 esperando </pre>
--	--

Figura 145. Prueba de error, detección de dispositivos

En la Figura 146, se muestra la solicitud de tarea, primero se selecciona la tarea 4 para el dispositivo 1 y finalmente se manad la solicitud; de lado derecho el esclavo 1 recibe soliciirtud de tarea 4, al ser un modelo generico se indica el “NOMBRE TAREA”. En este momento el escalvo ejecuta la funcion correspondiente y a su vez espera por timer la ejecucion, retomando el cap 4.2.2, la deteccion esta definida por “quién” temrina primero, entre la taarea y el tiempo estimado para la operación.

<pre> Adress id - 0xFF maestro esperando task: -20, Device: 1 0 detectados: S1 S2 0 S4 0 esperando task: 4 device 1 105enviando tarea 4 a 1 </pre>	<pre> Adress id - 0xA0 efluente ----- esperando Conexion 1 esperando tarea 4 NOMBRE TAREA </pre>
--	---

Figura 146. Prueba de error, solicitud de tarea

En la Figura 147 se muestra la detección de error. De lado izquierdo el esclavo detecta por Timer que hubo error, por lo que manda a detener sus procesos por reset de estados y pone en alto su línea de error. En el centro, el maestro detecta la línea de error, por planeación detecta la tarea de ejecución y por sensado lee las líneas de error; posteriormente manda la solicitud de paro de forma general. De lado derecho se muestra la solicitud de paro general en el esclavo.

Adress id - 0xA0 efluente	Adress id - 0xFF maestro	Adress id - 0xA0 efluente
esperando	esperando	esperando
Conexion 1	task: -20, Device: 1 0 detectados:S1 S2 0 S4 0	Conexion 1
esperando	esperando	esperando
tarea 4 NOMBRE_TAREA	task: 4	tarea 4 NOMBRE_TAREA
NOMBRE TAREA	device 1	NOMBRE TAREA
Existio Error, reset	105enviando tarea 4 a 1	Existio Error, reset
esperando	Error,	esperando
	codigo:1 t:1 id10000000	Pide error
	Deteniendo dispositivos	esperando
		Pide paro
		esperando

Figura 147. Prueba de error, detección de error y solicitud de paro

En la Figura 148 se muestra el proceso de verificación de la conexión de dispositivos para mostrar si todos siguen funcionando. Esta verificación permite al programador detectar los errores por evolución, en cuanto a la conexión entre el maestro y la interfaz, las operaciones son las mismas pero los comandos son enviados para generar notificaciones al usuario,

maestro	Adress id - 0xA0 efluente
.....	
esperando	esperando
task: -20, Device: 1 0 detectados:S1 S2 0 S4 0	Conexion 1
esperando	esperando
task: 4	tarea 4 NOMBRE_TAREA
device 1	NOMBRE TAREA
105enviando tarea 4 a 1	Existio Error, reset
Error,	esperando
codigo:1 t:1 id10000000	Pide error
Deteniendo dispositivos	esperando
.....	Pide paro
ACTIVE BOTON PARO EMERGENCIA	esperando
esperando	Conexion 1
task: 4, Device: 1 0 detectados:S1 S2 0 S4 0	esperando
esperando	

Figura 148. Prueba de error, verificación de dispositivos

Finalmente en la Figura 149 se muestra el indicador de fin de tarea del módulo maestro y en el módulo esclavo se muestra la indicación de error, comparece con la Figura 144.

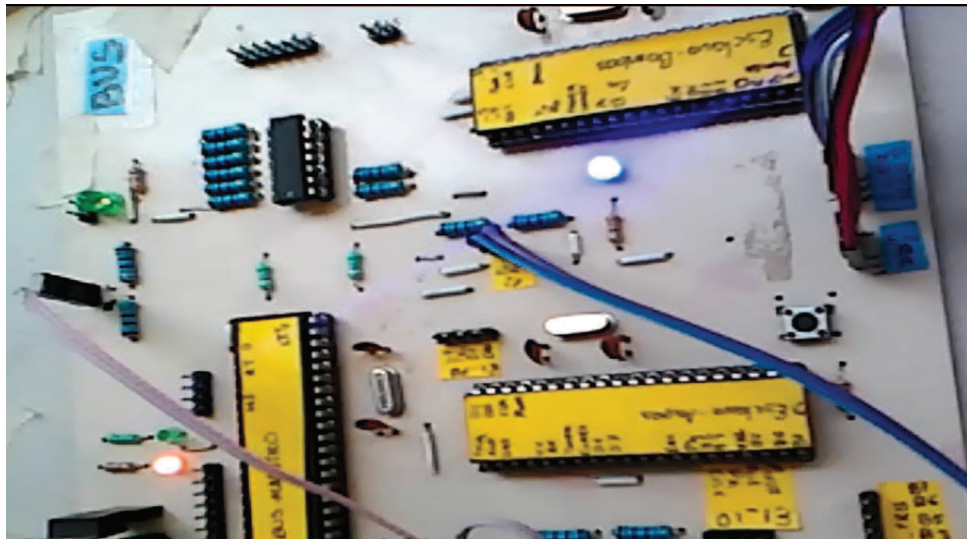


Figura 149, Estructura de comunicación, detección de errores, final por error

5.3 Prueba de secuencia

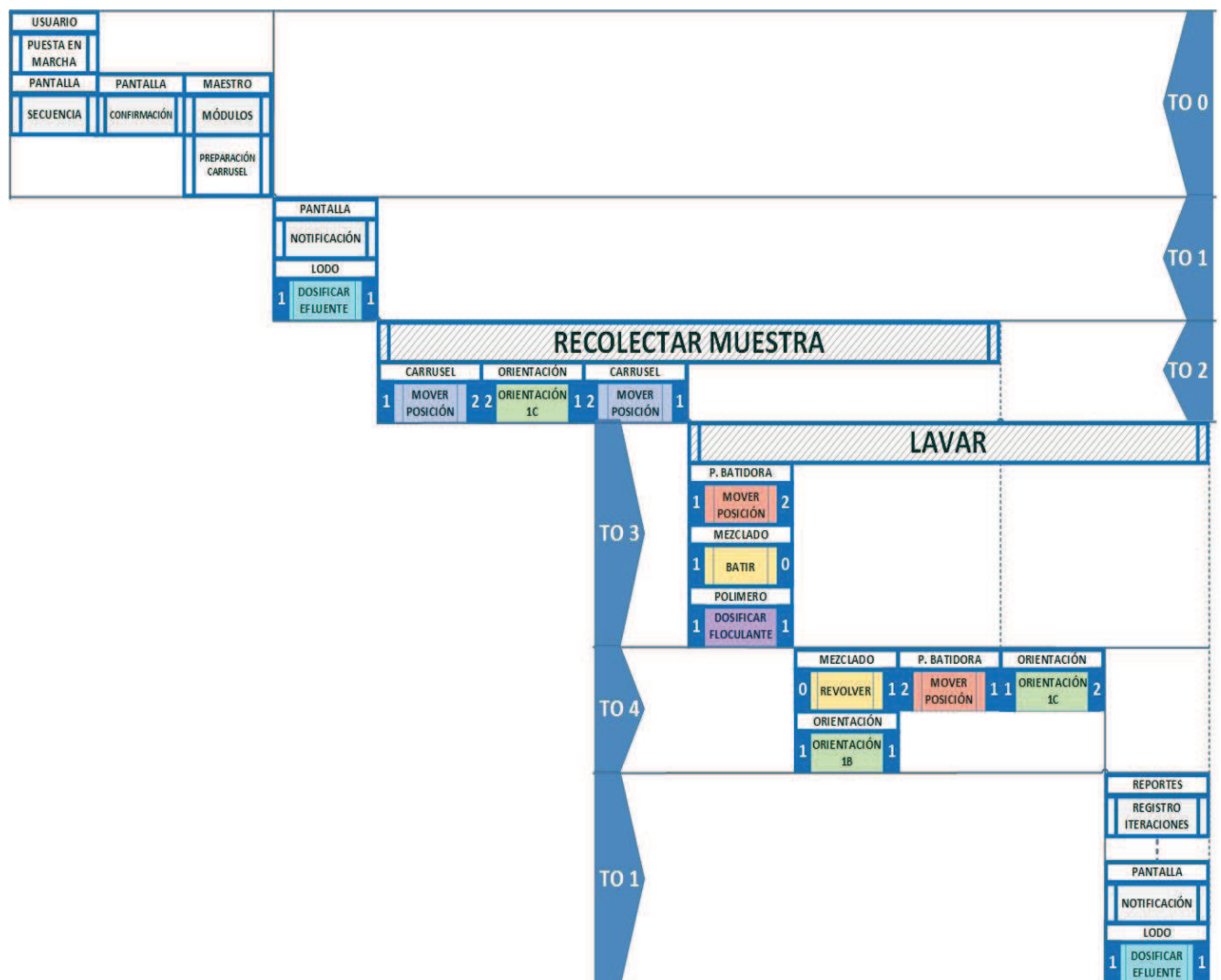


Figura 50. Tiempos de operación: Puesta en Marcha, 3

La prueba de las funciones de operación de secuencia, retomando la Figura 48 del cap. 4.1.4 sobre los tiempos de operación, representa el objetivo principal del proyecto sobre el proceso a automatizar, además integra la mayoría de las funciones de configuración, planeación y ejecución de la unidad de control y permite relacionar la evolución de las instrucciones de comunicación, Tabla 54-56. Todas las demás funciones de operación son características de carácter particular de la interfaz para configuración y monitoreo de los módulos, y describen de forma puntual el modelo de la estructura de comunicación; sus pruebas no son determinantes para la validación, además se siguen las condiciones de privacidad de empresa en cuanto a muestra de todas las funciones de la unidad de control.

De la Figura 48, para la prueba de las operaciones de secuencia, se inicia con el tiempo de operación 0, se comienza con la configuración de parámetros desde la interfaz, posteriormente se solicita la ejecución de secuencia y se muestra la evolución física del proceso; de las pruebas anteriores puede apreciarse su aplicación directa en el sistema físico.

Configuración de parámetros. Para la configuración se usa el módulo de interfaz, sus operaciones son de preparación del paquete de datos y el envío del set de instrucciones para las operaciones solicitadas por usuario; del Apéndice C se desarrollan las funciones gráficas para el despliegue de información de forma genérica, para avisos, confirmaciones, mensajes y menús.

En la Figura 150 se muestra el menú principal, con las 5 opciones de operaciones de configuración, izquierda; se selecciona “datos usuario”, el cual muestra antes de realizar cambios los datos actuales al usuario, derecha.

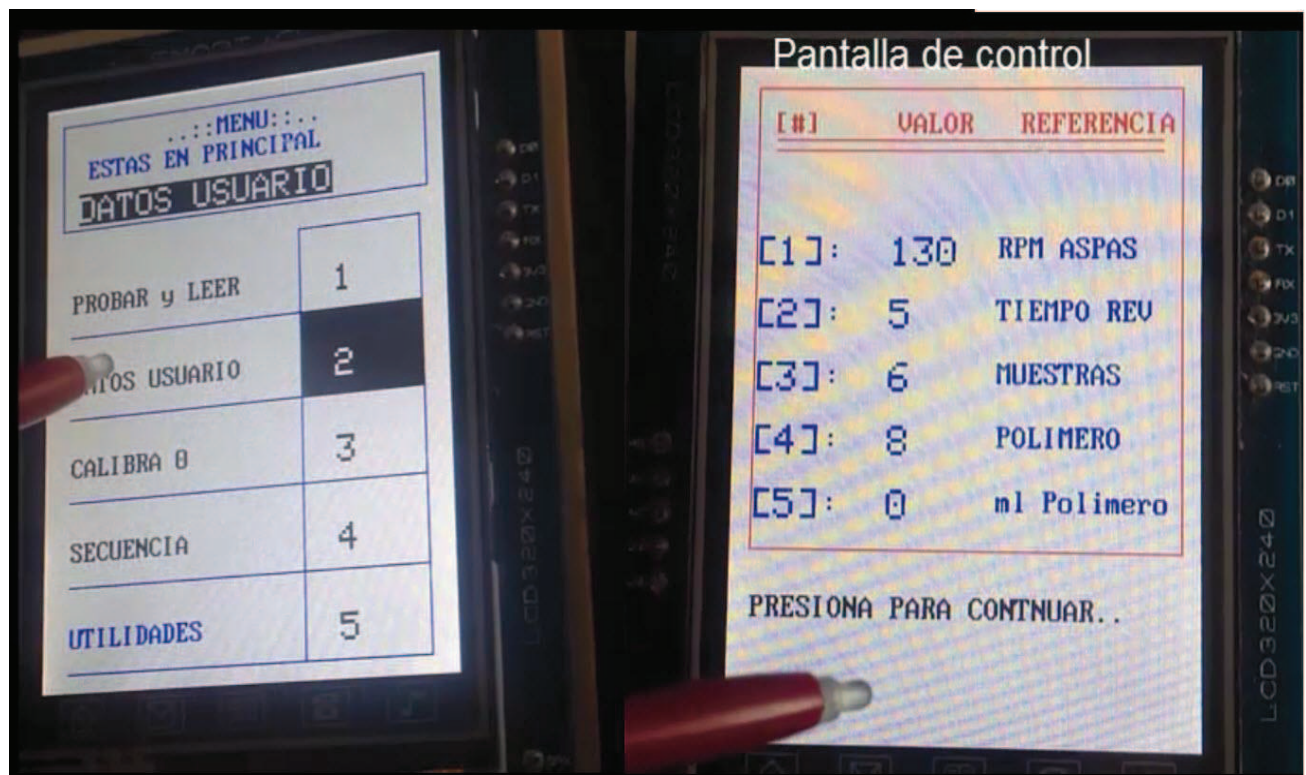


Figura 150. Prueba de interfaz, configuración de parámetros

En la Figura 151 se muestra la entrada al menudo de “datos usuario”; a la izquierda se presenta el mensaje de tipo de acción para indicar al usuario un cambio solicitado por él; de lado derecho se muestra el despliegue del menú de “datos usuario” con opciones de cambio de todos los parámetros.

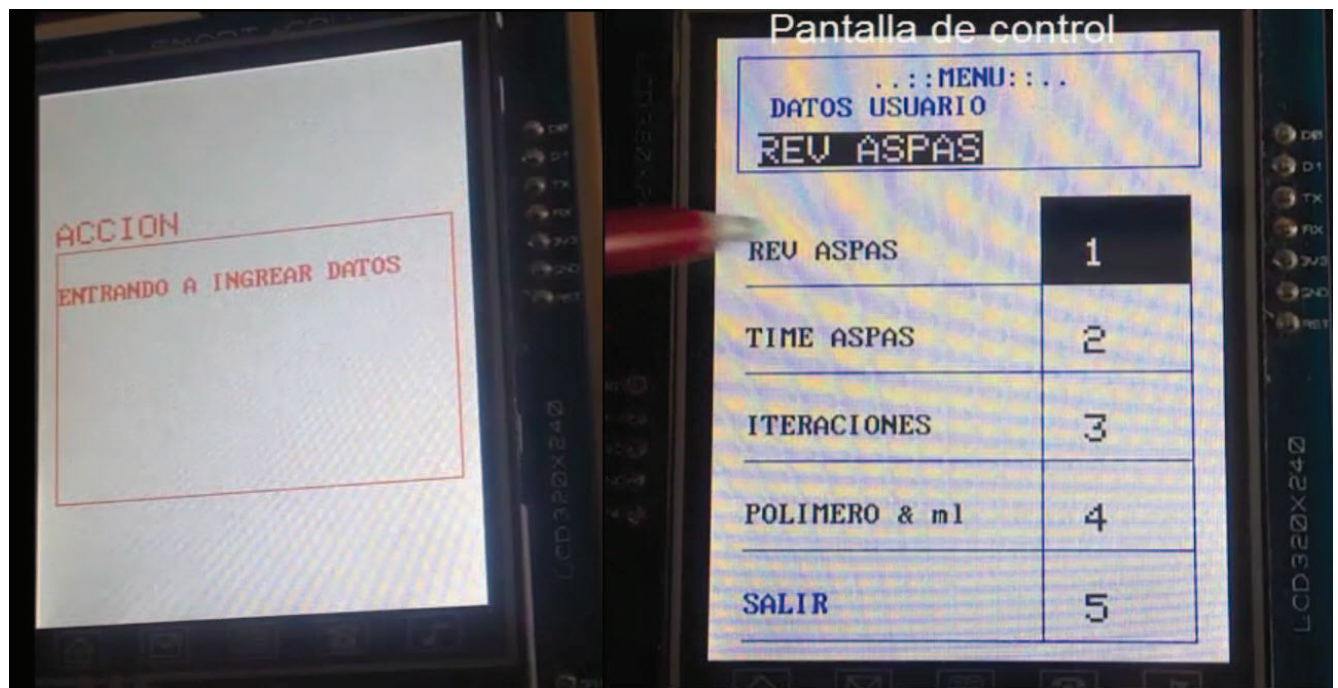


Figura 151. Prueba de interfaz, menu de datos

En la Figura 152 se muestra el ingreso del parámetro de rpm para la batidora; de lado izquierdo se da un mensaje a usuario con información particular del dato seleccionado, de lado derecho se despliega el formato de entrada de datos en formato numérico con información de la referencia máxima y las unidades del parámetro, confirmación y opción de borrado.

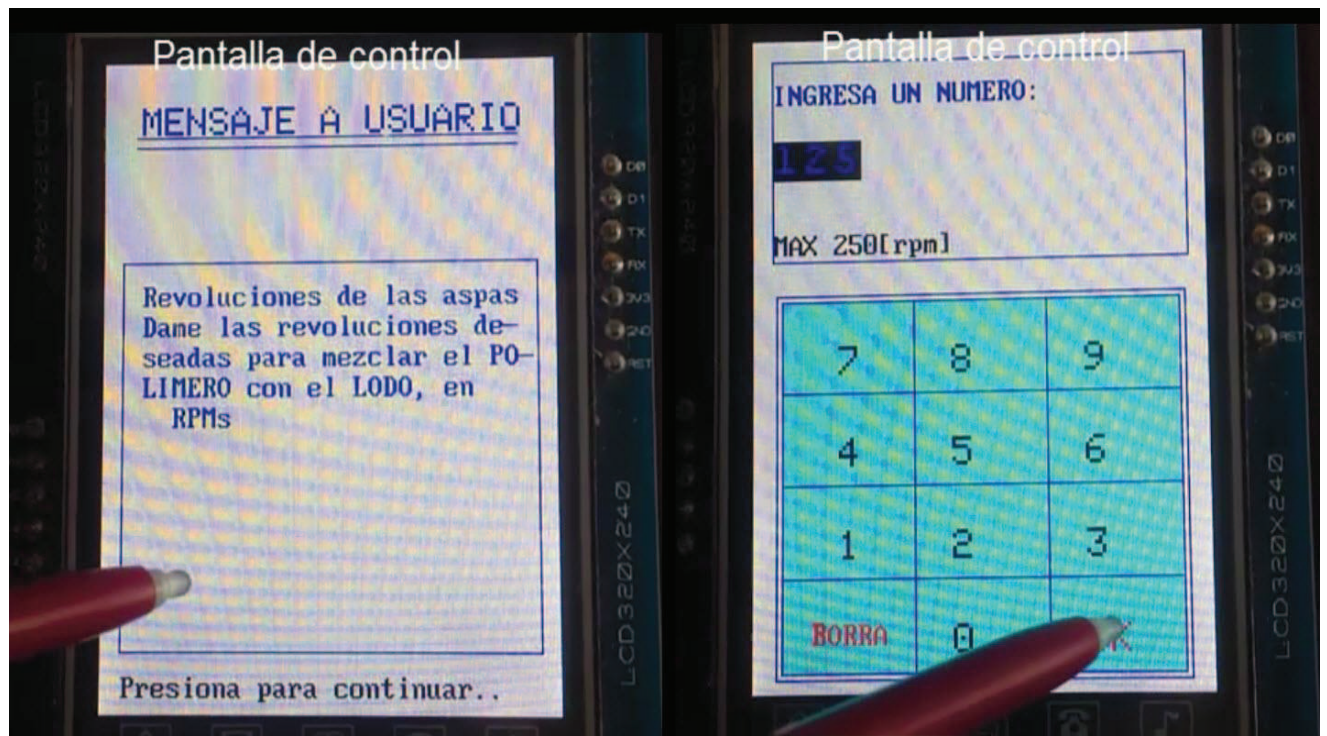


Figura 152. Pruebna de interfaz, ingreso de datos

En la Figura 153 se muestra la confirmación del dato de revoluciones, a la izquierda se presenta el dato seccionado en formato de mensaje para corroborar su valor numérico, a la derecha se solicita confirmación para asegurar cada parámetro.

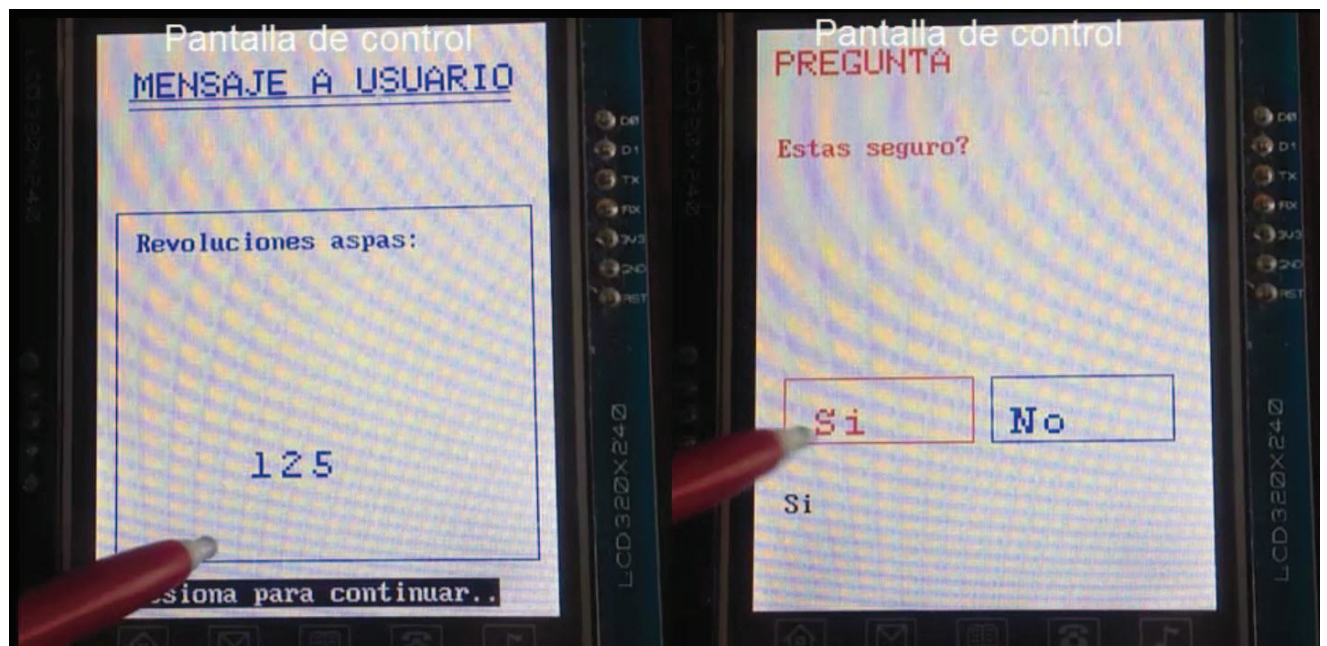


Figura 153. Prueba de interfaz, confirmación de dato

Para el cualquier tipo de operación en la interfaz, se ejecutan despliegues de información y de confirmación al usuario para corroborar las acciones y para auxiliar el uso de la interfaz, a manera de hacerlo intuitivo y seguro considerando condiciones físicas que puedan presentarse en planta, como alguna obstrucción independiente o una solicitud no deseada.

Solicitud de operaciones, secuencia. La solicitud de secuencia en la interfaz permite un monitoreo de los estados y de la evolución de las operaciones de forma simple y gráfica mediante mensajes de acción; por cada tiempo de operación concluido o detección de falla se notifica a la interfaz, este monitoreo fue desarrollado de forma independiente a la estructura de comunicación, por las características de los dispositivos, el objetivo no fue desarrollar funciones particulares, su desarrollo está en el Apéndice C. En la Figura 154 se muestra de lado izquierdo la selección de secuencia; al centro y a la derecha los mensajes para los tiempos de operación.

