

**PROPUESTA DE UN MÉTODO PARA EL DESARROLLO DE SISTEMAS DE
AUTOMATIZACIÓN DE PROCESOS EN PISO PLANTA EN EL CONTEXTO DE
LA TEORÍA DE DISEÑO DE PRODUCTOS**



Universidad
del Cauca

**Leidy Lorena Carvajal Rincon
Karol Yuliana Rodallega Trochez**

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Electrónica, Instrumentación y Control

Ingeniería en Automática Industrial

Popayán, 26 de enero de 2022

**PROPUESTA DE UN MÉTODO PARA EL DESARROLLO DE SISTEMAS DE
AUTOMATIZACIÓN DE PROCESOS EN PISO PLANTA EN EL CONTEXTO DE
LA TEORÍA DE DISEÑO DE PRODUCTOS**

**Leidy Lorena Carvajal Rincon
Karol Yuliana Rodallega Trochez**

Trabajo de grado presentado como requisito para obtener el título de:
Ingeniera en Automática Industrial

Director
Ph.D. Wilber Acuña Bravo

Directores Externos
Ph.D. Jordi Morató (Universidad Politécnica de Cataluña)
Mg. Ángel Fernandez (Universidad Politécnica de Cataluña)

Universidad del Cauca

**Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial**

Popayán, 26 de enero de 2022

Nota de aceptación

Director: _____

Ph.D. Wilber Acuña