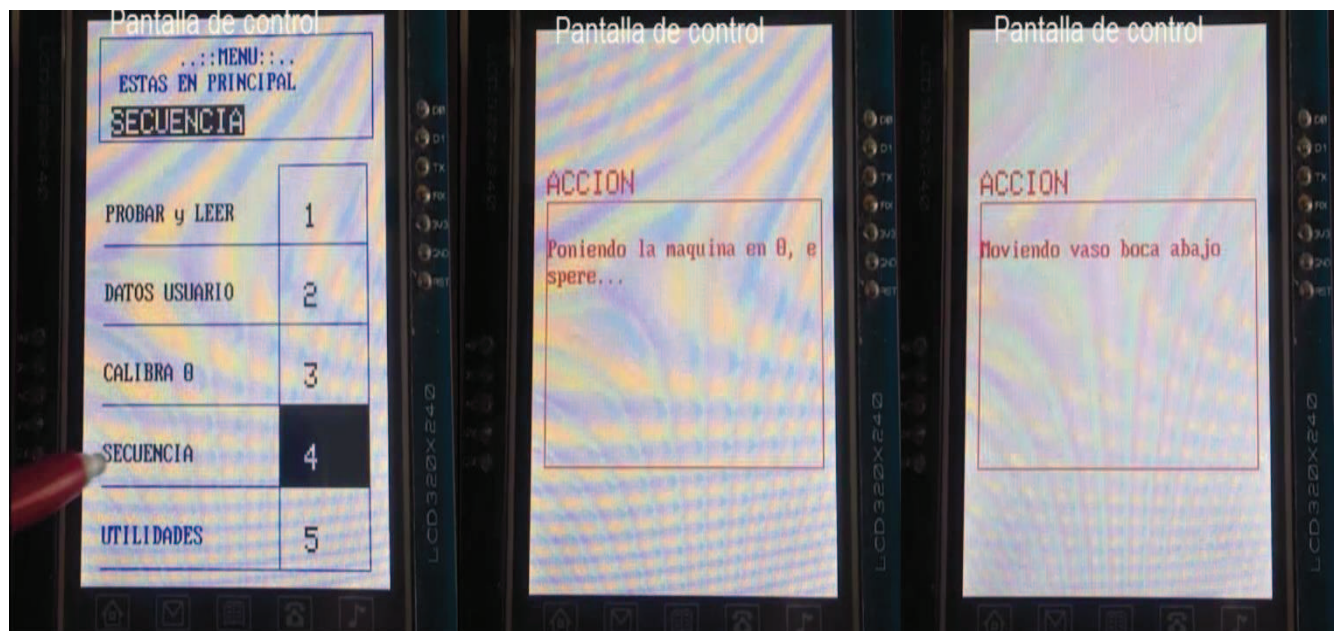


Figura 154. Prueba de interfaz, solicitud de secuencia.

Hasta ahora se ha representado el tiempo de operación 0 de las operaciones de secuencia, los siguientes tiempos son respetados por el movimiento de la estructura física. De las pruebas anteriores de comunicación, puede asociarse la evolución en la unidad de control, para una ejecución normal no se usan los monitores de los módulos y en la interfaz solo se notifica por mensajes de acción en cada tiempo de operación, como en la Figura 156. Los tiempos de operación se describen en las siguientes figuras, de la Tabla 22 se tienen las condiciones iniciales de la estructura, cap. 4.1.3 jerarquía de componentes. Previamente al solicitar la secuencia se envían los parámetros a los módulos, Tabla 56.

TO1. Figura 155, el maestro solicita la tarea al módulo efluente para la dosificación. Se habilita el actuador del módulo y se inicia el sensado, Figura 115.

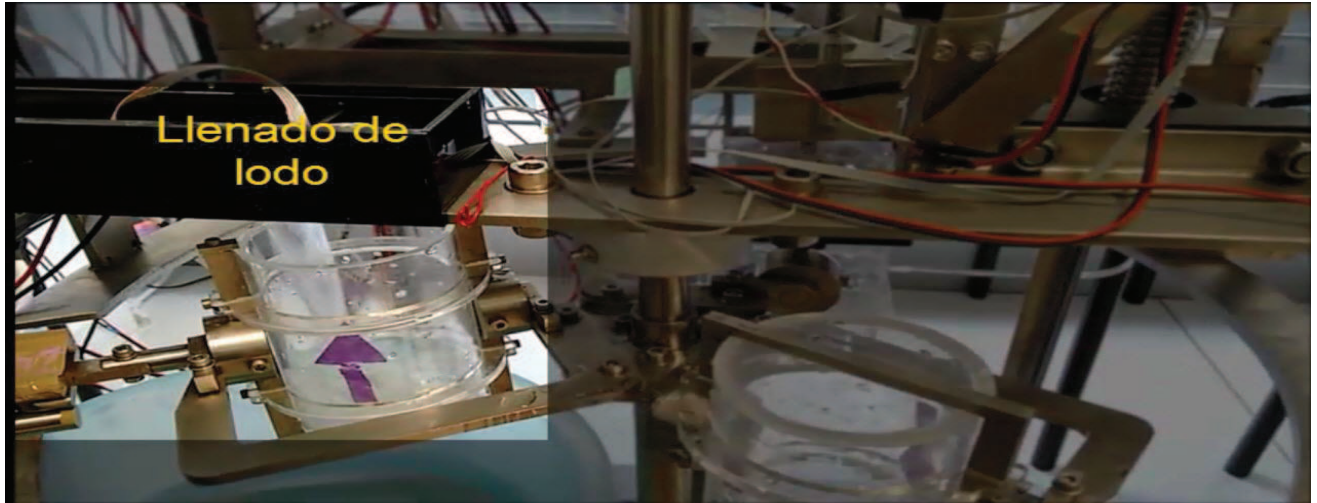


Figura 155. Prueba secuencia, TO1 dosificación lodo

TO2. Figura 156, terminando la dosificación, el maestro solicita tarea de cambio de posición de contenedor mediante tres tareas en el módulo estructura: primero en el sistema carrusel se solicita el movimiento a $\frac{1}{2}$, luego para el sistema de orientación se solicita cambio en 1C para que físicamente el contenedor pueda ser posicionado, mediante la solicitud en sistema carrusel de posición a 1, Figura 87, 88.

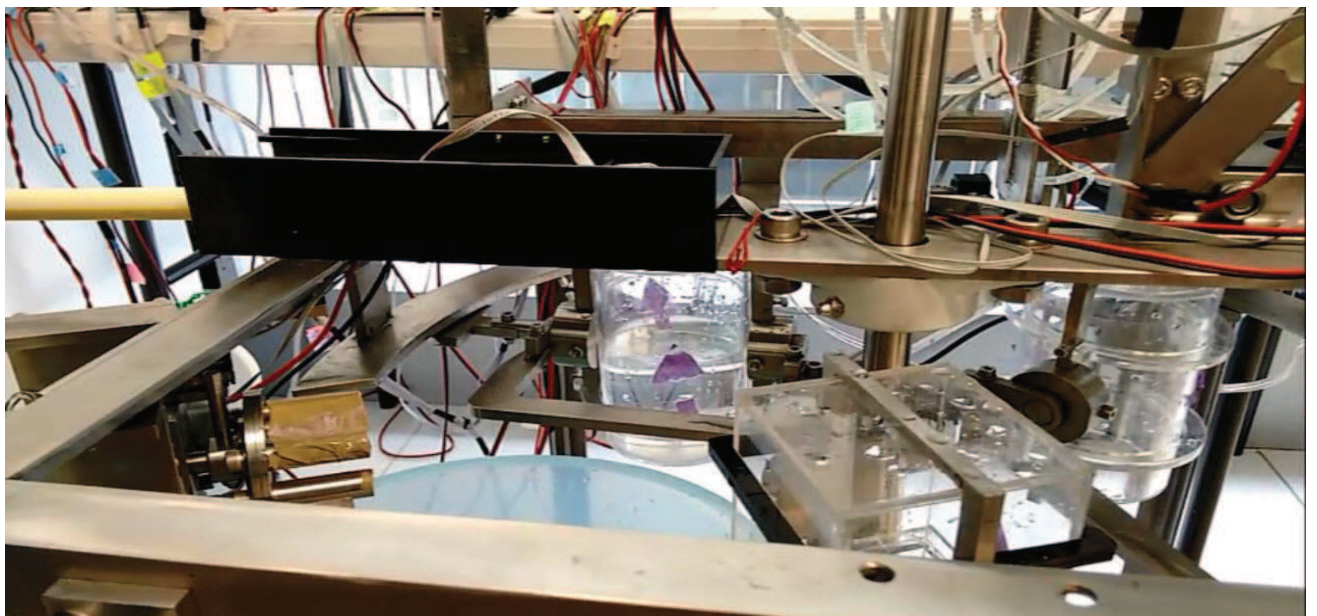


Figura 156. Prueba secuencia, TO2 movimiento contenedor

TO3. Figura 157, en la Figura de lado izquierdo, se muestra la ejecución de tres tareas simultáneas; el maestro solicita tarea a los módulos, Figuras 87, 104 y 117: en el de estructura se solicita cambio de posición de la batidora, en el de mezclado se solicita la ejecución de batir y en el de polímero la dosificación; por cada iteración se cambia el tipo de polímero con su cantidad correspondiente, en la Figura a la derecha se muestra la dosificación del polímero 1 mientras que en la izquierda el polímero 2, la operación de dosificación es de las más importantes del proceso.

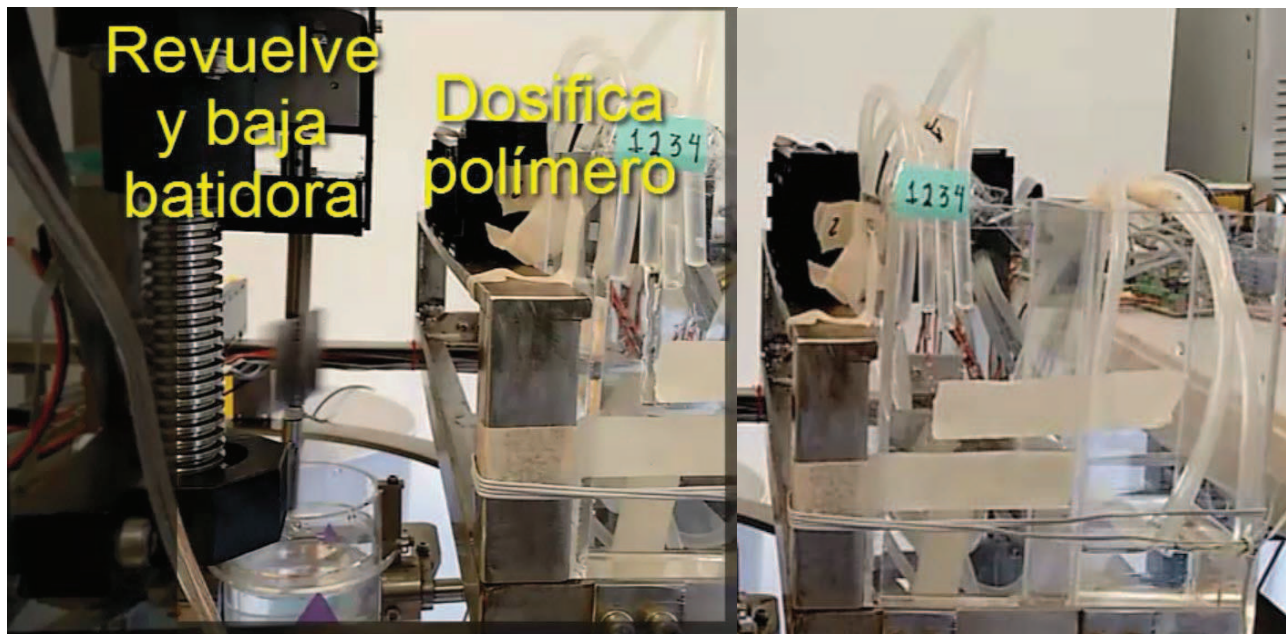


Figura 157. Prueba secuencia, TO3 dosificación y posicionamiento batidora

TO4. Figura 158, El maestro solicita al módulo de mezclado la tarea de mezclar el polímero con el efluente con una velocidad de usuario. Esta operación es la más importante de la secuencia, pues une las operaciones de la prueba de jarras de laboratorio con las pruebas en planta de tratamiento, cap. 4.1.1, las especificaciones del proyecto suponen la importancia de la exactitud de dosificación y del mezclado para igualar las condiciones y tener pruebas determinantes. Figura 104.



Figura 158. Prueba secuencia, TO4 mezclado

Figura 159, después del mezclado, el maestro solicita al módulo estructura el cambio de posición de la batidora, Figura 87. Tanto la bajada como la subida presentan unos tiempos largos de operación debido a las condiciones mecánicas; a manera de referencia, el equipo mecánico buscó la solución para evitar el uso de seguros mecánicos y reducir el número de actuadores para evitar el efecto de la gravedad por ser un movimiento vertical con lo que se definió el uso de husillo como solución, por el tiempo de manufactura se probó a finales de la etapa cuatro del proyecto donde se determinó el cambio de su actuador por el tiempo de operación. Dicho cambio no afectó ninguna condición ni estados de operación.

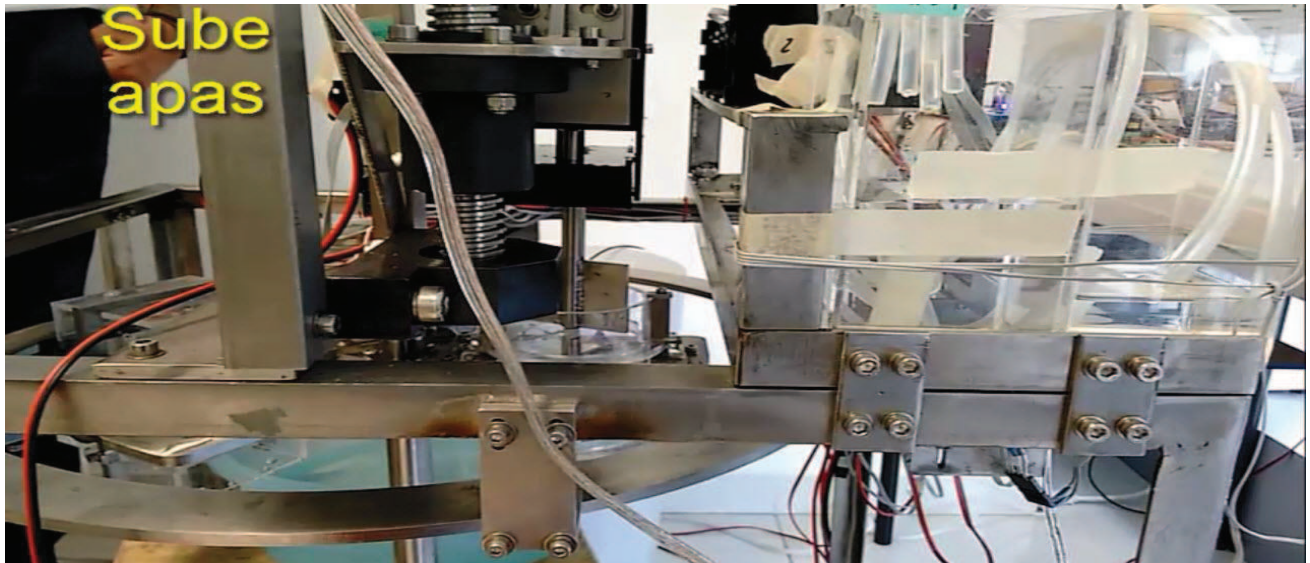


Figura 159. Prueba secuencia, TO4 posicionamiento batidora

Figura 160, el maestro solicita al módulo estructura el cambio de orientación en posición 1C para la recolección de la muestra, esta acción pertenece a la etapa 5 del proyecto, para su análisis. No se tiene definida alguna estructura mecánica que condicione la velocidad del giro.

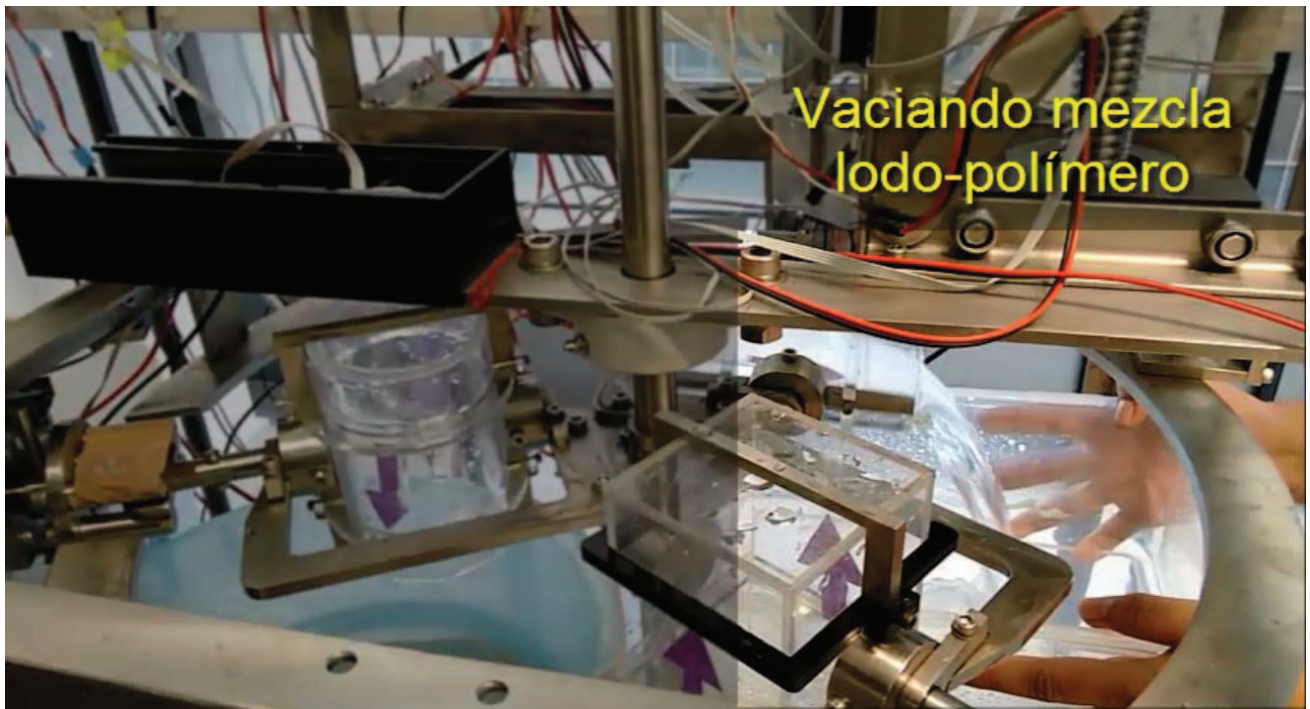


Figura 160. Prueba secuencia, TO4 cambio orientación para recolección muestra

Figura 161, Finalmente se realiza el cambio de orientación en posición 1B, comparado con la Figura 48, esta operación se posterga debido a que en la última presentación de avances se dio la recomendación de mantener secos los contenedores antes de nuevas pruebas, para no modificar las muestras, el cambio de esta operación permite mostrar la característica del sistema modular al no requerir una reprogramación total de la planeación, sino un cambio en tiempo de ejecución, el cual no afecta a los módulos esclavos.

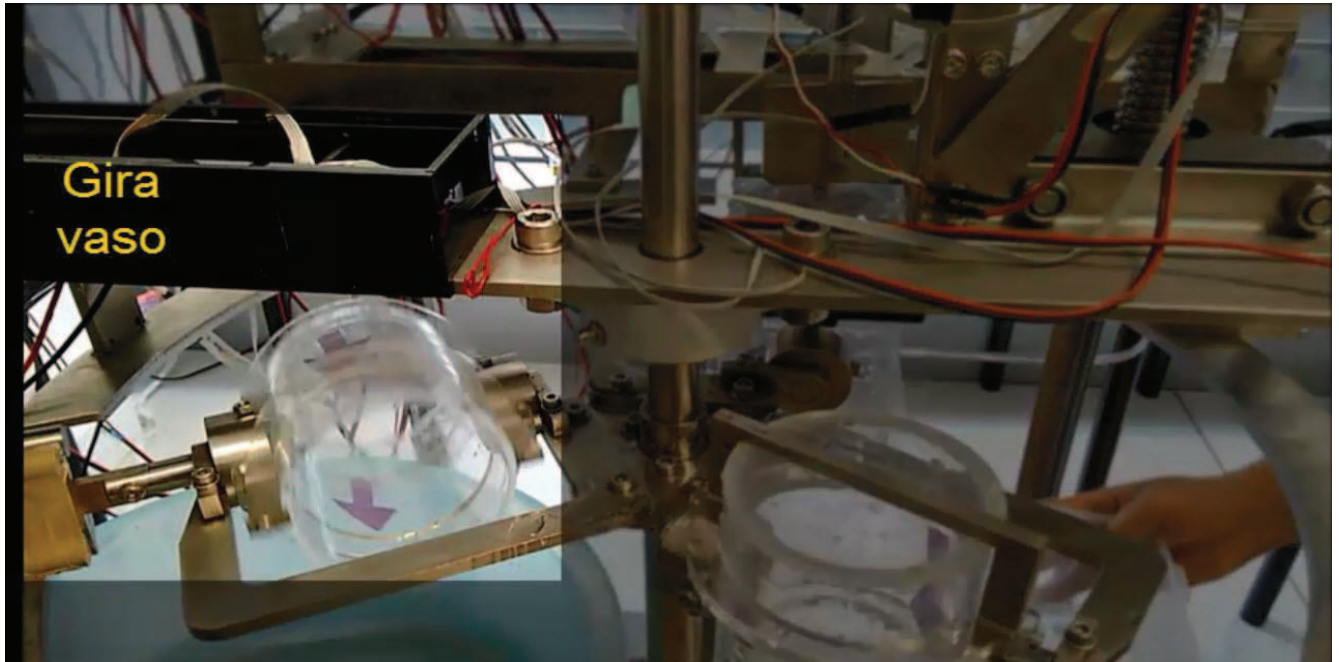


Figura 161. Prueba secuencia, TO4 orientación para dosificación efluente

5.4 Resultados y análisis de pruebas

Por cada prueba realizada se tiene un resumen del comportamiento de los estados físicos y los esperados. Se considera la importancia de tener fuentes dedicadas para cada tipo de etapa en el sistema electrónico para reducir y asilar las caídas de voltaje por consumo de corriente en caso de falla mecánica. Además, todas las pruebas fueron sobre la unidad de control, los sensores y actuadores fueron parte de las especificaciones y limitaciones iniciales, por lo que se probó a su vez su funcionamiento para definir sus cambios posteriores bajo el mismo concepto de funcionamiento. Es importante mencionar que el desarrollo del presente trabajo se centró en el establecimiento general de la unidad de control para adaptarse a las condiciones de privacidad del proyecto en cuanto a descripción de funciones específicas.

La aplicación de la solución se muestra con una aplicación particular en el proyecto independiente al diseño de la unidad de control al establecer la comunicación y la estructura del módulo de forma genérica, mientras que el análisis del proceso y sus componentes determina la generación de las funciones de operación y la de los bloques funcionales, como paso intermedio para la implementación en microcontroladores.

Etapa acondicionamiento sensores. Las señales fueron adecuadamente acondicionadas, se asegura una lectura de valor lógico ya que ninguna señal es variable. Por el tipo de aplicación y los tiempos de respuesta el acondicionamiento fue simple; durante las pruebas se presentaron caídas de voltaje al tener todos los sensores conectados, las cuales son provocadas por el amplificador

operacional para los encoders por el consumo del dispositivo, Tabla 69, de tal forma que la fuente dedicada para la etapa de acondicionamiento se establece con un voltaje de 6.5 [V], del cual se tiene entre 4.5 y 5 [V] de salida dependiendo del número de sensores activos. Por otra parte para los sensores mecánicos con este ajuste es de 5.5 a 6 [V], rango soportable para entradas del microcontrolador.

Tabla 69. Resultados pruebas de acondicionado sensores			
Sensor	Voltaje de salida sensor	Salida de acondicionado	Observaciones
Encóder por bloqueo	0-4 [V]	0 – 3.5 [V]	Los sensores de herradura aseguran la lectura por bloqueo, en la transición del bloqueo puede haber lecturas mientras la luz del emisor es totalmente bloqueada, el acondicionamiento asegura estados lógicos
Encóder por objeto	0 – 2 [V]	0 – 3.5 [V]	Los sensores por objeto presenta una señal variable dependiendo al distancia del encóder y del color de la superficie; el acondicionado asegura la señal lógica constante
Mecánico	0 – 4.5 [V]	0 – 4.5 [V]	No se presenta caídas de voltaje, el acondicionado permite la supresión de picos por el efecto del resorte mecánico para la activación

Por otra parte, las funciones de lectura de encóder para el control de velocidad y dosificación y del sensor ultrasónico para el volumen fueron constantes, la velocidad de procesamiento del microcontrolador pudo ejecutar las funciones de acondicionamiento por software, tanto del control PID como el proporcional basado en raíz del error y la determinación del volumen de efluente

En cuanto al error, los dispositivos que se relacionan con los parámetros de configuración representan las funciones con prioridad del proyecto, se estimó un error menor al 5 % considerando las pruebas ejecutadas por el operador, los errores relativos fueron variables por la referencia y el comportamiento físico de la variable medida, en la Tabla 70 se muestran los errores de medición en los parámetros de mayor importancia.

Tabla 70. Resultados pruebas de error en dosificación y mezclado				
Sensor	Función	Referencias	Valor medido	Error relativo %
Efecto hall	Mezclado	130 rpm	127.5 – 132.7	-1.92 – 2.08 , 4.62%
		131 rpm	130 – 132	-0.76 – 0.76 , 1.53%
		80 rpm	79.5 - 81	-0.63 – 1.25 , 1.88 %
Ultrasónico	Dosificar efluente	37 cm	36	2.7 %
		24 cm	24	0 %
Encóder	Dosificar floculante	3 ml	3.1	3.33 %
		5 ml	4.9	2 %
		10 ml	10.2	2 %

En cuanto al mezclado, el error a altas velocidades es menor de 5 % mientras que a velocidades bajas se menor a 2%, esto se debe a las ganancias obtenidas por ZN y por el efecto mecánico de las aspas de la batidora. Para el ultrasónico del volumen del efluente, el error varía dependiendo de la velocidad de llenado sin superar el 5 % por el tiempo, por otra parte, la bomba utilizada no es definitivo por las pruebas a realizar en planta, ele quipo mecánico selecciono la bomba de forma

temporal, por lo que este módulo puede cambiar en sensado. Finalmente, para el encóder de dosificación de floculante, el error fue menor al 5%, disminuye mientras más volumen sea solicitado debido a que el encóder propuesto solo tiene un punto de referencia, a pesar de esto puede fijarse el error en ± 2 gotas equivalente a 0.1 [ml] considerando la inercia del fluido a la salida de las vías.

Estas variaciones fueron aceptadas por el equipo de trabajo como primera iteración antes de generar el prototipo final, el modelo se valida para el sensado comparando los procedimientos del operador y a falta de control en las variables del proceso.

Etapla acondicionamiento potencia. La etapa de actuadores genera una caída de voltaje de 12 [V] hasta 9 [V] en función del número de actuadores usados; como el control de estos es por retroalimentación de posición o de alguna variable el tiempo no afecta para llegar al valor esperado, por otra parte, este tiempo para el microcontrolador es ajustado a un máximo en cuanto a la detección de error por sensado y por tiempo de operación. El consumo de corriente no fue medido ni considerado para el diseño, como se mencionó en la Fase 1, queda fuera de los alcances de este trabajo, se usó dispositivos que superaran la corriente nominal de trabajo de cada actuador, pero no se estudió la forma óptima ni para el consumo de sensores, además el desarrollo de la fuente queda bajo el cargo de otros miembros del equipo de electrónica.

Los movimientos de los actuadores fueron los esperados y los cambios y ajustes fueron mecánicos. En cuanto al actuador de mezclado, se espera el cambio del motor por sus irregularidades en especificaciones, este cambio no afecta en el tipo de sensado ni en las operaciones del microcontrolador, puede establecerse un nuevo análisis por ZN o caracterizarlo a partir de especificaciones de fabricante y banco de pruebas. Las bombas de dosificación cumplen con el requerimiento de exactitud para el control de su variable, no se espera su cambio. Los actuadores de la estructura mecánica funcionan adecuadamente para los movimientos, para el caso del posicionamiento vertical de la batidora, se observó un tiempo excesivo de 20 [s], por lo que se espera cambiar el tipo de motor, este no afecta en ninguna etapa pues tiene retroalimentación por sensores. Por otra parte, los tiempos de ejecución de operación son aceptables y quedan en función de la estructura mecánica, las rampas de velocidad al inicio permiten reducir los consumos de corriente y al final permiten reducir la inercia mecánica.

Asegurar las pruebas de sensado y de potencia permite reducir e identificar las causas de errores durante ejecución de operaciones. Esta primera iteración permite calificar el desempeño de los actuadores y sensores seleccionados con el sistema mecánico; la validación se centra en el sistema electrónico para estas etapas de acondicionamiento. Para la continuación del proyecto, se esperan cambios en sensores y actuadores bajo el mismo principio, a manera de mejorar los tiempos de operaciones.

Etapla de control. Las pruebas de la etapa de control de la unidad de control se centraron en probar la estructura de comunicación y en la ejecución de las operaciones de planeación. Debe recordarse que la estructura modular fue planteada a partir de una estructura de operaciones para la unidad de control la cual define operaciones de configuración, planeación y ejecución, de esta forma los módulos se vuelven independientes y únicamente se centran en la ejecución de sus tareas, siendo que en conjunto por las operaciones de planeación realizan tareas con un fin común, Figura 62.

Al separar las operaciones de configuración con las de planeación se permite la independencia del dispositivo de implementación con el set de instrucciones. Al definir la evolución de los procesos de planeación a partir de las instrucciones puede desarrollarse la interfaz en cualquier dispositivo,

siguiendo dicha evolución, además solo representan las operaciones básicas por lo que las capacidades del dispositivo final pueden mejorar la experiencia de usuario y el monitoreo del sistema por definir lianas de comunicación independientes en el módulo maestro.

El sentido del análisis de componentes y las funciones de operación del proceso fue establecer las condiciones y estados del proceso a automatizar en relación al sistema físico, de tal forma que en la planeación del módulo maestro se permite cambios en la ejecución de tareas, flexibilidad de operaciones, siempre que cumpla con estas condiciones y estados. En este sentido, la detección de errores, ejecución de multitareas y la facilidad de mantenimiento por parte del programador fue posible por la centralización de las operaciones de planeación y la independencia de los módulos esclavos.

Las pruebas de la comunicación muestran la implantación de la estructura del módulo, en cuanto a los paradigmas deliberativo y orientado a eventos, a su vez en el formato de envío de datos; la adaptación a microcontroladores y el uso del protocolo I2C para la red en paralelo junto con las líneas auxiliares de tarea y error representan la aplicación de una solución propuesta para asegurar que el sistema fuese robusto durante ejecución de operaciones.

El envío de solicitudes queda probado para la ejecución de tareas y envíos de datos, mientras que para los errores se presentó la evolución entre el maestro y esclavo. Asegurar la comunicación representa las capacidades del sistema modular para la ejecución de tareas, la gestión de dispositivos y la flexibilidad del proceso. El sentido modular bajo la separación de operaciones de planeación y ejecución en la estructura de operaciones de la UC representa un alto coste en cuanto a programación de la ejecución del proceso para el maestro, pues representa todas las operaciones que el operador efectúa durante las operaciones físicas, garantiza los tiempos de operación, la detección de los errores y el mantenimiento de los dispositivos.

En cuanto a las pruebas de operaciones de planeación y configuración, se presenta la parte representativa del proyecto en la ejecución de “secuencia” el cual implica la mayoría de las funciones de la unidad de control tanto para la interfaz, como el maestro y los esclavos. La evolución de las instrucciones de operación y los bloques funcionales de cada módulo se muestran indirectamente; el resultado muestra la ejecución sin errores pues por el análisis del proceso y la obtención de la jerarquía de componentes, de funciones de operación y el establecimiento de la estructura del módulo en comunicación y manejador de solicitudes mantuvieron la estabilidad del desarrollo de pruebas, sin pérdidas en las líneas de comunicación por el consumo de corrientes en los actuadores quedando un resultado satisfactorio y esperado. Las fallas se aseguran ser independientes al sistema de control, quedando en la seguridad del sistema mecánico y por la fuente de alimentación.

El peso de los resultados no está en la aplicación particular de las funciones de operación, si no en la estructura modular del sistema para la automatización de procesos, el trabajo realizado hace énfasis en analizar por completo al proceso basado en jerarquía de componentes y la separación de las operaciones en la unidad de control para establecer un sistema modular. Uno de los objetivos del proyecto es que sea un sistema de control robusto para el control de los parámetros para garantizar los resultados de pruebas en ambiente controlado con fin de mejorar el análisis de efluentes por floculantes. La comprobación de la robustez para ejecución de operaciones está directamente relacionada por la estructura de comunicación

Hasta este punto del trabajo, se muestra el análisis de la robustez en ejecución de operaciones, a manera de validar el modelo funcional. Retomando los capítulos 4.1.6 y 4.1.4 en la jerarquía de funciones y tiempos de operación se establece indirectamente la necesidad de un sistema modular que permite multi tareas, bajo las condiciones estados de operación y con capacidad de determinar los errores por funcionamiento, en los cap. 4.2.2 y 4.2.3 se establece la estructura de los módulos de la unidad de control para las operaciones de planeación y ejecución, directamente relacionadas con el proceso.

Se han establecido las funciones del microcontrolador para que el sistema sea modular, se define la estructura de envío de solicitudes y el manejador de estas, a partir de esto se establece la ejecución de tareas por el set de instrucciones y su evolución, la detección de errores se define como servicios independientes, interrupciones, siendo una solución particular basado en microcontroladores. De las muchas soluciones posibles para tipos de comunicación, se establece unas líneas auxiliares de tarea y error en la comunicación I2C para reducir la carga de datos en las líneas de tarea, la planeación de operaciones y sincronizar las operaciones.

La importancia de enfocar la comunicación en la ejecución de los procesos es para detectar los errores y permitir al operador y al programador determinar la solución sin que el sistema de control y el mecánico se dañen, además de darle flexibilidad y multitarea con un sentido modular. El presente trabajo se plantea tres acciones para hacer robusto la unidad de control en su comunicación durante elección de operaciones: lectura de estado de tarea, conteo de tiempo de operación y lectura de estado de error de las cuales a partir de “quién termina primero” es como se establece el tipo de fin del tiempo de operación.

En las pruebas de comunicación y operaciones, cap. 5.2, se describe la evolución de las operaciones; de forma general se definen dos tipos de evolución, ejecución normal y ejecución con error, para esto, en la Figura 162 se muestra un diagrama de tiempos y estados. De lado izquierdo el maestro y de lado derecho un esclavo. La estructura general para operaciones inicia en el maestro con el envío de solicitud y habilitación de la interrupción por cambio de estado en la vía de error, luego se inicia el conteo de tiempo estimado para el tiempo de operación y se lee el estado de tarea. Para el esclavo, al llegar la solicitud se ejecuta la función asociada y se inicia el conteo de tiempo estimado para el sensor de la función. Los tipos de ejecución son:

- Una ejecución normal es aquella en la que todas las operaciones de tarea terminan antes del tiempo estimado de operación y sin detección de errores.
- Una ejecución con error es aquella donde pueden pasar tres cosas:
 - En el maestro, el tiempo estimado de duración de tarea es sobrepasado
 - En el maestro, algún esclavo detecto error
 - En el esclavo, en la función de tarea algún sensor paso el tiempo estimado de su operación

La evolución de una ejecución normal es la siguiente:

- Maestro: se lee los estados de tareas de los módulos activos y las lianas de error, se finalizan todas las tareas antes de que pase el tiempo esperado del proceso.
- Esclavo: el Timer es reiniciado cada que en la función de tarea el sensor cumple una operación, de tal forma que el fallo es esperado por alguna obstrucción mecánica.

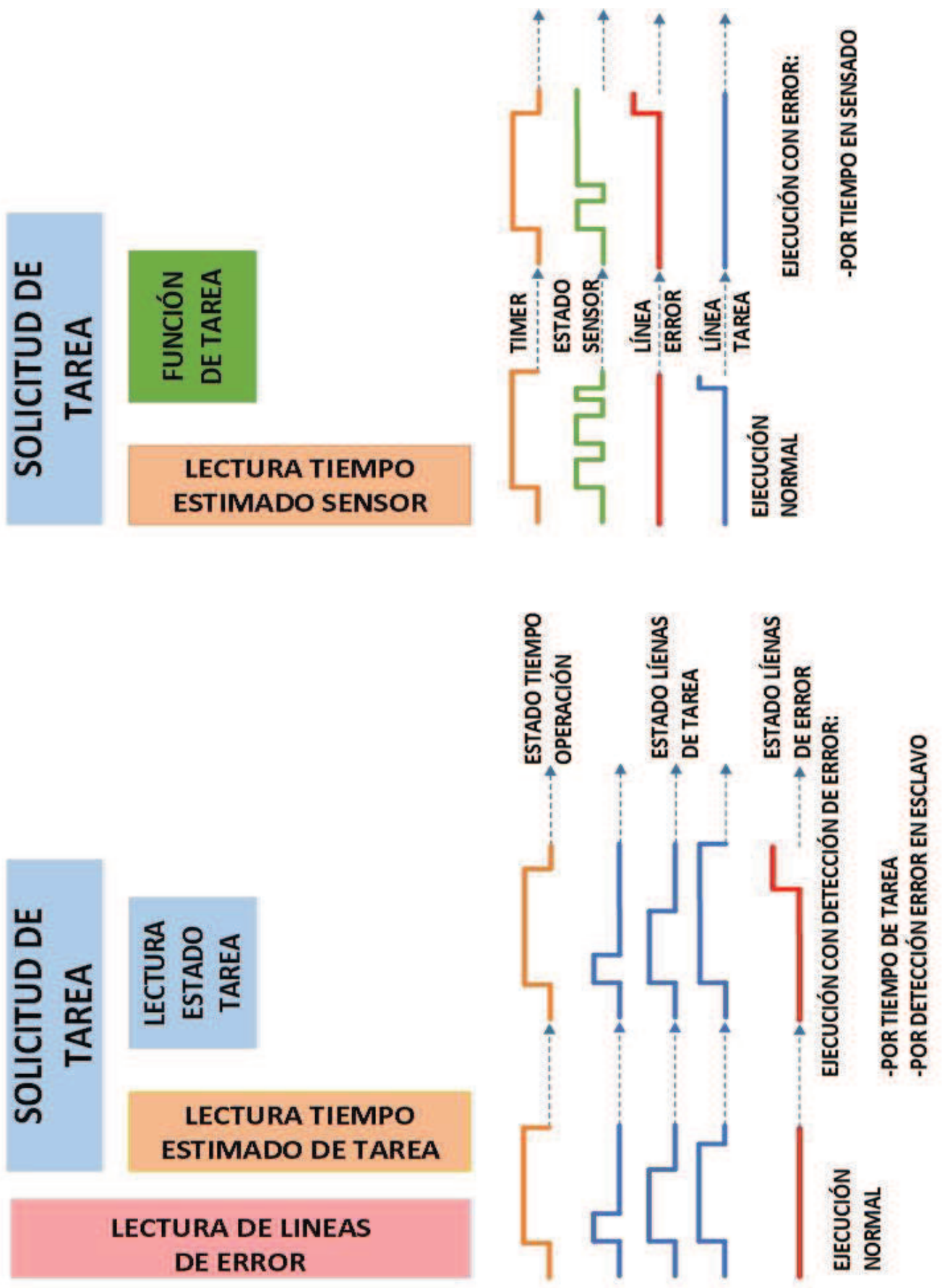


Figura 162. Detección de errores

La evolución de una ejecución con error es la siguiente:

- Maestro: se lee los estados de tarea, si el tiempo estimado del proceso es sobrepasado sin que las líneas de tarea sean puestas en 0 se notifica error a usuario, se fuerza el paro y solicita a cada esclavo sus estados. Si durante ejecución de tareas algún esclavo encuentra un error, antes de que termine el tiempo estimado y que la última tarea sea puesta en cero, se solicita paro y estados de los esclavos.
- Esclavo: cuando en ejecución de tarea el sensor no terminara una operación el Timer es sobrepasado, el cual atiba la identificación de error y reinicia sus estados esperando instrucciones del maestro, por otra parte, puede que el sensor cumpla su función, pero mecánicamente no se cumpla la condición de fin, siendo el maestro quien identifica por tiempo que, a pesar de no estar habilitado algún error, existe uno por fin de operación.

La detección mutua de errores en ambos extremos de la comunicación, por líneas auxiliares, permiten reducir la saturación de las líneas de comunicación, permiten establecer un sistema robusto para las operaciones y la capacidad de determinar el tipo de error. De esta forma, tanto por la comunicación, la ejecución de operaciones en la interfaz y del maestro – esclavo queda validado el modelo funcional, cumpliendo con las especificaciones y metas del proyecto en cauto a la unidad de control; la solución no única, no se puede establecer una comparación por dispositivos de implementación pero si en tiempos de ejecución, costos y el tipo de combinación.

CAP. 6 CONCLUSIONES

Actualmente se ha incrementado el desarrollo de nuevos dispositivos de control, tanto en industria como en la electrónica esto ha permitido que soluciones complejas puedan ser desarrolladas de forma eficaz en cuanto a tiempos y costos; la creciente tendencia al desarrollo de aplicaciones con microcontroladores con lenguajes de alto nivel o incluso gráficos ha dejado de ser un impedimento en cuanto capacidad y ha creado una globalización en la electrónica como en internet de las cosas IoT. Esta evolución fue determinante para el proyecto, pues en un inicio se planteó el uso de dispositivos Arduino, pero por la época, diciembre 2013, aun había una notable diferencia en cuanto a capacidades del uso de un microcontrolador sin funciones pre definidas, ahora su uso es considerado por los nuevos dispositivos de alta gama y de bajo costo, además de la gran literatura; no debe perderse la importancia de auxiliarse en métodos de diseño con el análisis pertinente para usar de forma eficiente las nuevas plataformas de desarrollo. En este sentido, el desarrollo de la unidad de control basado en microcontroladores muestra la alternativa a dispositivos de gama alta y las capacidades de aplicación mediante funciones básicas de un microcontrolador a partir del seguimiento de un método de diseño encaminado a cumplir objetivos y especificaciones.

Los resultados del sistema fueron satisfactorios, se cumplen las metas clave para el sistema de control sobre flexibilidad del proceso por ser modular, registro de pruebas al coordinar las operaciones de configuración y de comunicación, robustez por las capacidades del microcontrolador en cuanto a detecciones de errores en ejecución de tareas y operaciones de mantenimiento y configuración de módulo. En cuanto a las especificaciones de diseño se cumple la autonomía por la independencia modular, el control de mezclado mediante un módulo dedicado, adaptable por la flexibilidad de cambio de operaciones de planeación y el dispositivo de comunicación por las operaciones de planeación y de comunicación para la interfaz.

La solución del proyecto ha sido probada para operaciones de procesos mediante un microcontrolador particular, no representa un modelo único de solución pues existe una gran variedad de metodologías y dispositivos que pueden ser usados para resolver el mismo problema, por lo que no se compara ni se designa como mejor solución, si no se establecen los alcances considerando los tiempos de respuesta, las capacidades de operación, los periféricos, etc., para cumplir todas las especificaciones de un proyecto. Lo que se debe destacar es que no importa tanto los elementos con los que se resuelva, si no el tipo de solución.

Por otra parte, la unidad de control desarrollada puede ser implementada en diferentes aplicaciones, por permitir la multitarea y ejecución de funciones con identificación de errores; para adaptarse a la estructura desarrollada el proceso debe ser analizado por componentes para establecer mediante jerarquías y condiciones de operación las funciones del proceso; además, por el desarrollo general de la operaciones de la unidad de control y de la estructura de comunicación permite orientar la implementación en otros dispositivos.

La ventaja de sistemas modular ante secuenciales esta en operaciones de multitarea, la flexibilidad de cambios de operaciones de planeación, la identificación de errores y en funciones de mantenimiento; pero la descentralización de las operaciones de ejecución y las de paliación exige una coordinación entre maestro y esclavo, y en cuanto a velocidades de transmisión depende del número de dispositivos, por lo que para procesos discretos se adapta bien debido a que el tiempo no es determinante, no así en aplicaciones que requieran respuestas rápidas.

La automatización de los procesos requiere un análisis enfocado a conocer todas las características, estaos, condiciones y operaciones para poder determinar un pseudo algoritmo que describa y relacione todo este análisis, para después realizar su implementación. Al usar un sistema modular se requiere definir una estructura de comunicación que permite darle sentido de flexibilidad y multitareas para diferenciarlo de un sistema secuencial, una de las formas es dividir las operaciones de la unidad de control separando las de ejecución y planeación, con un costo relativamente alto de planeación, el cual es compensado por el análisis profundo del proceso. Existen diversos tipos de soluciones para la automatización de procesos, actualmente el enfoque modular permite no depender de un dispositivo único ni particular, permitiendo a diferentes dispositivos de diferentes familias de fabricantes realizar operaciones en conjunto sin depender de sus sistemas particulares.

La presentación de este trabajo tiene el objetivo de plantear un sistema genérico para la automatización de procesos de forma modular y basada en microcontroladores como opción de solución en un sistema que es definido como Mecatrónico, por incluir la entrada y acondicionamiento de señales con procesamiento y elementos actuadores de forma retroalimentada, implementados en un sistema físico que cumple con un conjunto de instrucciones con un fin, además de la interacción del usuario para reprogramar o configurarlo sin alterar el comportamiento esencial

6.1 Trabajo a futuro

Para la continuación de las etapas del proyecto, puede mejorarse el sistema de comunicación en cuanto a la consideración de otros microcontroladores, protocolos como RS485 o SPI permiten una comunicación más flexible por presentar líneas selectoras [28].

En cuanto a los módulos, pueden realizarse mejorar en las etapas de potencia o censando, al cambiar el tipo de actuador o de sensor; la modificación de cada módulo representa únicamente un análisis del nuevo proceso a desarrollar, por la separación de las operaciones de comunicación; por otra parte en el sistema de dosificación de floculante, puede agregarse un algoritmo de control manteniendo la estructura del módulo, o bien pueden cambiarse los paradigmas de manejo de funcione de operación.

Por el uso de líneas de comunicación de erro y tarea, se requiere en el módulo maestro compuertas OR para concatenar estas líneas en lectura de un pin; para las líneas de error se requiere leer cada una de las líneas para determinar el dispositivo que ha fallado, para aplicaciones con dispositivos mayores puede establecerse el uso de otro tipo de lectura para reducir el número de pines, tal que permita identificar todos los elementos con error.

La unidad de control puede ser implementado para otro tipo de aplicaciones como en domótica, en estación de monitoreo, en invernaderos o auxiliar para robots móviles, considerando operaciones de milisegundos.

BIBLIOGRAFÍA

- 1 IQ. Eduardo Suarez Oquendo (2011). Diseño y construcción de un equipo de prueba de jarras para la tratabilidad de aguas residuales.
- 2 Ing. Bernardo Servín Massieu (2008). Tratamiento de aguas residuales. México
- 3 Dolores Sanz (2008). Problemas y soluciones para el análisis de la calidad del agua. España
- 4 Samanta Giordani, Luzmari Gonzáles, Astraid Morales (2008). Factibilidad del uso de polímeros para la clarificación del agua potable en la ciudad de Mérida. U.E. Colegio "La presentación", Mérida
- 5 Ing. Darlyn Alejandra Valdés Uribe (2012). Diseño preliminar de un sistema de tratamiento de aguas residuales escala industrial para los efluentes del procesamiento de pieles. Quito
- 6 Ing. Manuel López Cano, IQ. Olga Marecos (2011). Evaluación de plantas de tratamiento. Paraguay
- 7 Honghui Zhu, Daniel W. Smith, Honngde Zhou (1995). *Improving removal of turbidity causing materials by using polymers as a filter aid*. Canadá
- 8 Maricruz Vargas Camareno, Luis Guillermo Romero Esquivel (diciembre 2006). Aprovechamiento de algunos materiales en el desarrollo de coagulantes y floculantes para el tratamiento de aguas en Costa Rica. Costa Rica
- 9 B. Abramovich, M.C. Lura, E. Carrera, M.I. Gilli (2004). Acción de distintos coagulantes para la eliminación de *Cryptosporidium* spp. En el proceso de potabilización del agua. Argentina
- 10 Jaime Humberto Carvajal Rojas (Junio 2005). Automatización y mecatrónica en la educación. Universidad de La Salle
- 11 C.F. Silva, J. Marcos, J.I. Armesto (2011). Autómatas programables y sistema de automatización. Universidad de Vigo
- 12 Lewis R.W, (1997). *Programming industrial control systems using IEC 1131-3*. IEE Control Engineering series 50, ISBN-0 85296 950 3
- 13 C. Petitpierre & A. Eliëns. *Active Objects Provide Robust Event-driven Applications*. SERP'02, Las Vegas, June, Pages 253-259, 2002.
- 14 M.I. Pedro Javier Gálvez Valdez (2015) Desarrollo de un banco de pruebas para el estudio de robots móviles mediante el protocolo ieee 802.11 en tiempo real con retroalimentación visual. México
- 15 Murphy R.R (2000), *Introduction to AI Robotics*, MIT Press
- 16 Michel, G. (1990), Autómatas programables industriales, Ed Marcombo
- 17 Pere Ponsa & Antoni Granollers(2009), Diseño y automatización industrial; Universidad politécnica de Cataluña
- 18 ABB México, artículo 2013-09-20, la automatización en la industria discreta versus la automatización en la industria de procesos. <http://www.abb.com.mx/>
- 19 N. Iriondo, F. Artaza, S. Calvo (2004), Metodología para el diseño jerárquico de sistemas de control industrial validado mediante modelado basado en componentes. País Vasco
- 20 José Mari Gonzales de Durana (Febrero 2004), Automatización de procesos industriales.
- 21 infoPLC_net_Structuring_program_development_Spanish_Version.pdf, pág. 1]
- 22 Josep Balcells, José Luis Romeral, Autómatas programables, Ed Marcombo
- 23 Karl T. Ulrich, Steven D. Eppinger, Diseño y desarrollo de productos, 4ta Ed McGrawHill
- 24 Eelco van der Wal, Desarrollando la estructuración de un programa con IEC 1131 - 3
- 25 Instituto Tecnológico de Celaya, Paradigmas de programación
- 26 Bibiana M. Vallejo, Sandra B. Vallejo (2006). Aspectos generales de la automatización industrial del sector farmacéutico. Colombia
- 27 Martin Pavlovic, Fotis N. Koumboulis, Maria P. Tzamtzi (2007), *Role of automation agents in agribusiness decision support systems*. USA
- 28 Microchip Technology Inc, Joseph J., Communications with the EUSART,reference AN1230 - SAE J1708/J1587

APÉNDICES

- A Fundamentación de Proyecto Lodos
- B Estructura mecánica, Proyecto Lodos
- C Interfaz de usuario, Proyecto Lodos

A. Fundamentación de Proyecto Lodos

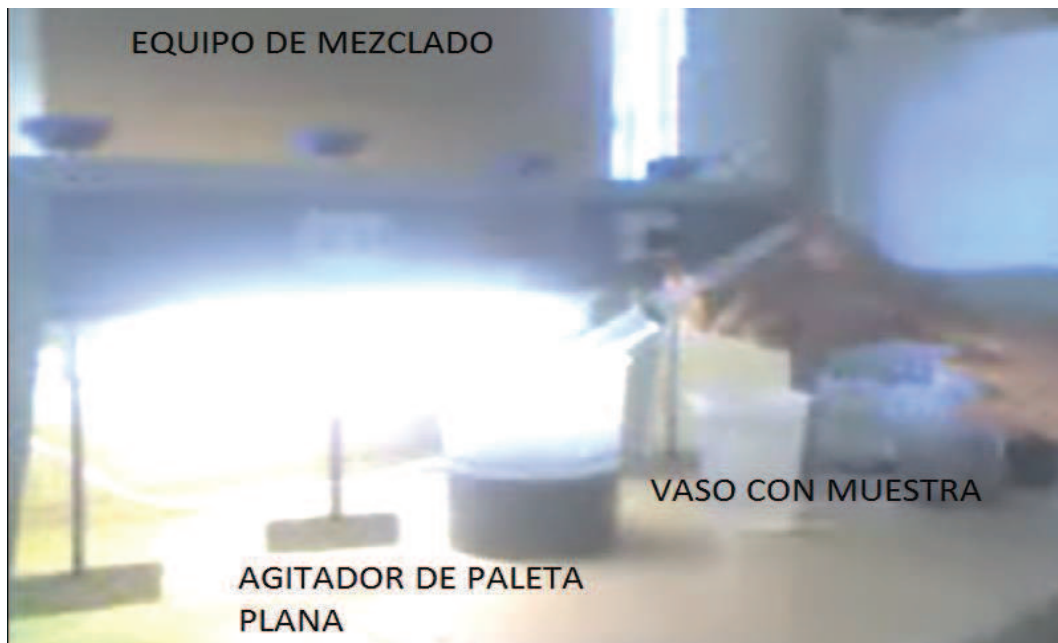
M.I. Arianna I. Sánchez Gutiérrez, M.I. H. Hiram Castillo Rosales, I. Roberto Carlos Oropeza
SNF Floerger de México, noviembre 2012

El lodo que se quiere tratar es una mezcla heterogénea, que provienen de una fuente no constante de contaminantes, estos lodos pueden variar en su composición a lo largo del año y dependiendo también del lugar donde se extraigan las aguas residuales; esto quiere decir que es necesario realizar la prueba de lodos siempre que la composición del lodo cambie de manera significativa o siempre que se tenga un nuevo cliente.

Prueba de jarras en la empresa

Para encontrar la dosis óptima de floculante se realiza una sencilla prueba de laboratorio (no hay un único procedimiento para hacer esta prueba y en este trabajo se menciona el procedimiento que la empresa utiliza), esta prueba consiste en agregar a una muestra de lodo una dosis de floculante con una concentración conocida, esta mezcla se agita y se filtra, a la par se miden parámetros como la cantidad de líquido filtrado y la masa de la torta, lo que se pretende obtener con esta prueba es un parámetro llamado “resistencia específica”, este procedimiento se repite agregando floculante con diferentes concentraciones, finalmente se ubica la dosis optima la cual se encuentra en el punto mínimo, en la Figura 1.3 se muestra una gráfica típica obtenida de este test. A continuación se muestra el proceso que realizan en planta.

Preparación de la mezcla



1. El equipo azul es un equipo convencional de mezclado que tiene cuatro agitadores que giran en paralelo a una velocidad entre 0 y 250 revoluciones por minuto.
2. El agitador es una paleta plana.
3. El vaso contiene una muestra de lodo, entre 200 y 400 ml.
4. A la muestra de lodo se le agrega polímero.
5. A la mezcla de lodo-polímero se le somete a agitación como se puede observar.

Mezclado



1. El tiempo de agitación es de aproximadamente de 10 a 20 segundos.
2. Como se puede observar, el polímero reacciona con el lodo formando flóculos.
3. Esta reacción provoca que el lodo se separe del agua. Este fenómeno se da en las tuberías de conducción de lodo después de haber reaccionado con el polímero.
4. El tamaño del flóculo formado se califica del 1 al 10 dependiendo del tamaño. El 1 es un flóculo prácticamente imperceptible y el 10 es un flóculo de gran tamaño.

Recolección



1. El lodo floculado, separado del agua, se vierte en un matraz buchner que tiene una malla con las mismas características de las que se utilizan en los filtros con banda industriales que ustedes vieron en Toluca.
2. Debajo del buchner se instala una probeta en la que se registrará el volumen de filtrado recolectado en 10 y en 20 segundos.

3. La curva de desempeño del polímero en cuanto ha filtrado es una parábola invertida. Es decir, empieza con valores de drenado bajos a bajas dosis de polímero, suben hasta alcanzar un máximo y después descienden conforme se incrementa la dosis de polímero.
4. Los valores de drenado a los 10 y 20 segundos se anotan.

Separación



1. Después de que termina el drenado del lodo que queda en el buchner, sobre la malla, se observa la calidad del filtrado.
2. Se asigna una calificación entre 0 y 3. 0 es un filtrado muy sucio, 2 regular y 3 extremadamente limpio.
3. La torta de lodo desaguado queda en el filtro buchner para su análisis posterior.

Análisis



1. La torta del lodo se desprende del filtro buchner.
2. Se califica la limpieza de la malla.
3. Se califica la resistencia de la torta exprimiéndola entre las manos.

4. Si la torta es de buena calidad, al exprimirla prácticamente todo el lodo se quedará entre las manos. Significa que el polímero está manteniendo fuertemente unidas las partículas de lodo y que en los filtros con banda industrial resistirá el esfuerzo que ejerce la banda sobre la torta sin que el lodo se derrame por los costados de los filtros.
5. Si la torta es de mala calidad, al exprimirla prácticamente casi todo el lodo se escurrirá entre las manos. Significa que el polímero no tiene la capacidad suficiente para mantener las partículas de lodo unidas y que si se utiliza a nivel industrial, la torta se perderá por los lados de las bandas.

El problema es que existen cerca de 1000 polímeros y que en cada evaluación por lo menos hay que realizar aproximadamente unas 20 evaluaciones que demandan aproximadamente 20 minutos cada una.

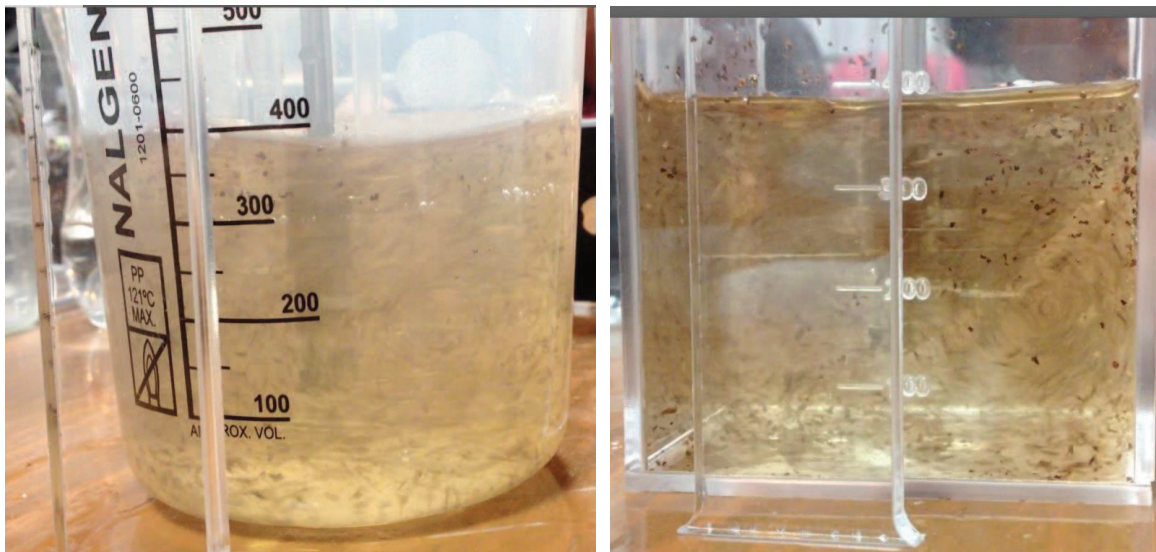
El objetivo es poder realizar la agitación, la filtración, la calificación del filtrado y la consistencia de la torta de manera automática con una mínima intervención de los técnicos.

Pruebas de jarras en laboratorio

A continuación se presenta las pruebas de ambiente controlado realizadas para el análisis de proceso de pruebas que realiza la empresa.

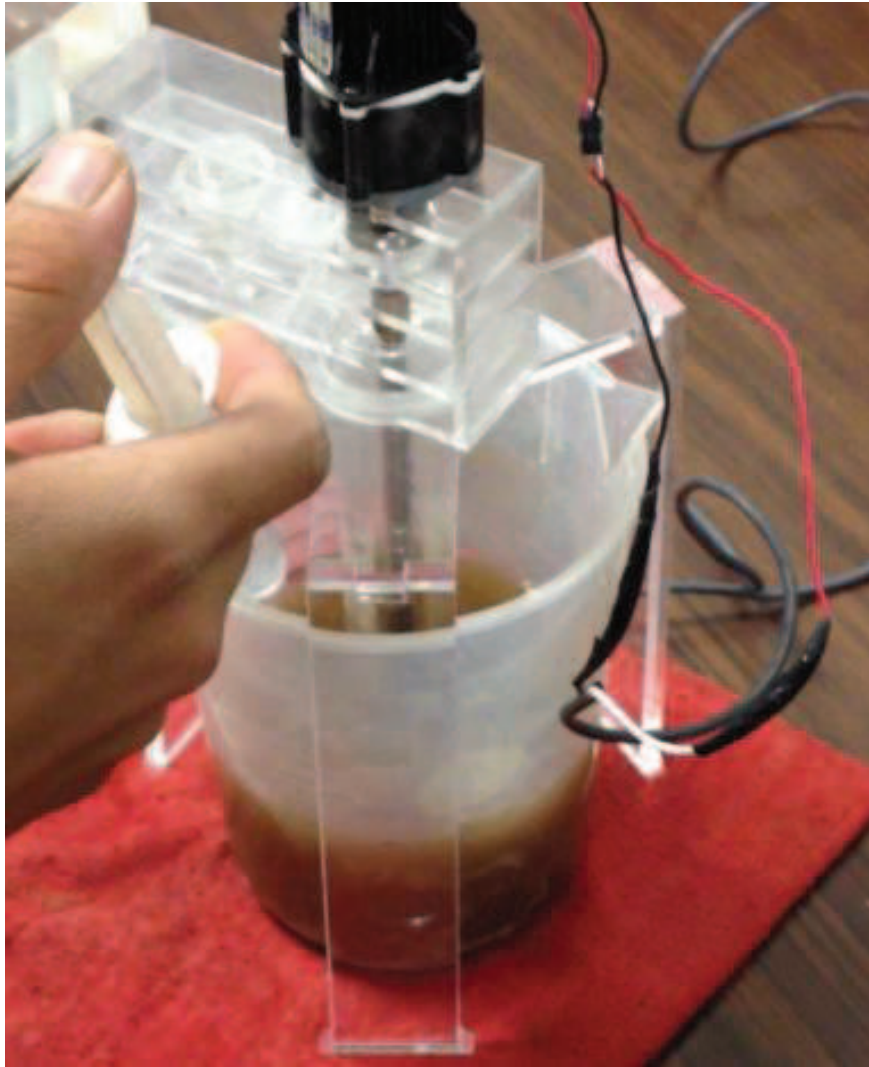
Prueba 1.- 17 de noviembre de 2012.

- Preparación del polímero: 500 ml agua 0.5 g de polímero
- Prueba a.- vaso circular y las aspas propuestas
- Prueba b.- vaso cuadrado y aspas rectangulares



Prueba 2.- 23 de noviembre de 2012

- Preparación del polímero: 250 ml agua 0.25 g de polímero.
- Se revolvió por aproximadamente 2 horas (verificar dato)
- Preparación del lodo: 300 ml de agua con tierra fina con 20 ml de polímero preparado
- Tiempo de agitación (contado desde que se vacía la primera jeringa de polímero): 20 segundos



Consideraciones y observaciones de las pruebas

PROS:

- La separación de la torta y el agua fue instantánea.
- La propuesta del vaso redondo y dos aspas que giran encontradas cumple con el objetivo de revolver la mezcla polímero-lodo sin romper el flóculo. (Falta evaluación de un experto)

CONTRAS:

- Dado que ese no es el vaso ni el lodo que se utilizará se tendrá que repetir las pruebas en las condiciones exactas en que será utilizado, sin embargo se espera que esta variación no cambie los resultados obtenidos.

PLAN DE ATAQUE

- Acoplar un motor comercial, verificar medidas, rpm, etc.
- Fabricar un vaso con las dimensiones correctas.
- Ajustar dimensiones del vaso y el motor en planos para mandar a maquinar.
- Modelado de la tubería.

El método que actualmente se utilizan para encontrar las concentraciones óptimas de floculante tienen algunas desventajas, ya que la mayoría de las pruebas se tienen que hacer decenas de veces con un mismo producto a diferentes concentraciones, debido a esto, se genera un trabajo que es muy monótono y aburrido para el laboratorista, además estos experimentos se realizan a mano, se genera poca repetitividad y exactitud en los resultados, si a esto le agregamos que la única máquina que se utiliza para acelerar la prueba es una mezcladora para jar-test Figura 1.4, cuya única función es agitar las muestras, nos encontramos con un proceso que si se automatiza podría presentar grandes ventajas.



Figura 1.4. Equipo jar-test utilizado en la prueba de lodos (4).

Entre las principales mejoras al automatizar esta prueba es que se tendría total control de parámetros como tiempo de agitado, además la repetitividad del proceso aumentaría y se podrían hacer pruebas con concentraciones diferentes de polímeros a intervalos más cerrados, esto tendría como resultado un efecto directo sobre la precisión de la dosificación óptima de polímero, lo cual se reflejaría directamente sobre el costo del proceso al ocupar sólo la cantidad necesaria de poli electrolito para tener el proceso más eficiente, si a esto le agregamos que el banco de pruebas se diseñaría pensando en que no se necesite de una capacitación especial para operarlo ni de un mantenimiento costoso, el costo de operación del prototipo sería muy bajo.

Sistemas

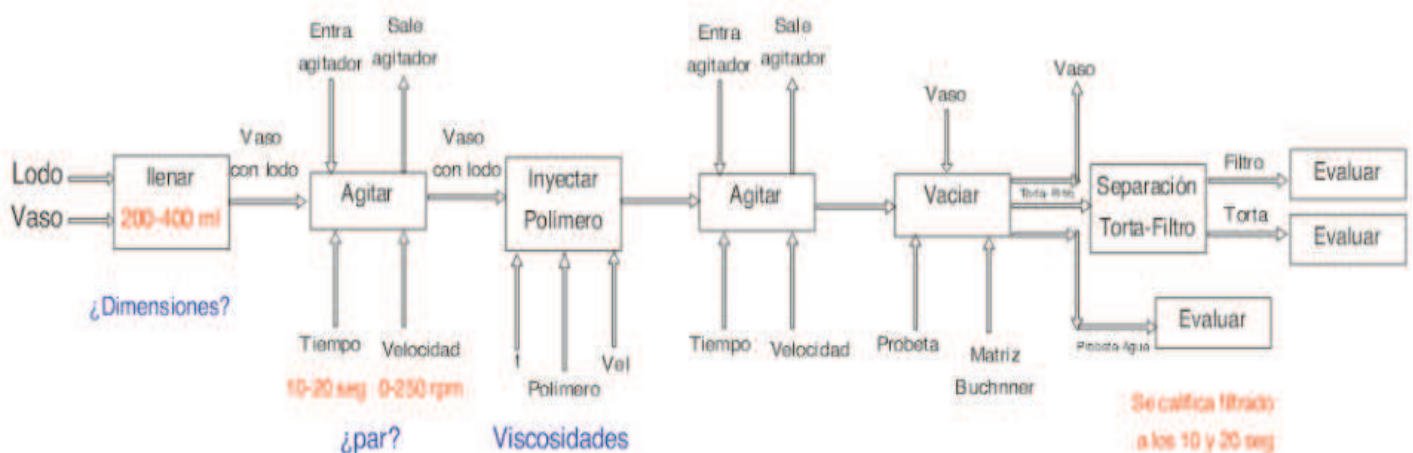


Figura 1.5. Diagrama de bloques de la prueba de lodos.

Sistemas necesarios en el prototipo, sin considerar el análisis de las muestras:

-
- Diagrama de un sistema de procesamiento de polímeros, mostrando los siguientes componentes:
- Sistema de Llenado
 - Sistema Carrusel
 - Sistema de Lavado
 - Sistema de Inyección de Polímero
 - Sistema Sube
 - Sistema de Agitado
 - Sistema Voltea

Figura 1.6 Diseño conceptual del prototipo del banco de pruebas para la prueba de floculación

Después de una lluvia de ideas, se propuso tres configuraciones diferentes, cada una integrando los sistemas que se habían identificado como necesarios, finalmente se eligió un sólo concepto que será el que a partir de aquí en adelante se desarrollará, en la Figura 1.6, se muestra un esquema del diseño conceptual del prototipo, se observa la primera propuesta de donde se encuentran ubicados los sistemas.

Observando el diagrama anterior el equipo se percató que se sería necesario transportar los vasos donde se harán las pruebas, entre los diferentes sistemas propuestos, el concepto de este sistema fue una figura geométrica similar a un carrusel, de donde se deriva su nombre. El sistema carrusel entonces debe ser capaz de adaptar sistemas puntuales como el de lavado, inyección polímero, y el de agitado, siendo una carcasa móvil para los contenedores de las muestras.

La planeación de la estructura del proyecto está definida por etapas para darle flexibilidad en cuanto a adaptaciones y cambios no previstos a la fecha. De esta forma la primera etapa queda definida para el diseño de la estructura de carrusel, debido a que esta integrara los sistemas y de ella dependerán sus diseños mecánicos. La propuesta de las demás etapas para el desarrollo de los demás sistemas se propone con base en los tiempos de construcción e implementación; queda estructurado de siguiente manera:

- Etapa 1: posicionamiento
- Etapa 2: agitado
- Etapa 3: dosificación
- Etapa 4: lavado
- Etapa 5: recolección de muestras

Se agrega una quitan etapa para el posterior desarrollo del sistema de recolección de muestras para su análisis. Se consideró los tiempos de desarrollo y junto con en análisis del proceso de laboratorio.

B. Estructura mecánica, Proyecto Lodos

M.I. Arianna I. Sánchez Gutiérrez, I. Roberto Carlos Oropeza, Antonio Fonseca Alvarado.
SNF Floerger de México, abril 2014

Introducción

El proyecto LODOS termina la primera Fase con la construcción la estructura general de mezclado, la cual integra a “sistema carrusel” y “sistema voltea”. Esta estructura físicamente está dispuesta para agregar los sistemas con forme se haga el avance del proyecto.

Las tareas de los sistemas carrusel y voltea son de prioridad debido a que se considera que la automatización será por producción secuencial, con lo que la disposición de los contenedores y la mezcla son primordiales pues los demás sistemas serán implementados por posicionamiento. Así, las tareas de los sistemas mencionados pueden definirse como:

- Sistema carrusel: Posicionamiento en tres etapas de los contenedores para dosificación de lodo y polímero además de su mezclado, para la recolección de la muestra y para el lavado
- Sistema giro: Del posicionamiento de carrusel, es la etapa donde se vacía el contenedor para el sistema de recolección.

Las consideraciones mecánicas para su manufactura fueron basadas en que el ambiente de trabajo será corrosivo y las sustancias son de origen industrial, por lo que se pensó en usar aleaciones de acero inoxidable y alejar los actuadores de los componentes. Retomando el análisis del proceso de floculación de la empresa, los pasos ordenados de forma general, partiendo de que los contenedores son la parte más importante durante todo el proceso debido a que en ellos se lleva a cabo desde la dosificación hasta la entrega de muestra, son:

- Dosificar “lodo”
- Dosificar polímero
- Revolver
- Obtener muestra
- Analizar

De esta forma, las entradas al sistema con sus consideraciones quedan expresadas en la siguiente tabla:

Sistema	Entrada	Consideraciones
Sistema llenado lodo	Cantidad de lodo	De 250ml – 400 ml
Sistema inyección polímero	Cantidad de polímero	0-20 ml
	Polímero usado	100 polímeros diferentes
Sistema mezclado	Revoluciones de mezclado	De 0 a 250 rpm
	Tiempo de mezclado	De 10 a 20 segundos
Sistema carrusel	Evaluaciones por polímero	20 por cada uno

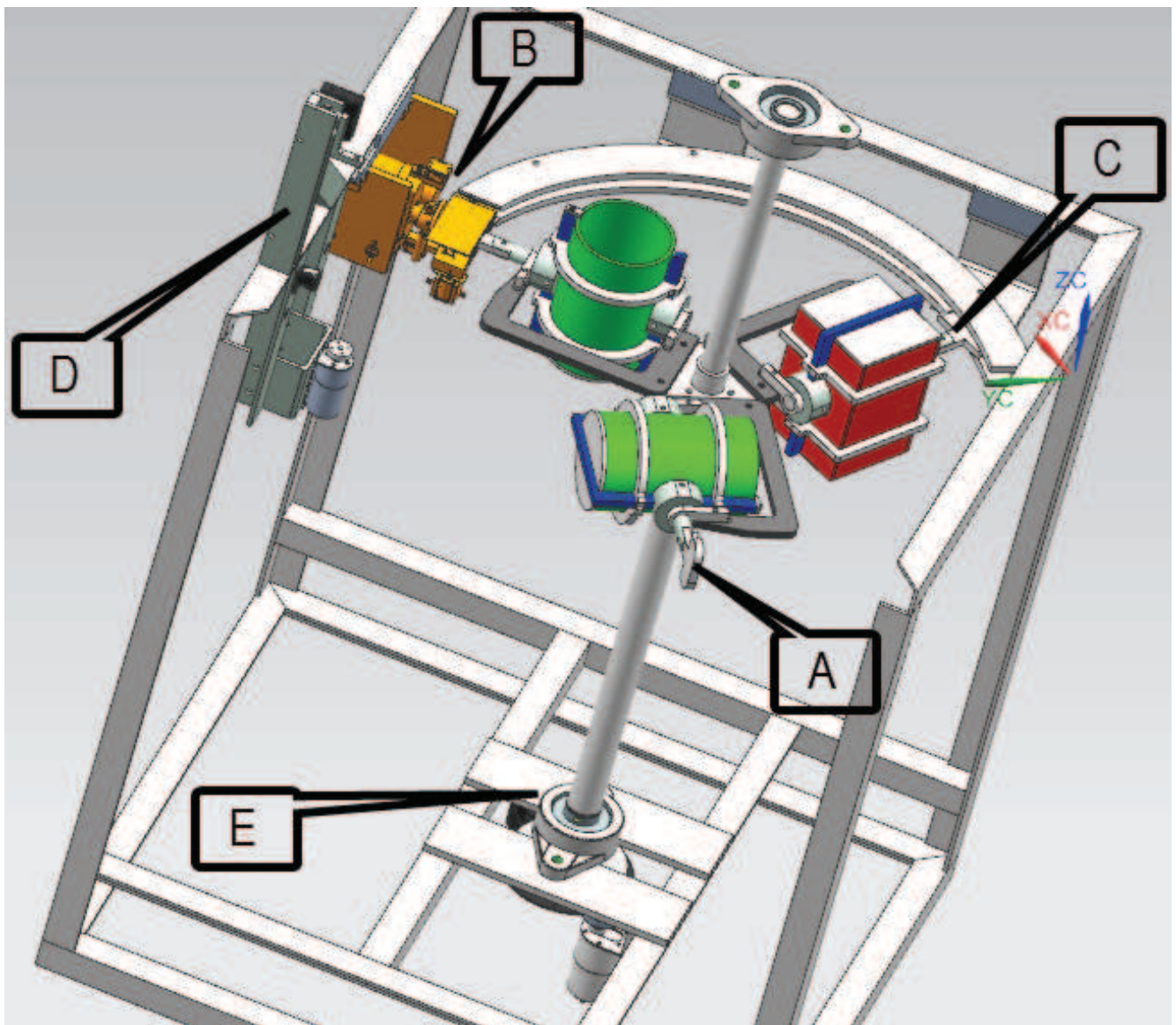
Debe observarse que los sistemas de lavado, sistema voltea y sistema sube no requieren un análisis complejo pues su accionamiento es por posición y únicamente realizan tareas simples, como lavar, orientar y posicionar respectivamente. Así el peso de estos sistemas será el diseño mecánico.

Descripción de estructura mecánica

En la imagen se muestra la estructura de carrusel; se encuentra compuesto por los siguientes elementos donde se integrarán los sistemas del proyecto, estas posiciones son:

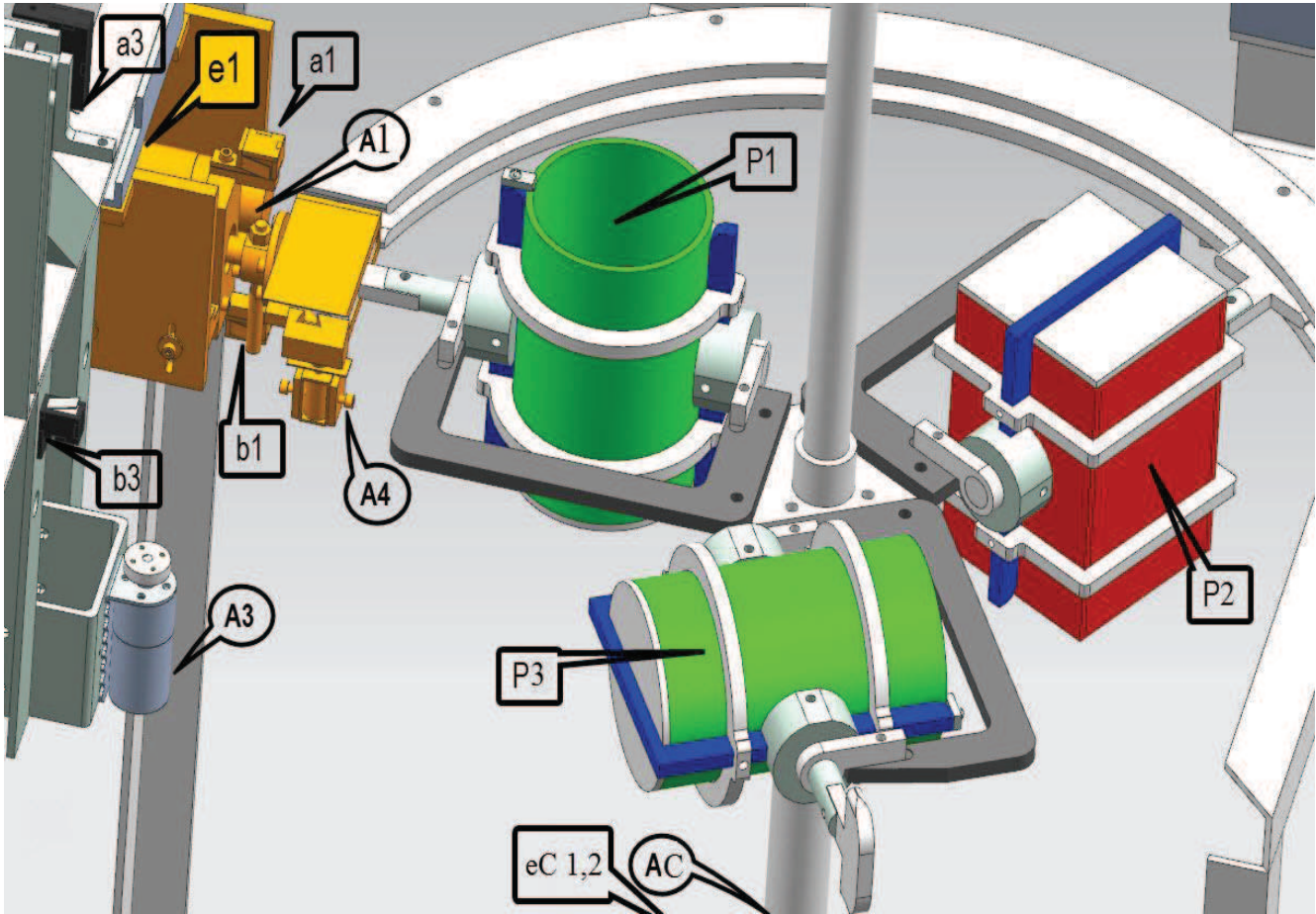
- F. Giro de contenedor, para después del lavado debe orientarse para la dosificación de lodo.
- G. Giro de contenedor, para que la mezcla pase al sistema de recolección de muestra
- H. Lavado de contenedor, en esta posición estará el sistema de lavado
- I. Posicionamiento batidora para el sistema de mezclado y dosificación polímero
- J. Posicionamiento carrusel, para las tres Fases del proceso: llenado, mezclado y lavado

Nota: Debido a que en las posiciones B y A se trata de un posicionamiento del contenedor, para estos casos se implementara el mismo sistema.



A partir de la estructura mecánica, se definen las posiciones que intervienen así como los actuadores y sensores necesarios para el prototipo final. Donde A es actuador, P posición, E encóder, a posición superior y b posición inferior.

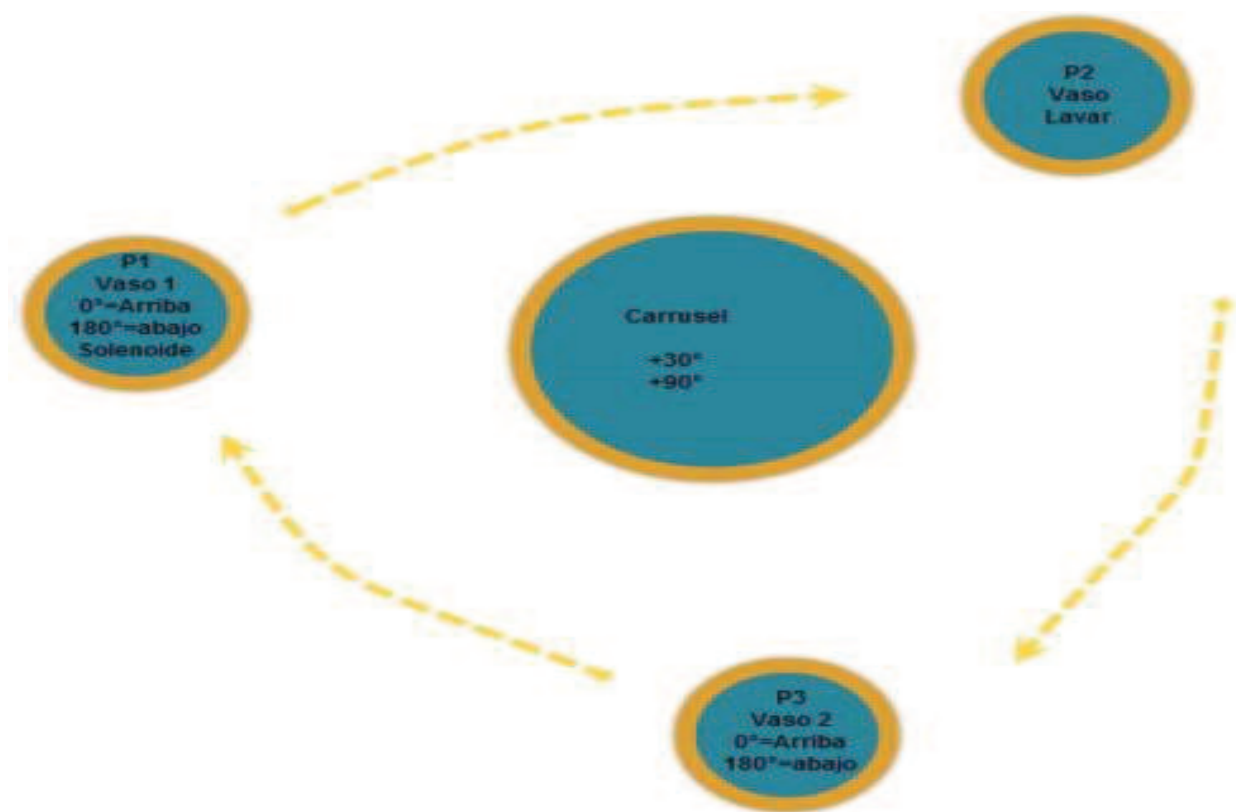
Posición	Proceso	Actuador (Actual)	Sensor estado	Etapas proyecto
Posición 1	Vaciado de contenedor	A1	a1 e1 b1	1
	Posicionamiento de agitador	A3	a3 b3	2
	Mezclado	-	-	2
	Dosificación polímero	-	-	3
Posición 2	Lavado	-	-	4
Posición 3	Orientación de contenedor	A2	a2 e2 b2	1
	Dosificación de lodo	-	-	3
Posicionamiento contenedor		AC A4	eC1 eC2	1



Por consideración mecánica para la implementación electrónica, se agregan sensores de posición y encoders para los actuadores encargados de la orientación, tanto de los vasos como del carrusel, donde se determinó que para el carrusel se requerían dos encoders, uno para posicionamiento y otro auxiliar de su movimiento con el fin de asegurar tanto para el carrusel como para los vasos su adecuada orientación. La lista de estos elementos se encuentra en la siguiente tabla:

Actuadores	Referencia	Tipo	Función
	A1	Motor DC	Orientación vaso en P1
	A2	Motor DC	Orientación vaso en P3
	A3	Motor DC	Posicionamiento vertical de agitador
	A4	Solenoide	Liberación carrusel
	AC	Motor DC	Orientación carrusel
Sensores	a1	Posición, superior	Asegurar orientación vaso en P1
	b1	Posición, inferior	
	e1	Encóder	Posición angular de vaso en P1
	a2	Posición, superior	Asegurar orientación vaso en P3
	b2	Posición, inferior	
	e2	Encóder	Posición angular de vaso en P3
	a3	Posición, superior	Asegurar posición de agitador
	b3	Posición, inferior	
	eC1	Encóder	Posición angular de carrusel
	eC2	Encóder	Auxiliar en posición de carrusel

Descripción lógico-programable del proceso de mezclado



De forma general, se puede sintetizar las posiciones y sus acciones en el siguiente diagrama circular. Donde la orientación de los vasos es:

- 0° posición boca arriba
- 180° posición boca abajo

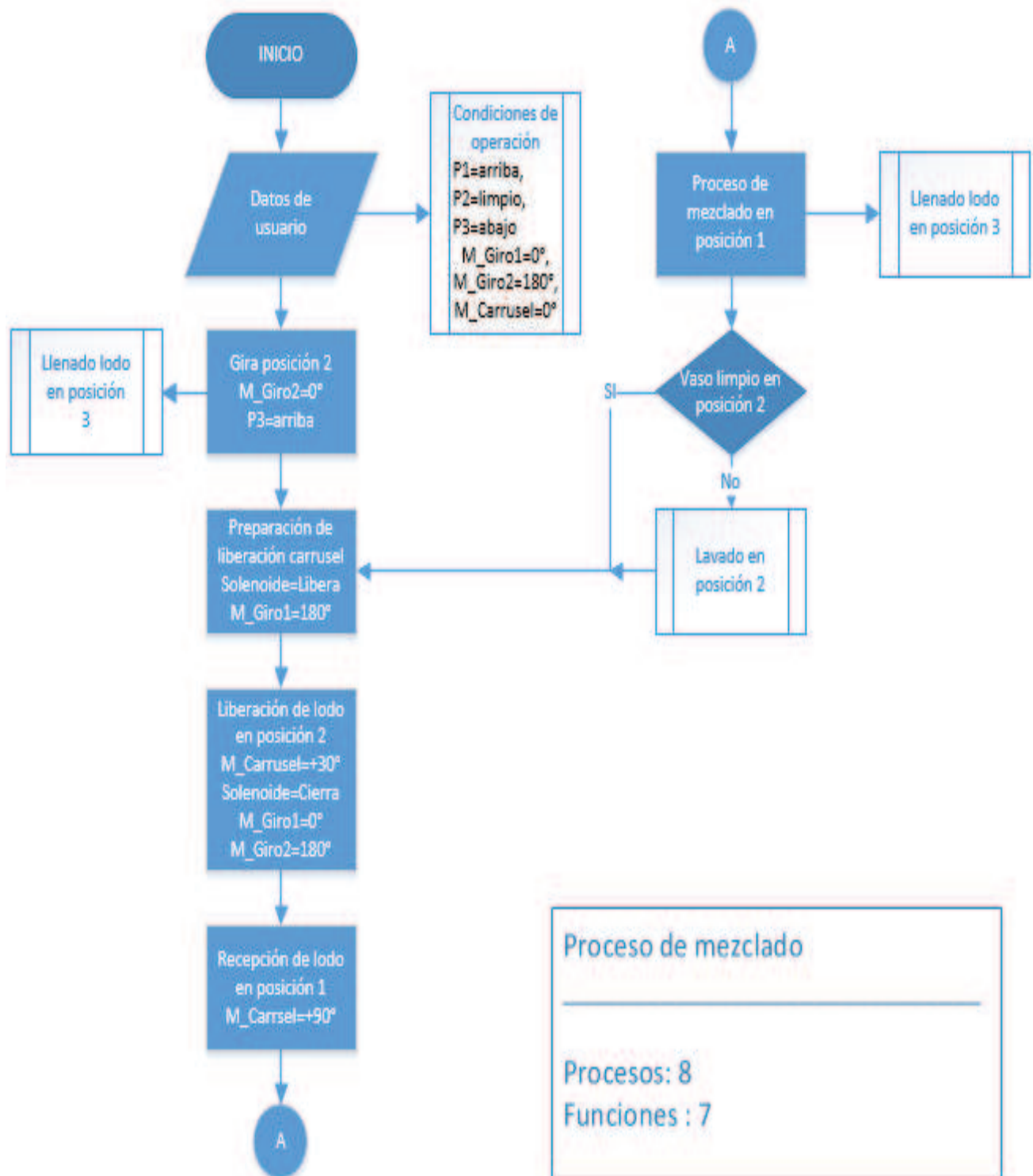
La orientación del carrusel es:

- 30° de avance para liberación de carrusel. Y permitir el giro de los actuadores
- 90° para terminar de recorrer una posición

Por la estructura física, se tienen las siguientes condiciones de inicio de operación:

- Vaso 1 en la posición 1, boca arriba
- Vaso en la posición 2, boca abajo
- Vaso 2 en la posición 3, boca abajo.

En la siguiente figura se muestra el algoritmo secuencial en relación al proceso de laboratorio:



De acuerdo al proceso de laboratorio se pueden resumir las siguientes Fases del proceso para la implementación en la estructura:

- I. Se reciben los datos del usuario (velocidad, tiempo, polímero, iteraciones).
- II. A2, del vaso 2 en la posición 3, cambia de orientación (boca arriba 0°). El vaso 2 en la posición 3 se llena con lodo.
- III. A4, solenoide en posición 1, se activa para liberar movimiento del carrusel. A1, del vaso 1 en posición 1, cambia de orientación (boca abajo 180°).
- IV. A3, del carrusel, gira 30°. A4, el solenoide, se apaga. A1, del vaso 1 en posición 1, cambia de orientación (boca arriba a 0°). A2, del vaso 2 en posición 3, cambia de orientación (boca abajo 180°).
- V. A3, del carrusel, gira 90° para posicionar lodo de 3 a 1. Se inicia secuencia de Mezcla y dosificación en la posición 1
- VI. A2, del vaso 2 gira en la posición 3, cambia de orientación (boca arriba 0°). El vaso 2 en la posición 3 se llena con lodo.
- VII. Se pregunta si el vaso en la posición 2 está limpio: No, se inicia secuencia; Si, Sigue

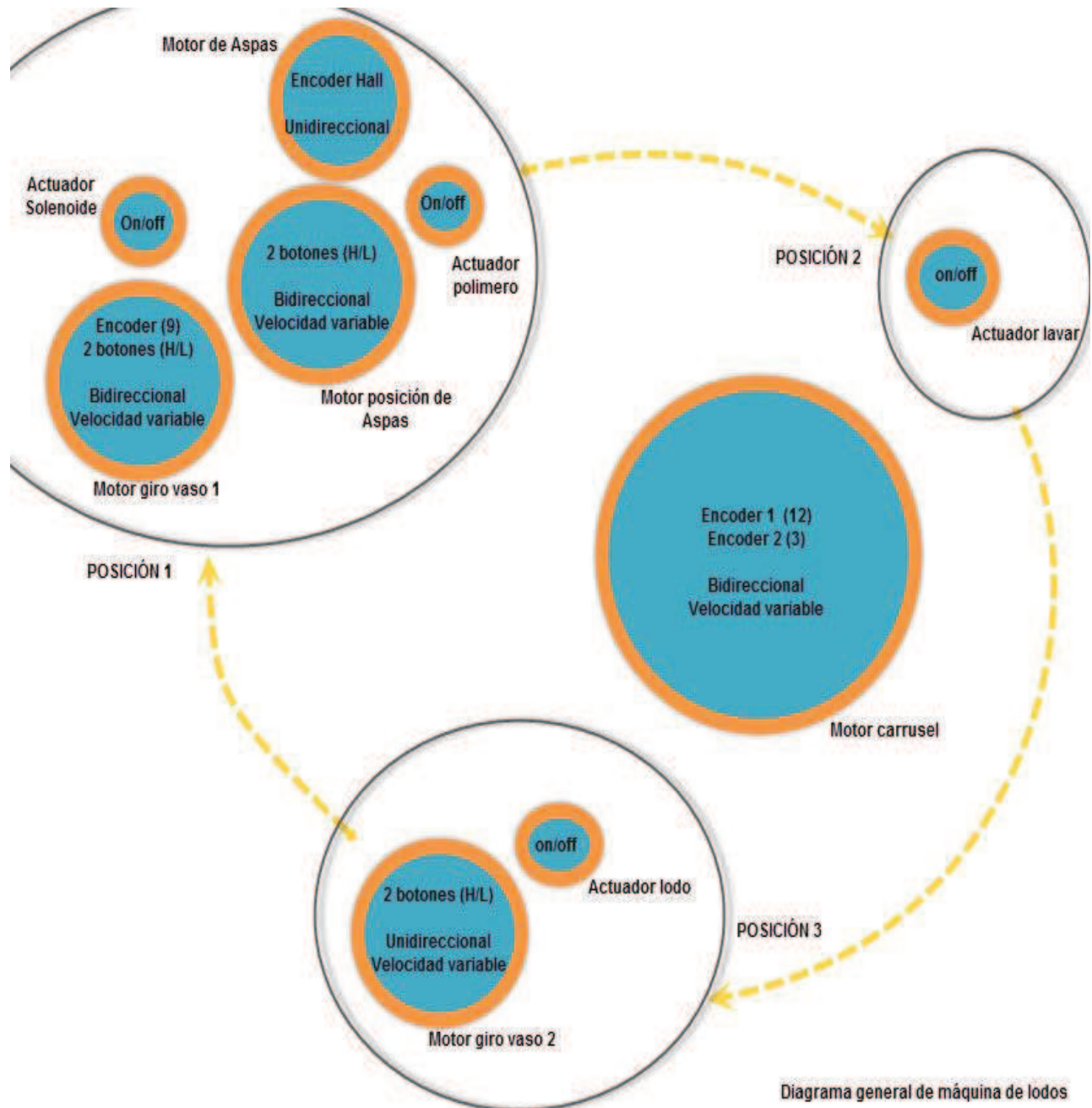
Se reinicia el ciclo en III.

Propuesta secuencial de funciones para la programación secuencial, noviembre 2013

La propuesta para la programación del prototipo costa de 8 funciones para la secuencia.

Función	Descripción	Consideraciones	Proceso
(0)	Garantiza condiciones de inicio de operación, estado 0.	Posiciones de máquina y datos de usuario.	I
(1)	Se introduce lodo sobre el vaso2 de la posición 3.	Vaso orientado boca arriba, 0°. Carrusel posición por giro de +90°.	II
(2)	Encendido de solenoide antes de que el motor de giro de vaso 1 pase a 180° (boca abajo).	Por condiciones físicas del solenoide, no es posible activarlo cuando va contra gravedad.	III
(3)	El motor del carrusel gira 30°.	Pasa solo cuando el motor del vaso 1 en posición 1 está en 180°.	IV
(4)	Apagado de solenoide, con lo que el motor de giro de vaso 1 va a 0°.	Posición 1 recibe vaso boca abajo, y el motor de giro de vaso 2 va a la posición de 180°.	IV
(5)	El motor del carrusel gira +90° con lo que queda en posición.	Se debe garantizar que el motor de giro del vaso 1 este en 0°, boca abajo.	V
(6)	Secuencia de mezcla de lodo con polímero.	Bajan las espas, revuelve, se ingresa el polímero, termina de revolver y suben las espas.	V
(7)	El motor de giro de vaso 2 se posiciona a 0°	Para voltear el vaso que recibió por (5).	VI
(8)	Se lava en la posición 2.		VII

Relación del proceso con el sistema mecánico



Entradas por usuario

- Velocidad de aspas
- Tipo de polímero
- Tiempo de giro de aspas
- Iteraciones

Entradas Adicionales

- Para la calibración manual, 2 botones
- Pantalla resistiva para interfaz

Observaciones, enero 2014

- Primera iteración: No se requiere lavar el vaso en la posición
- Última iteración: No se requiere llenar lodo ni hacer la mezcla
- Proceso I.: Conviene que el estado inicial de la posición 3 sea en 0° , P3=arriba, debido a que en la última iteración, si no se agrega lodo, termina en 0° . Esto quita un proceso. P1 debe empezar en 180° por que al iniciar y terminar el ciclo, conviene que estos queden boca-abajo para reducir el número de instrucciones.
- Lavar y llenar de lodo: Si en el proceso VIII se realiza antes del proceso VII, es decir que pregunta si lavar antes de girar vaso en la posición 3, el vaso lavado estará más tiempo boca-abajo por lo que en el proceso V se espera que tenga menos humedad por el lavado.
- Giro del vaso en posición 3: De la posición 2 a la 3, el vaso se mantiene boca-abajo; mientras que de la posición 3 a la 1 el vaso se mantiene boca arriba. De acuerdo con lo anterior, hacer un giro por iteración en la posición 3 conviene para que en el proceso V se reduzca una instrucción al no haber vaso. Esto siempre que el motor del giro 2 sea unidireccional y solo sea para orientar al vaso.

Requerimientos

- Estado inicial: Se requieren encoders para asegurar las posiciones, así como guardarlas en la EEPROM del microcontrolador para no perder las referencias al apagar la máquina.
- Recuperación de iteraciones: En caso de apagado no planeado, se requiere regresar a las iteraciones faltantes o bien, regresar al estado 0. Para ello es necesario asegurar el inicio en posición 0 o recuperarlo.
- Programación: Secuencial discreta por eventos. Se requiere uso de clases y funciones, con variables globales para mantener los estados en toda la operación.

Cambios en los estados del proceso y modificación secuencial

La propuesta consiste en dividir a todo el proceso en tres bloques, con base en las iteraciones, debido al orden secuencial para no perder banderas al incluir todo el proceso como un bloque; se propone que la primera y la última iteración sean específicas. Así, el bloque principal es aquel en el cual se lleva a cabo la secuencia sin interrupciones por condónales, III, debido a que no tiene condicionales intermedias en su ejecución y solo se toma en cuenta el avance de las iteraciones máquina.

Cada bloque consta de sub-funciones “básicas”, que son las que cumplen una función determinada en todo el proceso y pueden ser llamadas cuando se les requiera sin afectar a la secuencia, tomando en cuenta banderas del proceso. Al poder ser integradas en bloques funcionales, se puede generar “supe funciones” que cumplen tareas más elaboradas.

En cada una de las secuencias básicas, se guardará el valor de las banderas en EEPROM, por si se apaga de forma no planeada. Si se apaga sin estar en menú, al cargar los datos de memoria, al inicio dará mensaje al usuario del apagado y dependiendo si termino alguna secuencia se podrá reanudar o poner en 0 la máquina.

Descripción de análisis de funciones y estados

Bloque	Función básica	Descripción	Banderas	Banderas con otra función
0	Datos usuario	Antes de iniciar secuencia se piden datos para la misma, en el menú	Ya datos Nuevos datos	
I	Cargar datos usuario	Se cargaran los datos que ingrese el usuario		Ya datos
	Estados en 0	Antes de iniciar la secuencia, se verificara que la maquina se encuentre en posición 0, si no, lo llevara a 0. Se requiere que antes sea calibrado de forma manual así como en caso de apagado no planeado se lleve a 0 antes de desplegar menú	Ya cero Ya secuencia final	Carrusel en 0 vaso1, boca-abajo vaso2, horizontal Aspas en alto
II	Secuencia 0	Es la primera iteración sin lavar. Al final, la posición uno queda con vaso boca-abajo, sucio; en la posición dos el vaso está limpio; en la posición tres queda el vaso boca-arriba.	Ya secuencia 0 Iteración máquina: 0	Ya cero
	Llenado lodo	Se activa actuador para verter lodo en la posición 3 siempre que el vaso este boca-arriba		Carrusel en 0
	Gira carrusel 30°	El carrusel se mueve 30° para que el motor del vaso 1 pueda ponerse en posición de recibir vaso. Cada que inicie esta función debe verificarse que el solenoide está activado, para que pueda moverse, si no, se encenderá.	Ya 30° Carrusel estático	Vaso 1, boca-abajo Solenoide encendido Vaso2, horizontal Carrusel en 0
	Giro vaso 1, boca-arriba	Se comprobara que el carrusel este estático en posición +30°, Se apaga el solenoide y se gira en posición para recibir vaso, boca-arriba.	Vaso1, boca-arriba	Carrusel estático Ya 30° Aspas en alto
	Giro carrusel 90°	El carrusel girara 90° si el vaso 1 esta boca-arriba, y si el carrusel ha girado antes 30°. Se posiciona para que los giros de vaso puedan acomodar los vasos en su posición.	Ya 90° Carrusel estático	Ya 30° Vaso 1, boca-arriba Vaso 2, horizontal
	Hacer mezcla	Función que revuelve el polímero con el lodo. Debe estar el carrusel en posición 0, y el vaso 1 este boca-arriba. Primero se bajan las aspas; luego se revuelve el lodo; se ingresa el polímero y las revoluciones se ponen en las deseadas; se mantienen las revoluciones por el tiempo de usuario; finalmente se levantan las aspas.	Aspas en alto	Carrusel en 0 Vaso 1, boca-arriba Carrusel estático
	Giro vaso 1, boca-abajo	El vaso en la posición 1 debe girar en posición boca-abajo, para verter el lodo con el polímero. Las aspas deben estar arriba y el carrusel estático. El solenoide se activara mientras gira y permanecerá hasta que	Vaso 1, boca-abajo	Carrusel en 0 Solenoide activado Aspas en alto

		vuelva a girar el vaso 1, boca-arriba.		
	Giro vaso 2, +180°	El vaso en la posición 3 solo girara cuando el carrusel gire a 90°, momento en el cual el vaso está en esa posición boca-abajo. Es la única vez que se requiere que se gire el vaso.	Vaso 2, horizontal	Carrusel estático Carrusel en 0
III	Secuencia 1	En este bloque se encuentra un ciclo controlado por iteraciones, hasta que se realice la n-1. Al finalizar, el vaso en la posición dos queda limpio; en la posición uno el vaso queda sucio luego de verter el lodo; en la posición tres el vaso queda boca-arriba, limpio. En este momento ya se han realizado todas las iteraciones	Ya secuencia 1 Iteración máquina: n-1	Ya secuencia 0 Iteración máquina: 0
	Llenar lodo Giro carrusel 30° Giro vaso 1, boca-arriba Giro carrusel 90°			
	Lavar	Se activa el actuador de lavar una vez que el carrusel gire a 90° y antes de que se realice la mezcla, con esto se tiene más tiempo para que el vaso en la posición dos quede más tiempo boca-abajo y pueda reducir la humedad por el lavado.		Carrusel estático Carrusel en 0
	Hacer mezcla Giro vaso1, boca-abajo Giro vaso 2, 180°			
III	Secuencia final	Del bloque II, se termina con un vaso sucio en la posición uno. En esta secuencia no se vertía lodo ni se usara la función de mezclar. Se busca lavar ese último vaso y dejar todo en posición 0.	Ya secuencia final	Ya secuencia 1
	Giro carrusel 30° Giro vaso 1, boca-arriba Giro carrusel 90° Lavar Giro vaso1, boca-abajo Giro vaso 2, 180°			

Propuesta de sistema modular, enfoque Mecatrónico

El proyecto lodos pretende automatizar un proceso complejo mediante un sistema secuencial, esto trae desventajas durante la ejecución por la limitación en la flexibilidad de configuración y planeación e operaciones, no puede permitir una adecuada planeación y es propenso a errores por estados. El planteamiento del proyecto se describe en la siguiente tabla:

Planteamiento del Proyecto Lodos	
Descripción de proyecto	Máquina para simular mezclado de floculante y efluente en planta
Propuesta de valor	- Sistema automatizado - Control y monitoreo de datos
Metas clave	- Flexibilidad del proceso - Registro de pruebas

	• Robusto
Mercado primario	• Proveedores de floculantes • Empresas con plantas de tratamiento
Mercado secundario	• Laboratorios
Suposiciones	• Automático • Electrónico • Interfaz usuario máquina • Innovador
Involucrados	• Operador • Departamento de ingeniería • Plantas de tratamiento

Las necesidades que se plantearon para el desarrollo de la electrónica de control son:

- A. Adaptable
 - a. Sistema que permita cambiar orden del proceso.
 - b. Modificación fácil del proceso (programación).
 - c. Tareas simultáneas para reducción de tiempo.
- B. Autónomo
 - a. Detección y solución de errores.
 - b. Repetitividad.
 - c. Configurable.
- C. Dispositivo comunicación
 - a. Control en parámetros.
 - b. Registro de actividades.
 - c. Informe de eventos.
 - d. Configuración de pruebas.
- D. Mezcla controlada
 - a. Exactitud de dosificación.
 - b. Limpieza de contenedores.
 - c. Limpieza de medios de dosificación.

Una alternativa para asegurar las necesidades y las especificaciones que surjan para el sistema de control es enfocar el desarrollo a un sistema modular.

Mientras en la automatización clásica es un solo control centralizado el que supervisa la máquina o instalación y procesa todas las señales, el sistema de automatización modular hace posible la distribución lógica de las funciones de control. A través de protocolos de comunicación se intercambian y sincronizan datos de proceso o control, información de diagnóstico y de proceso. La función de control no depende de la ubicación en que se ejecuta la sección de programa correspondiente.

En lugar de un control centralizado, el usuario dispone de un programa de aplicación distribuido en la fase de ejecución en el marco de un proyecto centralizado. A través del proyecto centralizado se

configuran, programan y diagnostican todos módulos de la red. El resultado es un manejo sencillo y unificado dentro del proyecto global.

Ventajas de un sistema modular vs secuencial

1. Fusión de seguridad y automatización

Para facilitar el intercambio de comunicación se utiliza un entorno para seguridad y automatización caracterizado por la fusión inteligente de hardware y software. El sistema es físicamente mixto, pero tiene la parte lógica separada y está libre de reacciones. La configuración del protocolo de la red de comunicación asegura una transmisión estable a través de la red y llegan con seguridad al destinatario previsto.

2. Menos volumen de ingeniería, tiempos de ejecución de proyectos más cortos

En muchos sistemas de automatización es preciso seleccionar el hardware antes de poder realizar la configuración/programación. Cualquier cambio posterior resulta muy laborioso. Un sistema modular es diferente:

- La elección del hardware y la distribución del programa en el hardware puede realizarse muy tarde al ser un paso generalmente independiente de la configuración.
- Tiempos de ejecución de proyectos más cortos mediante procesamiento paralelo de tareas parciales: posibilidad de seleccionar el hardware y la distribución del programa a nivel de hardware en una fase mucho más avanzada
- Ampliaciones posteriores de una máquina: el programa de aplicación puede distribuirse sin mayores complicaciones en controles adicionales
- Puestas en marcha parciales y funcionamiento individual de partes de máquinas.

3. Sistema abierto para alto grado de flexibilidad

El sistema de automatización modular es un sistema abierto que se integra fácilmente en arquitecturas de automatización existentes y en diferentes controles de otros fabricantes.

4. Facilidad de programación y configuración

La plataforma de software engloba varias metodologías y numerosas estructuras de módulos en software y hardware, por lo que las herramientas de configuración de proyectos, programación, puesta en marcha están estrechamente ligadas. Los interfaces de datos están unificados y simplifican el intercambio de información en todas las fases de la automatización. La creación de programas para funciones de seguridad y de automatización es rápida e intuitiva.

5. Diagnóstico y monitoreo

Los tiempos de parada de la maquinaria y las interminables búsquedas de errores son cosa del pasado gracias a las posibilidades de diagnóstico en las líneas de comunicación. Además del diagnóstico del sistema que ejecuta el mismo usuario puede configurar un diagnóstico específico del proceso. Diferentes medidas permiten detectar errores de forma rápida y eficiente:

- Mensajes de texto claro detallados con ubicación o indicador de equipo para cada evento

- Soluciones ‘paso a paso’ detalladas
- Priorización de eventos y definición de competencias
- Posibilidad de adaptar fácilmente mensajes predeterminados.
- Los textos de diagnóstico pueden leerse cómodamente en diferentes tipos de visualizadores, como el dispositivo de control y visualización.

Requerimientos para un sistema modular

Un sistema modular requiere definir dos puntos clave: un bus de comunicación y un tipo de controlador. Existen numerosos métodos, instrumentos, protocolos y dispositivos que pueden adaptarse a un sistema modular. El proyecto está enfocado en sistemas de control por microcontroladores, por su bajo costo, pero requieren sistemas de acondicionamiento. La diferencia con un sistema secuencial se encuentra en la estructura de comunicación, lo que permite tener todas las ventajas mencionadas anteriormente.

A partir de las necesidades del bus se puede seleccionar el dispositivo de comunicación, en cuanto a microcontroladores, por lo que la definición de los elementos mínimos para asegurar un sistema modular son:

- Funciones de solicitud de envío y recepción de datos
- Estructura de datos
- Detección de estados de operaciones de tarea y error
- Detección de solicitudes de lectura

Las capacidades del microcontrolador seleccionado deben satisfacer al menos estos puntos para garantizar que la ejecución y configuración de las operaciones del proceso sean flexibles y no se limiten por la cantidad de operaciones.

C. Interfaz de usuario, Proyecto Lodos

Antonio Fonseca Alvarado

SNF Floerger de México, proyecto LODOS, mayo 2014

Introducción

El proyecto Lodos requiere una interfaz intuitiva para el usuario que le permita configurar, monitorear y probar los sistemas mecánicos. Para poder adaptar el sistema a las funciones de operación de configuración de la unidad de control, se desarrolla un sistema general que se adapte y cumpla con los requerimientos de ser intuitivo para usuario y permita flexibilidad de programación, esto mediante el establecimiento de una plantilla genérica que prepare el uso de la interfaz.

Especificaciones y limitaciones

La interfaz gráfica debe cumplir con las siguientes funciones

- Menús desplegables
- Información de parámetros
- Solicitud de confirmación
- Notificación de acciones
- Notificación de eventos
- Notificación de advertencias

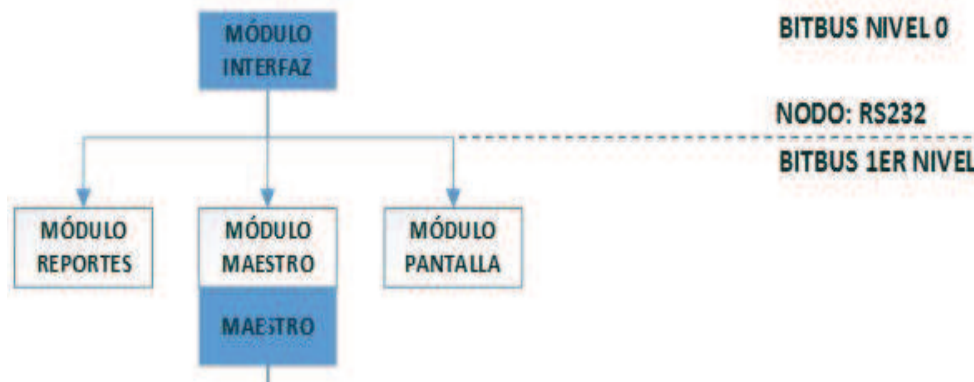
No se debe confundir estas funciones con las funciones de operaciones de configuración, estas son analizadas en el capítulo 4.1.4 y 4.2.2, por lo que debe entenderse por funciones principales a las que gestionen el acondicionamiento de la interfaz.

Los componentes electrónicos para el desarrollo del sistema son una pantalla resistiva gráfica, se ha propuesto agregar un elemento esclavo para dedicar la función *data logger* para seguir la estructura modular Bitbus. Estos componentes se describen en la siguiente tabla.

Componente	Característica	Comunicación	Observaciones
Pantalla resistiva	Pantalla con touch resistivo 3 pulgadas, grafica	Configuración por RS232	Es una versión 1 descontinuada, es funcional pero lenta y requiere dedicar un micro para que sea estable
Arduino mega	16 [MHz] a 5 [V] 70 pines de propósito general	RS232, 4 USART dedicados, I2C y SPI	Dedicado para el uso de pantalla
Reloj en tiempo real	Oscilador modulo bajo costo basado en DS1307	Configuración por I2C	Requiere una batería extra para su integrado
Adaptador memoria SD	Adaptador de pines de memoria SD a SPI	Configuración por SPI	Se debe usar una memoria de 2Gb o menos
Arduino uno	16 [MHz] a 5 [V] 19 pines de propósito general	RS232, I2C, y SPI	Dedicado para el reloj y la memoria

Estructura

Al seguir un modelo Bitbus, la interfaz cuenta con tres módulos esclavos, uno de ellos el maestro de los módulos del sistema físico, los otros dos son la pantalla y el generador de reportes. En la figura se observa su nivel de Bitbus y el tipo de comunicación. El Arduino mega funge como maestro en este nivel. La estructura de los componentes de la interfaz sigue la planteada en el diseño conceptual, el maestro únicamente recibe datos de usuario y es quien gestiona la configuración del sistema, mientras que los esclavos actúan en función de las configuraciones.



La definición para los módulos a de interfaz y de reportes se resuelve de forma particular al ser elementos comerciales y usar librerías para su configuración, a partir de estos se definen funciones genéricas para el acondicionamiento de sus operaciones, se hace a detalle con el fin de que la implementación en otro tipo de dispositivo sea relativamente fácil, o se use como referencia para el mejoramiento de la interfaz

Pantalla

La interfaz de usuario se especifica el desarrollo de los menús a partir de los módulos del sistema mecánico; estos menús tienen como objetivo ordenar operaciones en grupos para hacerle intuitivo al usuario el uso de la unidad de control.

Del fabricante, se tiene dispuesto un paquete de librerías para el uso de la pantalla, se trata del modelo SMARTGPU v1, de VIZIC TECHNOLOGIES. Cuenta con más de 30 funciones para gráficos, letras y gestión del touch; cabe aclarar que este modelo ha sido descontinuado y en el fabricante ha dejado de dar soporte. Las funciones principales que se toman son:

- `uint8_t erase()`: Erase the SMARTGPU screen
- `uint8_t orientation(uint8_t side)`: Change display orientation
- `uint8_t setScreenBackground(int color)`: Change screen background color for erase function
- `uint8_t setTextBackground(int color)`: Set text background color for letters and strings
- `uint8_t drawLine(int x1, int y1, int x2, int y2, int color)`: Draw a line on the screen
- `uint8_t drawRectangle(int x1, int y1, int x2, int y2, int color, uint8_t fill)`: Draw a rectangle `uint8_t`
- `putLetter(int x, int y, int color, uint8_t font, uint8_t fill, uint8_t letter)`: Draw a letter on the screen
- `uint8_t string(int x1, int y1, int x2, int y2, int color, uint8_t font, uint8_t fill, char text[])`: Draw a string
- `uint8_t touchScreen(int buffer[])`: Ask for a touch on the screen, if `return=1`, touch coordinates are stored on the `buffer[]`

Se deja indicado el tipo de función y parámetros que debe recibir para asociarlas a las funciones que se desarrollarán; las funciones de interfaz deben ser de forma genérica para que la programación reduzca la cantidad de código. De esta forma se busca que la implementación en otro tipo de interfaz no depende de las funciones particulares si no que se pueda reusar el tipo de función.

Las funciones se plantean para describir las operaciones básicas de la interfaz, los despliegues de información se desarrollan en las siguientes 7 funciones básicas, se escribe el código a manera de especificar el tipo de operación para que pueda implementado en otra plataforma. Estas funciones son propuestas y se aceptaron por parte del equipo de trabajo

MENSAJE NORMAL, sin espera

(TEXTO MOSTRAR, VALOR A MOSTRAR CON REFERENCIA AL TEXTO [OPCIONAL])

Esta función recibe texto y si se requiere, muestra variable, no modifica ninguna bandera o variable; limpia pantalla al terminar. Auxiliar en recibir número de usuario o para indicar alguna acción en ejecución.

```
void mensajeNormalT(char texto[], long valor){

    int temp=0;                                //bandera para verificar un valor introducido, no 0
    char valorcito[15];                        //cadena de caracteres, para conversión si hay valor

    lcd.erase();                               //limpiamos pantalla
    lcd.drawRect(10,100,230,250,RED,UNFILL);  //rectángulo para mensaje
    lcd.string(10,80,230,95,RED,FONT4,TRANS,"ACCION"); //Escribe de que trata mensaje
    lcd.string(10,120,230,250,RED,FONT3,TRANS,texto); //se escribe el mensaje

    if(valor!=temp){                            //si valor no es nulo
        ltoa(valor,valorcito,10);               //conversión de entero a char
        lcd.string(80,200,230,300,BLUE,FONT5,TRANS,valorcito);
    } //escribimos el valor con formato

} //fin void mensaje normal
```



MENSAJE NORMAL, con espera de toque

(TEXTO MOSTRAR, VALOR A MOSTRAR CON REFERENCIA AL TEXTO [OPCIONAL])

Esta función recibe texto y si se requiere, muestra variable, no modifica ninguna bandera o variable limpia pantalla al terminar. Auxiliar para desplegar información al usuario, como instrucciones o avisos antes de realizar operaciones.

```
void mensajeNormal(char texto[], long valor){
    int temp=0;                //bandera para verificar un valor introducido, no 0
    char valorcito[15];        //cadena de caracteres, para conversión si hay valor

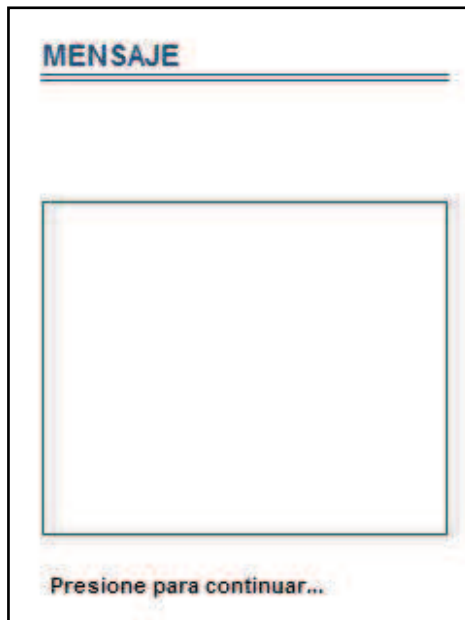
    lcd.erase();               //limpiamos pantalla
    lcd.string(20,20,230,20,BLUE,Font4,TRANS,"MENSAJE"); //Escribe de que trata mensaje
    lcd.drawLine(17,37,220,37,BLUE); //línea
    lcd.drawLine(17,42,220,42,BLUE); //línea

    lcd.drawRect(10,100,230,290,BLUE,UNFILL); //rectángulo para mensaje
    lcd.string(20,110,230,280,BLUE,Font3,TRANS,texto); //se escribe el mensaje

    if(valor!=temp){           //si valor no es nulo
        ltoa(valor,valorcito,10); //conversión de entero a char
        lcd.string(80,230,230,300,BLUE,Font5,TRANS,valorcito);
    } //escribimos el valor con formato char

    //Para la salida por espera de usuario
    delay(1000);
    lcd.string(10,300,230,310,BLACK,Font3,TRANS,"Presiona para continuar..");
    while(!lcd.touchScreen(touch)); //espero que toque pantalla

    //Indicación grafica de que se ha presionado
    lcd.string(10,300,230,310,WHITE,Font3,COLOUR,"Presiona para continuar..");
    delay(400);
    lcd.erase();               //limpio pantalla
} //fin void mensaje normal
```



MENSAJE ERROR

(TEXTO MOSTRAR, VALOR A MOSTRAR CON REFERENCIA AL TEXTO [OPCIONAL])

Esta función recibe texto y si se requiere, muestra variable, no modifica ninguna bandera o variable; limpia pantalla al terminar, a diferencia de mensaje normal, pone fondo rojo para llamar la atención del usuario. Auxiliara para cuando se requiere alguna acción para corregir el error de forma manual por programador

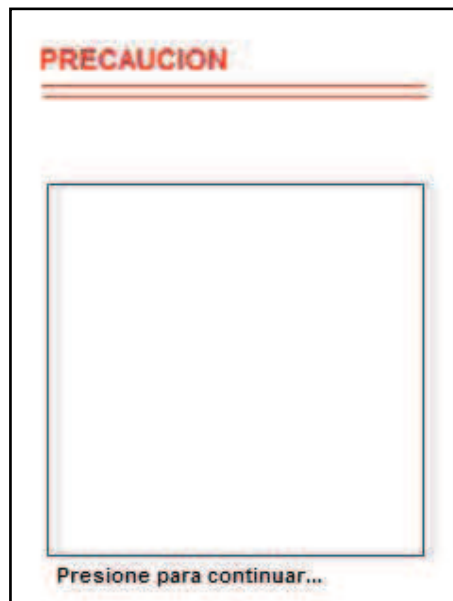
```
void mensajeError(char texto[], int valor){
  lcd.setScreenBackground(WHITE);      //cambia fondo a rojo
  int temp=0;                          // para ver si hay entrada de valor entero
  char valorcito[15];                  //cadena char para conversión del valor de entrada

  lcd.erase();                        //limpia pantalla
  lcd.drawRect(10,100,230,290,BLUE,UNFILL); //rectángulo para mensaje
  lcd.string(20,20,230,20,RED,FONT4,TRANS,"PRECAUCION"); //Escribe de que trata mensaje
  lcd.drawLine(17,37,220,37,RED);     //línea
  lcd.drawLine(17,42,220,42,RED);     //línea

  lcd.string(20,110,225,280,BLUE,FONT3,TRANS,texto); //Se escribe el mensaje de error
  if(valor!=temp){                    //Si hay valor entero de entrada
    itoa(valor,valorcito,10);         //conversión de entero a char
    lcd.string(80,200,230,300,BLUE,FONT5,TRANS,valorcito);
  } //muestra el valor que se cambió a char

  delay(300);
  lcd.string(10,300,220,280,BLACK,FONT3,TRANS,"Presiona para continuar..");
  while(!lcd.touchScreen(touch));     //espera a que se toque
  lcd.string(10,300,220,280,WHITE,FONT3,COLOUR,"Presiona para continuar..");

  delay(300);
  lcd.setScreenBackground(WHITE);     //regresamos al color default
  lcd.erase();                        //limpio pantalla
} //fin void mensaje error
```



Entrada de datos

Para esta función se requiere definir 3 auxiliares para hacerlo de forma genérica:

- Dibujo de teclado, a partir de un formato previo, se dibuja de forma gráfica un teclado
- Lectura en teclado, el usuario debe seleccionar algún número visual, con esta función se determina si se ha tocado y regresa el valor asociado
- Obtención de número de 4 dígitos

DIBUJAR TECLADO

Esta función no entrega valor, exhibe un texto para indicar a qué tipo de dato se asocia el valor de entrada. Genera un teclado de 0 al 9 con dos botones OK y BORRAR. Auxiliar para: número de 4 dígitos por usuario.

```
void Dibujar_teclado(char AQUE[]){
    lcd.erase();                                //Limpio pantalla
    lcd.drawRect(10,10,230,110,BLUE,UNFILL);    //draw a rectangle for message
    lcd.string(10,20,230,105,BLUE,FONT3,TRANS,"INGRESA UN NUMERO:"); //pongo mensaje de numero
    lcd.string(10,95,230,110,BLACK,FONT3,TRANS,AQUE); //opciones del teclado

    lcd.drawRect(12,127,228,313,BLUE,UNFILL);    //draw a rectangle for message
    lcd.drawRect(15,130,225,310,BLUE,UNFILL);    //draw a rectangle for teclado
    lcd.drawRect(16,131,224,309,CYAN,FILL);      //draw a rectangle for teclado

    //horizontales y verticales
    lcd.drawLine(15,175,225,175,BLUE);           //draw a line
    lcd.drawLine(15,220,225,220,BLUE);           //draw a line
    lcd.drawLine(15,265,225,265,BLUE);           //draw a line
    lcd.drawLine(85,130,85,310,BLUE);            //draw a line
    lcd.drawLine(155,130,155,310,BLUE);          //draw a line

    //Letras del teclado
    lcd.putLetter(45,150,BLACK,FONT4,TRANS,'7'); //write a single letter '7'
    lcd.putLetter(45,195,BLACK,FONT4,TRANS,'4'); //write a single letter '4'
    lcd.putLetter(45,240,BLACK,FONT4,TRANS,'1'); //write a single letter '1'
    lcd.string(30,285,70,300,RED,FONT3,TRANS,"BORRAR");
    lcd.putLetter(105,150,BLACK,FONT4,TRANS,'8'); //write a single letter '8'
    lcd.putLetter(105,195,BLACK,FONT4,TRANS,'5'); //write a single letter '5'
    lcd.putLetter(105,240,BLACK,FONT4,TRANS,'2'); //write a single letter '2'
    lcd.putLetter(105,285,BLACK,FONT4,TRANS,'0'); //write a single letter '0'
    lcd.putLetter(175,150,BLACK,FONT4,TRANS,'9'); //write a single letter '9'
    lcd.putLetter(175,195,BLACK,FONT4,TRANS,'6'); //write a single letter '6'
    lcd.putLetter(175,240,BLACK,FONT4,TRANS,'3'); //write a single letter '3'
    lcd.string(175,285,220,300,RED,FONT4,TRANS,"OK");
} //fin void dibujar teclado
```

TECLADO

Esta función no recibe valor. Cambia el valor de variable global **“button”** para el valor recibido por pantalla, hasta que sea uno del teclado. Auxiliar para: número de 4 dígitos por usuario

```
void Teclado(){
    int X1, Y1;                                //variables locales para pantalla
    button = 12;                               //bandera global para valor de botón
```



```

while(button==12){           //hasta que sea una tecla
  while(!lcd.touchScreen(touch)); //espera a que sea tocada la pantalla
  X1=touch[XCOORD];
  Y1=touch[YCOORD];         //coordenadas del toque

  if(X1>=156 && X1<=224){    //toque en última columna derecha
    if(Y1>=131 && Y1<=174)button=9; //toque en primera fila
    if(Y1>=176 && Y1<=219)button=6; //toque en segunda fila
    if(Y1>=219 && Y1<=264)button=3; //toque en 3ra fila
    if(Y1>=266 && Y1<=309)button=11;
  } //toque en 4ta fila, B

  else if(X1>=86 && X1<=154){ //toque columna en medio de en medio
    if(Y1>=131 && Y1<=174)button=8; //toque en primera fila
    if(Y1>=176 && Y1<=219)button=5; //toque en segunda fila
    if(Y1>=219 && Y1<=264)button=2; //toque en 3ra fila
    if(Y1>=266 && Y1<=309)button=0;
  } //toque en 4ta fila

  else if(X1>=16 && X1<=84){ //toque en primera columna izquierda
    if(Y1>=131 && Y1<=174)button=7; //Toque en primera fila
    if(Y1>=176 && Y1<=219)button=4; //toque en 2da fila
    if(Y1>=219 && Y1<=264)button=1; //toque en 3ra fila
    if(Y1>=266 && Y1<=309)button=10;
  } //toque en 4ta fila, A

} //fin while asegura un botón

} //FIN TECLADO

```

NUMERO 4 DIGITOS

(CHAR NOMBRE DE VARIABLE)

Esta función recibe un valor tipo char, y regresa un entero. El valor de char es para introducir el nombre a la variable a la que se le dará el entero de salida. Solo admite entrada de 4 dígitos, si es mayor repite la entrada.

```

long numero_4dig(char AQUE[],char NomVar[]){

  ya_entrada_num=0;           //bandera global, de ya se obtuvo 4 dig
  temp_1=0;temp_2=0;temp_3=0;temp_4=0;unidades=0;
  contador_num=0;             //bandera global, contador de unidades
  numero_entrada=0;           //bandera global, entrada por teclado
  Dibujar_teclado(AQUE);      //función, dibujar teclado 4*3

  while(ya_entrada_num==0){    //espera cambio de bandera
    Teclado();                 //llama a teclado, regresa un cambio de variable, button

    if(button!=11 && button!=10){ //revisa si button es diferente de A(10) o B(11)
      contador_num++;           //incremento contador

      if(contador_num==1){      //unidades

```

```

temp_1=button;
unidades=1;
numero_entrada=button;
} //numero_entrada es valor del botón

else if(contador_num==2){           //decenas
temp_2=button;
unidades=2;
numero_entrada=numero_entrada*10;      //numero_entrada es 10 veces mas [decenas]
numero_entrada=numero_entrada+button;
} //numero_entrada se suma con unidad de botón

else if(contador_num==3){           //centenas
temp_3=button;
unidades=3;
numero_entrada=numero_entrada*10;      //numero_entrada es 10 veces mas [centenas]
numero_entrada=numero_entrada+button;
} //numero_entrada se suma con unidad de botón

else if(contador_num==4){           //miles
temp_4=button;
unidades=4;
numero_entrada=numero_entrada*10;      //numero_entrada es 10 veces mas [miles]
numero_entrada=numero_entrada+button;
} //numero_entrada se suma con unidad de botón

else if(contador_num==5){           //entrada muy grande
mensajeError("Numero muy grande, ingresa nuevo numero. Ultimo digito: ",numero_entrada);
ya_entrada_num=0;                    //reseteo de variable
contador_num=0;                      //reseteo de bandera
numero_entrada=0;                    //reseteo de bandera
Dibujar_teclado(AQUE);               //dibujo de nuevo el teclado, incluye limpiar
}
mensajeNumero(numero_entrada); //imprimo el valor de numero entrada
} //fin no fue botón OK(10) ni BORRAR(11)

else if(button==10){                //si es BORRAR, me equivoque, reinicia valores
ya_entrada_num=0;                    //reseteo de variable
contador_num=0;                      //reseteo de bandera
numero_entrada=0;                    //reseteo de bandera
Dibujar_teclado(AQUE);               //dibujo de nuevo el teclado, incluye limpia
} //fin if si presiono A[10]

else ya_entrada_num=1;               //fue OK, termina entrada datos, se cambia bandera

} //fin while de pedir numero

mensajeNormal(NomVar,numero_entrada); //escribo nombre variable y valor que tendrá
return numero_entrada;                //regresa el valor de numero entrada

} //fin función int numero4 dígitos

```

Los despliegues de pantalla a partir de una confirmación de entrada son los siguientes:



DIBUJAR MENU 5

(NOMBRE MENU, OPCIÓN 1, OPCIÓN 2, OPCIÓN 3, OPCIÓN 4, OPCIÓN 5)

Esta función entrega valor ideal para usar un siwtch con bandera asignada.

```
int Dibujar_Menu5(char MENU[],char op1[],char op2[],char op3[],char op4[],char op5[]){
```

```
    int X1, Y1;                //variables locales para pantalla
    char menucito[1];          //variable char para meter opción menú
    button =12;                //bandera global para valor de botón

    lcd.erase();               //Limpio pantalla
    lcd.drawRect(10,10,230,70,BLUE,UNFILL);    //draw a rectangle for messege
    lcd.drawRect(155,85,225,310,BLUE,UNFILL);  //draw a rectangle for menu

    //horizontales
    lcd.drawLine(15,130,225,130,BLUE);          //draw a line
    lcd.drawLine(15,175,225,175,BLUE);          //draw a line
    lcd.drawLine(15,220,225,220,BLUE);          //draw a line
    lcd.drawLine(15,265,225,265,BLUE);          //draw a line
    lcd.drawLine(15,310,225,310,BLUE);          //draw a line

    //Letras del teclado
    lcd.putLetter(175,105,BLACK,FONT4,TRANS,'1'); //write a single letter '9'
    lcd.putLetter(175,150,BLACK,FONT4,TRANS,'2'); //write a single letter '9'
    lcd.putLetter(175,195,BLACK,FONT4,TRANS,'3'); //write a single letter '6'
    lcd.putLetter(175,240,BLACK,FONT4,TRANS,'4'); //write a single letter '3'
    lcd.putLetter(175,285,BLACK,FONT4,TRANS,'5'); //write a single letter 'B'

    //Opciones del teclado
    lcd.string(15,105,145,105,BLACK,FONT3,TRANS,op1); //write a single letter '7'
    lcd.string(15,150,145,150,BLACK,FONT3,TRANS,op2); //write a single letter '7'
    lcd.string(15,195,145,195,BLACK,FONT3,TRANS,op3); //write a single letter '4'
    lcd.string(15,240,145,240,BLACK,FONT3,TRANS,op4); //write a single letter '1'
    lcd.string(15,285,145,285,BLUE,FONT3,TRANS,op5);  //write a single letter 'A'
```

```

lcd.string(10,15,230,25,BLUE,Font3,TRANS,"...: MENU :...");
lcd.string(70,30,230,40,BLUE,Font3,TRANS,menucito);
delay(200);

while(button==12){                                     //hasta q sea una tecla

    while(!lcd.touchScreen(touch));                   //espera a q sea tocada la pantalla
    X1=touch[XCOORD];
    Y1=touch[YCOORD];                                 //coordenadas del toque

    if(X1>=20 && X1<=224){                             //toque en última columna derecha
        if(Y1>=86 && Y1<=129)button= 1;                //toque en cero fila
        if(Y1>=131 && Y1<=174)button=2;                //toque en primera fila
        if(Y1>=176 && Y1<=219)button=3;                //toque en segunda fila
        if(Y1>=219 && Y1<=264)button=4;                //toque en 3ra fila
        if(Y1>=266 && Y1<=309)button=5;
    } //toque de opciones de menu
    else button=12;

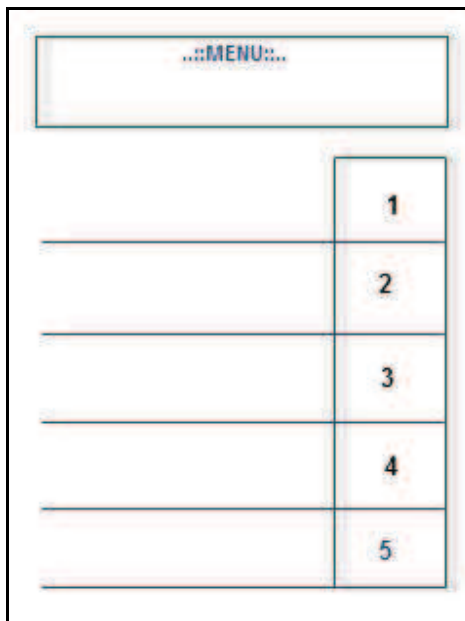
} //fin while asegura una opción del menucito

if(button==5){
    lcd.string(20,50,220,70,WHITE,Font4,COLOUR,op5);    //poniendo opción en menú arriba
    lcd.drawRect(155,265,225,310,BLACK,FILL);           //poniendo iconito en negro
    lcd.putLetter(175,285,WHITE,Font4,TRANS,'5');        //poniendo color
}
else if(button==4){
    lcd.string(20,50,220,70,WHITE,Font4,COLOUR,op4);
    lcd.drawRect(155,220,225,265,BLACK,FILL);
    lcd.putLetter(175,240,WHITE,Font4,TRANS,'4');
}
else if(button==3){
    lcd.string(20,50,220,70,WHITE,Font4,COLOUR,op3);
    lcd.drawRect(155,175,225,220,BLACK,FILL);
    lcd.putLetter(175,195,WHITE,Font4,TRANS,'3');
}
else if(button==2){
    lcd.string(20,50,220,70,WHITE,Font4,COLOUR,op2);
    lcd.drawRect(155,130,225,175,BLACK,FILL);
    lcd.putLetter(175,150,WHITE,Font4,TRANS,'2');
}
else if(button==1){
    lcd.string(20,50,220,70,WHITE,Font4,COLOUR,op1);
    lcd.drawRect(155,85,225,130,BLACK,FILL);
    lcd.putLetter(175,105,WHITE,Font4,TRANS,'1');
}

delay(1000);
return button;

} //FIN DIBUJAR MENU de 5 Opciones

```



MENSAJE DE 5 NUMEROS

(CUANTOS MOSTRAR, N1, N2, N3, N4, N5)

Función que recibe número de elementos a imprimir y los que imprimirá, no cambia valores globales. Requiere llamar antes la función clear

```
void mensaje5Numeros(int cuantos, long numerito1, long numerito2, long numerito3, long numerito4, long numerito5){
```

```
    char valorsito1[5], valorsito2[5], valorsito3[5], valorsito4[5], valorsito5[5]; //variable locales
```

```
    lcd.drawRectangle(10,10,230,230,RED,UNFILL);
    lcd.string(10,20,230,20,RED,FONT3,TRANS," [#]   VALOR   REFERENCIA");
    lcd.drawLine(20,35,220,35,RED);
    lcd.drawLine(20,40,220,40,RED);
```

```
    if(cuantos>=1){
        ltoa(numerito1, valorsito1, 10);
        lcd.string(10,80,230,50,BLUE,FONT4,TRANS,"[1]: ");
        lcd.string(80,80,230,50,BLUE,FONT4,TRANS,valorsito1);
    } //imprime valor convertido a char
    if(cuantos>=2){
        ltoa(numerito2, valorsito2, 10);
        lcd.string(10,110,230,80,BLUE,FONT4,TRANS,"[2]: ");
        lcd.string(80,110,230,80,BLUE,FONT4,TRANS,valorsito2);
    }
    if(cuantos>=3){
        ltoa(numerito3, valorsito3, 10);
        lcd.string(10,140,230,110,BLUE,FONT4,TRANS,"[3]: ");
        lcd.string(80,140,230,110,BLUE,FONT4,TRANS,valorsito3);
    }
    if(cuantos>=4){
        ltoa(numerito4, valorsito4, 10);
        lcd.string(10,170,230,140,BLUE,FONT4,TRANS,"[4]: ");
```



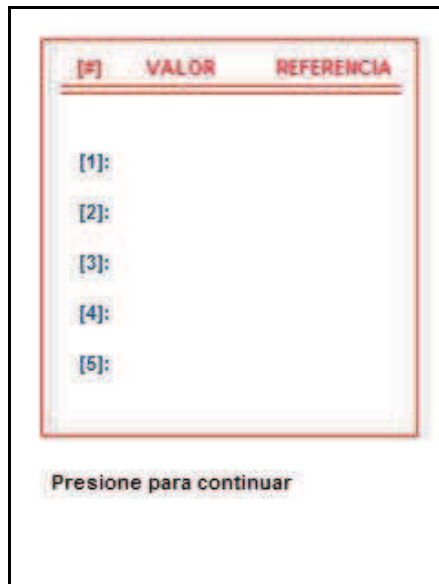
```

    lcd.string(80,170,230,140,BLUE,FONT4,TRANS,valorcito4);
}
if(cuantos>=5){
    ltoa(numerito5,valorcito5,10);
    lcd.string(10,200,230,170,BLUE,FONT4,TRANS,"[5]: ");
    lcd.string(80,200,230,170,BLUE,FONT4,TRANS,valorcito5);
}

lcd.string(10,250,230,280,BLACK,FONT3,TRANS,"Presione para continuar");
delay(1000);
while(!lcd.touchScreen(touch));

} //fin impresión de hasta 5 números

```



DECISIÓN SI/NO

(MENSAJE SOBRE A QUÉ VA, ELECCIÓN 1, ELECCIÓN 2)

Esta función recibe valores char y entrega un entero. Ideal para preguntar qué acción tomar.

```

int DesicionSiNo(char Que[],char About1[], char About2[]){
    int decision=0;                //Bandera local, decisión a tomar
    int X1,Y1;                     //bandera local, coordenadas de toque
    lcd.erase();                  //borro pantalla
    lcd.string(10,10,220,50,BLACK,FONT4,TRANS,"PREGUNTA");
    lcd.string(10,60,220,205,BLACK,FONT3,TRANS,Que);           //escribo sobre que será la decisión
    lcd.drawRectangle(15,195,115,230,RED,UNFILL);             //draw a rectangle para op1
    lcd.drawRectangle(125,195,220,230,RED,UNFILL);            //draw a rectangle para op2
    lcd.string(20,210,110,210,RED,FONT5,TRANS>About1);         //escribo que es la op1
    lcd.string(130,210,225,210,RED,FONT5,TRANS>About2);        //escribo que es la op2

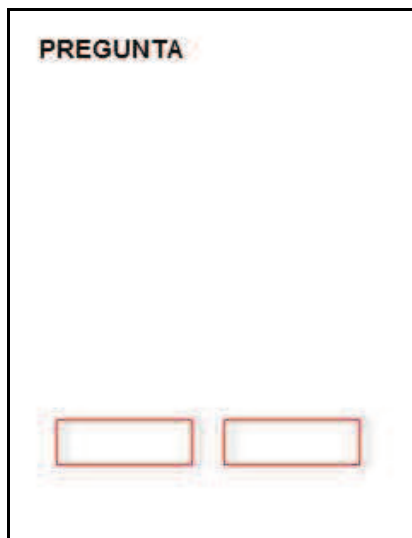
    while(decision==0){
        while(!lcd.touchScreen(touch));                         //espero un toque
        X1=touch[XCOORD];
        Y1=touch[YCOORD];
        if(X1>=15 && X1<=115){
            //hasta que escoja una opcion
            //guardo coordenadas de toque
            //opción 1, izquierda

```

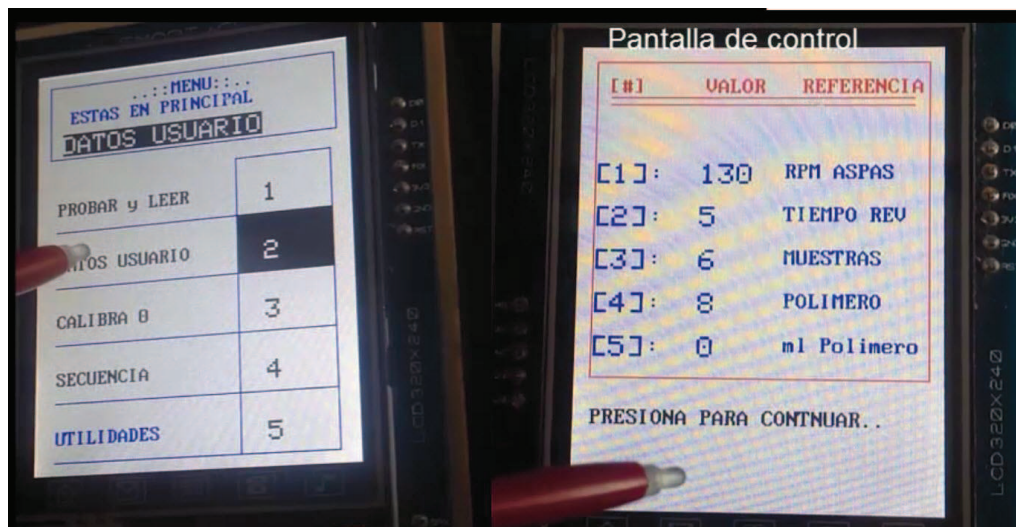
```

if(Y1>=195 && Y1<=230){
    decision=1;                                //cambio de bandera a 1
    lcd.string(15,256,220,300,BLACK,Font3,TRANS,About1);
}
} //escribo que opción se tomo
else if(X1>=125 && X1<=220){                  //opción 2, derecha
    if(Y1>=195 && Y1<=230){
        decision=2;                            //cambio bandera a 2
        lcd.string(15,256,220,300,WHITE,Font3,COLOUR,About2);
    }
} //escribo que opción tomo
else decision=0;                              //si no se tocó cuadros, no cambio bandera
} //fin obtención de decisión
return decision;                              //retorno el valor de opción seleccionado
} //fin mensaje si/no, 1 u 2

```



En el capítulo 5.3 se muestran las pruebas realizadas a la interfaz de usuario, donde se puede apreciar la ejecución de la interfaz y su correcto funcionamiento. En la siguiente imagen se muestra una muestra de las opciones en pantalla.



Reportes

La generación de reportes sigue una secuencia simple, cada que el maestro solicite una llamada de datos, se termine una tarea o algún esclavo del ultimo nivel del Bus genere un cambio, se llamara al esclavo para que guarde datos, no se espera respuesta de este , por lo que no afecta en opresiones del proceso. El principal motivo para generar este tipo de registros es para tener una referencia auxiliar en las pruebas, de tal forma que se identifique los parámetros usados. Se requiere dos tipos de archivos a guardar:

- Para el almacenamiento de todo tipo de eventos registrado
- Para el almacenamiento de encendidos de maquina

El uso de un reloj en tiempo real permite establecer referencias de fechas, mientras que el uso de la memoria SD permite guardar una gran cantidad de archivos de registros. Las funciones para el acondicionamiento del módulo RTC y para el uso de la memoria SD son genéricas y están en función de librerías., por lo que no se detalla en ellas, por otra parte, se describen dos funciones para el registro de datos y generación de archivos. Se plantean las siguientes 4 funciones específicas:

ESCRITURA DE ARREGLO DE CARACTERES EN SD

La función guarda en el archivo la entrada correspondiente, por el tipo de función, el archivo es abierto, al confirmar su existencia escribe en la última línea. Al asegurar esta función, la forma gráfica de mostrar los datos depende del programador.

```
void escribe_SD(char cosa[]){
  Archivo = SD.open(reporte,O_CREAT | O_APPEND | O_WRITE);
  if (Archivo!=0) {
    Archivo.println(cosa);
    Archivo.flush();
    Archivo.close();
  }
}
```

ESCRITURA DE FECHA EN SD

La función para generar la fecha permite tener referencia en tiempo de los eventos, recurre a una función **hoy()** que fue desarrollada a partir del modelo genérico de la librería, no se describe al no se nada específico, únicamente se guarda en un arreglo la fecha.

```
void escribe_Fecha(){
  hoy(); //función genérica para obtener fecha de RTC
  Archivo = SD.open(reporte,O_CREAT | O_APPEND | O_WRITE);
  if (Archivo!=0) {
    Archivo.println(" ");
    Archivo.print("Evento nuevo: ");Archivo.println(Fecha);
    Archivo.flush();
    Archivo.close();
  }
}
```

ARCHIVO PARA EL REGISTRO DE INICIO

La función permite registrar cada que se inicie o reinicie la máquina, esta es llamada por el módulo maestro cada que inicia operaciones. El nombre del archivo esta en función de la fecha, por lo que por cada mes se genera un reporte de inicio de sesión, los datos almacenados para este archivo son de estado general de SD, el nombre de archivo y la hora de sesión.

```
void SD_star(){
    //creacion de reporte de inicio
    Archivo = SD.open(sesion,O_CREAT | O_APPEND | O_WRITE);
    if(Archivo!=0){
        Archivo.println(" ");
        Archivo.println("===== Lodos Sys =====");
        Archivo.println("===== REPORTE LOGIN =====");
        Archivo.println(" ");
        Archivo.print("Fecha: ");Archivo.print(Fecha); Archivo.println(" ");
        Archivo.print(" Estado de errores : "); Archivo.println(Ee);
        Archivo.print(" Reportes en : ");Archivo.println(reporte);
        Archivo.print(" Inicio Sesion en: ");Archivo.println(sesion);
        Archivo.print(" ESTADO SD: ");Archivo.println(SD_ok);Archivo.println(" ");
        Archivo.flush();
        Archivo.close();
    }
} //fin
```

ARCHIVO PARA EL REGISTRO DE EVENTOS

La función genera un archivo para el guardado de los reportes por eventos, el formato del archivo es por día, por lo que se puede tener un mejor registro de todos los cambios en la configuración de la máquina, esto auxilia al operador para el mantenimiento, para detectar las operaciones con error y los cambios en los parámetros de las pruebas.

```
void SD_starR(){
    if(!SD.exists(reporte)){
        Archivo = SD.open(reporte,O_CREAT | O_APPEND | O_WRITE);
        if(Archivo!=0){
            Archivo.println("===== Lodos Sys =====");
            Archivo.println("===== NUEVO REPORTE GENERADO =====");
            Archivo.println(Fecha); Archivo.println(" ");
            Archivo.flush();
            Archivo.close();
        }
    } //si no existe nuevo reporte
}
```

Cabe mencionar que únicamente se ponen funciones para el guardado de datos en SD, se reitera que las demás funciones para la configuración del RTC o de la misma SD son genéricas por librería, y para la programación de las demás operaciones del módulo, al ser de carácter particular, no se describen¹⁰. En la siguiente figura se muestra los archivos generados.

¹⁰ La forma particular de la solución hace que el código sea único en solución, por lo que no se agrega en este desarrollo

```
1114.TXT: Bloc de notas
Archivo Edición Formato Ver Ayuda

===== Lodos Sys =====
===== REPORTE LOGIN =====
== F.A.H. Team ATL v 5.0.3 ==

Fecha: 20-11-14 01:37
Estado de inicio sesion...
Estado de errores : M
Reportes en : 111420.txt
Inicio Sesion en: 1114.txt
Status : 1
Pruebas realizadas : 0

===== Lodos Sys =====
===== REPORTE LOGIN =====

Fecha: 20-11-14 00:40
Estado de errores : M
Reportes en : 111420.txt
Inicio Sesion en: 1114.txt
ESTADO : 1

===== Lodos Sys =====
===== REPORTE LOGIN =====

Fecha: 20-11-14 01:50
Estado de errores : M
Reportes en : 111420.txt
Inicio Sesion en: 1114.txt
ESTADO : 1

===== Lodos Sys =====
===== REPORTE LOGIN =====

Fecha: 20-11-14 01:53
Estado de errores : B
Reportes en : 111420.txt
Inicio Sesion en: 1114.txt
ESTADO : 1

===== Lodos Sys =====
===== REPORTE LOGIN =====

Fecha: 20-11-14 01:53
Estado de errores : B
Reportes en : 111420.txt
Inicio Sesion en: 1114.txt

111420.TXT: Bloc de notas
Archivo Edición Formato Ver Ayuda

===== Lodos Sys =====
===== NUEVO REPORTE GENERADO =====
20-11-14 01:53

Evento nuevo: 20-11-14 01:53
Conexion con Maestro

Evento nuevo: 20-11-14 01:53
Inicio de pruebas

Evento nuevo: 20-11-14 02:05
Conexion con Maestro

Evento nuevo: 20-11-14 02:05
Carga de datos:
001
004
056
004
005
006
007
008
009
rpm aspas, t aspas, ml Pol, id1, id2, id3, id4, Lodo

Evento nuevo: 20-11-14 02:06
Inicio de pruebas

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4

Evento nuevo: 20-11-14 02:06
Fin una iteracion, se usaran los polimeros con id 1,2,3 y 4
```


ANEXOS

1. HERRAMIENTA GRAFCET

Automatización de Procesos Industriales. Ingeniero de Organización: José Mari González de Durana, Dpto. I.S.A., EUITI e ITT - UPV/EHU Vitoria-Gasteiz, Febrero 2004.

El Grafcet (Graphe de Comands Etape/Transition) es un sistema gráfico de modelado de automatismos secuenciales basado en Cartas de Funciones Secuenciales, SFC. Fue introducido en Francia por P. Girault, en su tesis doctoral. La norma IEC-848¹¹ da una completa descripción de Grafcet y ha sido adoptada por diversos fabricantes para crear interfaces gráficas de usuario que facilitan la programación de sus autómatas programables. Es aplicable, por principio, a un sistema automatizado de producción compuesto de dos partes: una parte operativa (PO) y una parte de comando (PC).

La parte operativa (P.O) está formado por los diversos dispositivos que interactúan sobre el producto: pre actuadores, actuadores y captadores. Los preaccionadores actúan como relés de potencia entre el mando y los actuadores que se encargan de transformar el producto. Los captadores recogen informaciones como posición del producto, alarmas, etc. Que reflejan el estado del proceso en todo momento. Los cambios en el estado del proceso medidos por los captadores son entradas que provocan que el sistema de mando responda de la forma adecuada para la que ha sido diseñado.

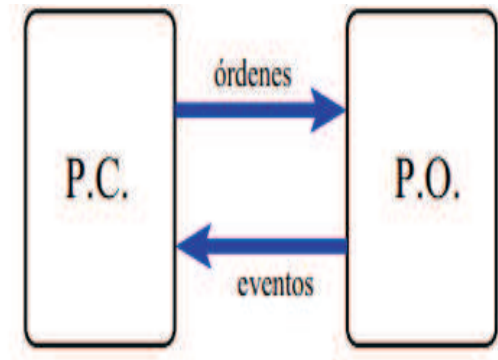


Figura 4.1: Sistema automatizado de producción

La parte de comando (PC) está integrada por los equipos de control: computadores, procesadores o autómatas, junto con los programas y todos los datos precisos.

El Grafcet se compone de los siguientes elementos básicos:

- Etapas o estados a las que van asociadas acciones.
- Transiciones a las que van asociadas receptividades
- Uniones orientadas que unen etapas y transiciones
- Segmentos paralelos.

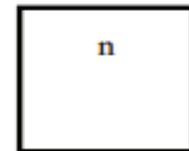


Figura 4.2: Etapa

Etapas

Un sistema dinámico de cierta complejidad evoluciona en el tiempo siguiendo una determinada secuencia de actividades de trabajo o etapas. Una etapa representa un estado o modo de

¹¹ IEC-848 (Preparation of function charts for control sys) presenta una forma general de descripción de las acciones asociadas a las etapas.

funcionamiento estable del sistema o de una parte del mismo. En cada etapa, la Parte de Comando del sistema (al menos la parte asociada a esa etapa) se mantiene invariable.

En Grafcet, cada etapa se representa por un rectángulo en que se escribe un número n que indica su número de orden en cada momento, el sistema tiene una o varias etapas activas (las que están actualmente en funcionamiento) que se marcan con un pequeño círculo negro.

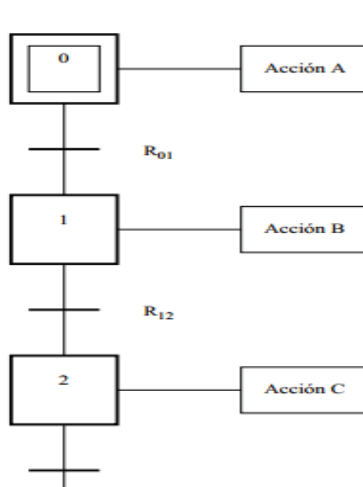


Figura 4.3: Grafcet

La primera etapa que se activa al inicial el Grafcet se llama etapa inicial. Un Grafcet ha de tener al menos una etapa inicial. Estas etapas se encuadran en un doble rectángulo.

Cada etapa puede llevar asociada una o más acciones. Estas acciones se describen, literal o simbólicamente, en uno o varios rectángulos unidos por una línea al rectángulo de la etapa. Según la norma IEC-848, una acción puede estar precedida por un carácter que indica su tipo:

- C: Acción condicionada
- D: Acción retardada
- L: Acción limitada en el tiempo
- P: Impulso
- S: Acción memorizada

Transiciones

Las transiciones se representan con un pequeño segmento horizontal que corta la línea de enlace entre dos etapas. Son etapas de entrada a una transición todas las etapas que conducen a una transición. Son etapas de salida a una transición las etapas que salen de una transición.

La condición o condiciones que se deben cumplir para poder pasar a una transición reciben el nombre de receptividades. En una transición puede haber:

- Una condición simple
- Una función booleana
- La señal de un temporizador o contador
- La activación de otra etapa del Grafcet

Cada transición se une con la etapa anterior y con la siguiente mediante unas rectas horizontales y verticales llamadas líneas de enlace. Las líneas de enlace que tienen sentido ascendente se marcan en el centro con una punta de flecha.

Segmentos paralelos

En un Grafcet se pueden representar varios procesos que evolucionan de forma concurrente. Esto se hace disponiendo de varias secuencias verticales, en paralelo, de etapas y transiciones. La sincronización de estos procesos se hace posible mediante unos segmentos paralelos horizontales. Por ejemplo, en la Figura 4.4 los procesos que se inician en las etapas $n+1$ y $n+2$ discurren en paralelo y están sincronizados con

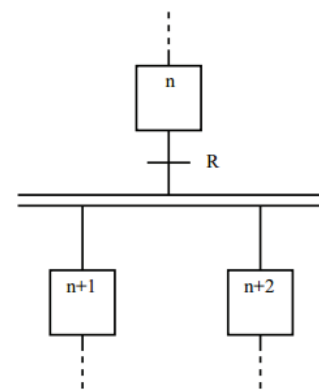


Figura 4.4: Grafcet

el proceso que finaliza en la etapa n . Si está activa la etapa n y se cumple la receptividad R entonces, en ese instante y simultáneamente, pasaran a ser activas las etapas $n+1$ y $n+2$.

Este tipo de conexión sincronizada entre un proceso simple y varios otros en paralelo se llama **divergencia AND**.

Reglas de funcionamiento

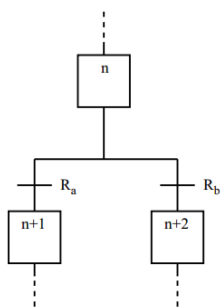
- El funcionamiento del modelo de Grafcet viene definido por cinco reglas o condiciones básicas.
- Activación inicial: en el instante inicial sólo se activan las etapas iniciales y esta activación es incondicional
- Condición de validación: para activar una etapa es necesario que la etapa que le precede esté activada y si preceden varias etapas, en paralelo, entonces todas ellas deben estar activadas.
- Condición de franqueo de una transición: una transición es franqueada si, y solo si, la receptividad asociada es verdadera. Franquear una transición significa primero desactivar la etapa o etapas precedentes y a continuación activar la etapa o las etapas siguientes.
- Franqueo simultáneo: todas las transiciones franqueadas en un determinado instante son franqueadas simultáneamente.
- Conflicto de activación: si una etapa ha de ser desactivada y activada simultáneamente, debido al franqueo simultáneo de las transiciones siguiente y anterior, entonces permanece activa.

Estructura básica

Son las configuraciones que sirven para modelar ciertas situaciones interesantes de los sistemas de eventos discretos y que se utilizan frecuentemente en Grafcet.

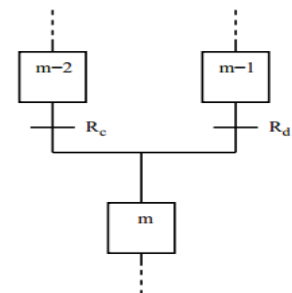
Secuencia simple

La más sencilla de todas las estructuras es la secuencia simple. Como su nombre indica, consta de una serie de etapas seguidas..., n , $n+1$,... que se irán activando una tras otra según se vaya validando las correspondientes receptividades..., R_n , R_{n+1} , R_{n+2} ,...



Divergencia OR

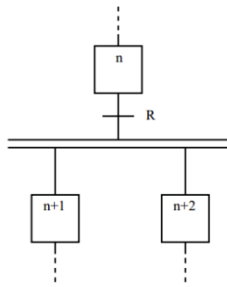
Representa la posibilidad de bifurcación entre secuencias simples. Estando activa la etapa n , final de una secuencia simple, y según cual sea la receptividad R_a , R_b ,... que se valide en primer lugar, el sistema pasará a una de las etapas $n+1$, $n+2$,..., de inicio de respectivas secuencias simples.



Convergencia OR

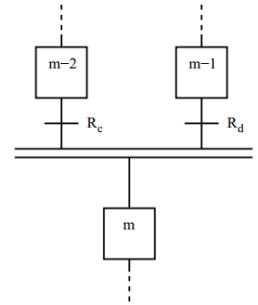
Es la estructura recíproca de la anterior. Indica que un sistema con varias posibles secuencias simples puede pasar desde una de las etapas finales $m-1$, $m-2$,..., en paralelo, a la etapa siguiente n que es el

inicio de otra secuencia simple. Dicho paso se dará en el instante en que una de las receptividades $R_c, R_d, \dots$ que anteceden a la etapa n se haga válida.



Divergencia AND

El símbolo lógico AND indica simultaneidad. La estructura llamada divergencia AND permite modelar el paso de un proceso de secuencia única a otro con varias secuencias concurrentes. Si el sistema se encuentra en la etapa n y la transición R se hace válida, entonces, en ese mismo instante, se activan simultáneamente las etapas $n+1$ y $n+2$.

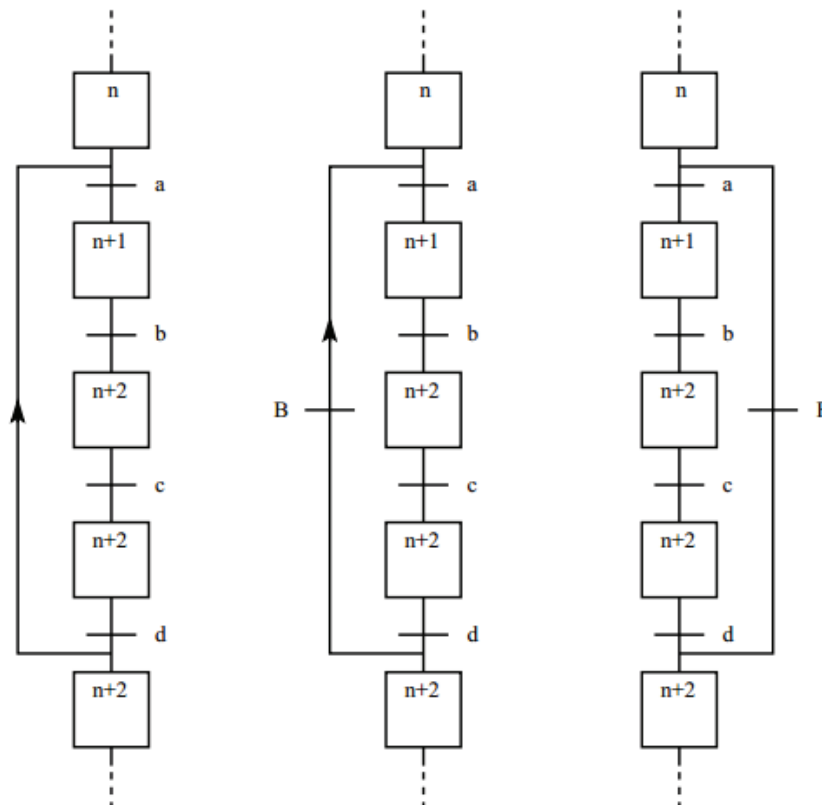


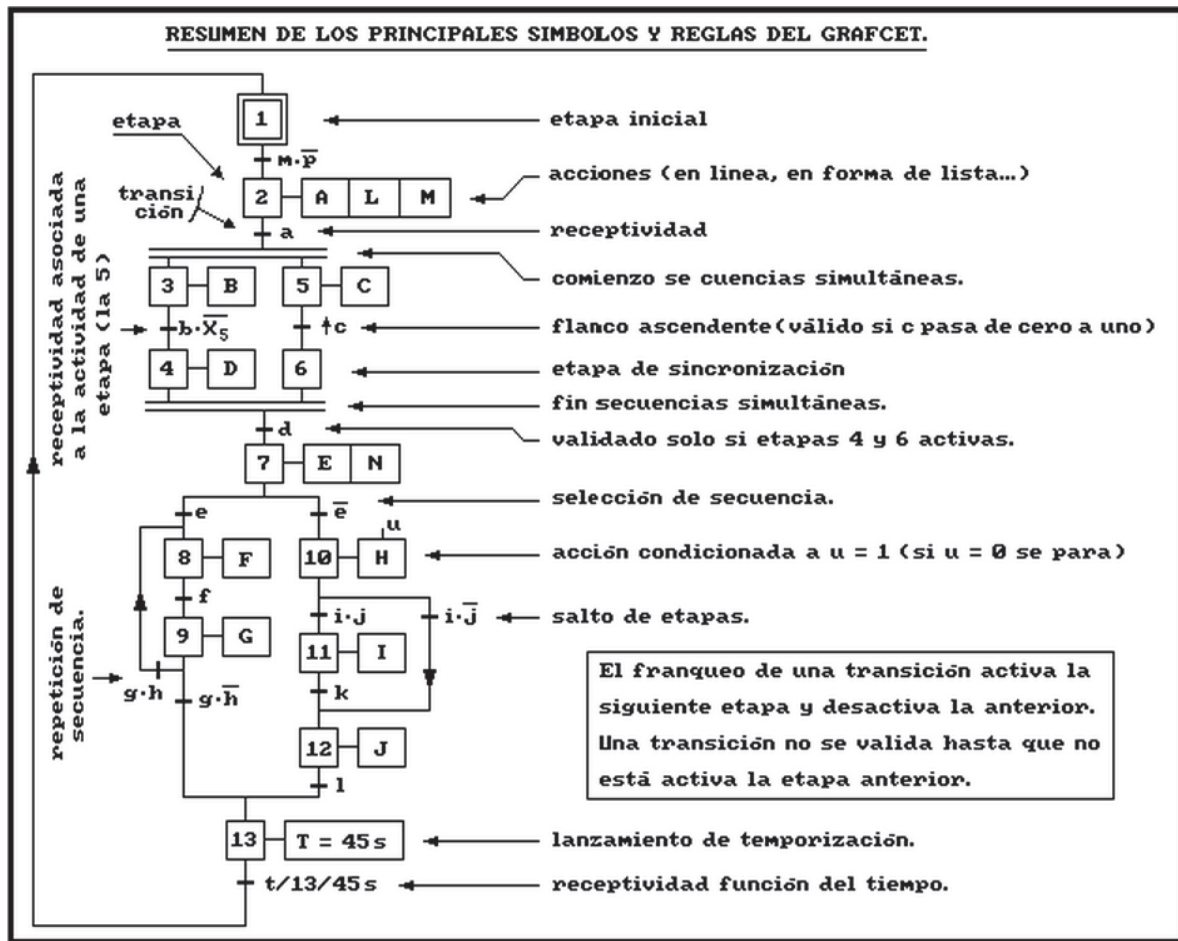
Convergencia AND

En la estructura recíproca de la anterior. Representa el paso simultáneo desde varias secuencias concurrentes, que terminan en las etapas $m-1, m-2, \dots$, a una secuencia única que empieza en la etapa m . Para que en un instante dado se produzca el paso, es necesario que en dicho instante sean válidas todas las receptividades $R_c, R_d, \dots$

Salto

En Grafcet son posibles los saltos condicionales, siempre y cuando este expresada en la receptividad de una transición, y los incondicionales. Los saltos pueden ser hacia adelante o hacia atrás, siendo estos últimos fácilmente identificables por la punta de flecha indicada de la línea de enlace hacia atrás. En la figura se indican algunas posibilidades.





Posibilidades avanzadas

Grafcet no sólo vale para hacer modelos de sistemas de eventos discretos con estructura simple, sino que también ofrece la posibilidad de modelar sistemas más complejos. Para ello cuenta con algunas características avanzadas, tales como el paralelismo, la sincronización la jerarquía

Paralelismo

Grafcet permite el paralelismo. Es posible activar varias etapas a la vez en el estado inicial. También permite mediante segmentos paralelos modelar el paso de un proceso secuencial simple a un proceso compuesto por varias secuencias que trabajan en paralelo.

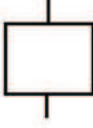
Sincronización

Es la posibilidad de que dos o más etapas se activen a la vez. El paralelismo exige muchas veces la sincronización entre etapas, siempre que al final de un proceso con secuencia simple haya que empezar otro de secuencia en paralelo múltiple o viceversa. Los segmentos paralelos de Grafcet brindan esta posibilidad.

Jerarquía

Macro etapas. La posibilidad de que en un proceso pueda albergar a otros como subprocesos se denomina jerarquía.

Elementos GRAFCET de programación

Símbolo	Nombre	Descripción
	Etapa inicial	Indica el comienzo del esquema GRAFCET y se activa al poner en RUN el autómatas. Por lo general suele haber una sola etapa de este tipo.
	Etapa	Su activación lleva consigo una acción o una espera.
	Etapa	Su activación lleva consigo una acción o una espera.
	Unión	Las uniones se utilizan para unir entre sí varias etapas.
	Transición	Condición para desactivarse la etapa en curso y activarse la siguiente etapa, Se indica con un trazo perpendicular a una unión.
	Direccionamiento	Indica la activación de una y/u otra etapa en función de la condición que se cumpla/n. Es importante ver que la diferencia entre la "o" y la "y" en el grafcet es lo que pasa cuando se cierran (ver más adelante).
	Proceso simultáneo	Muestra la activación o desactivación de varias etapas a la vez.
	Acciones asociadas	Acciones que se realizan al activarse la etapa a la que pertenecen.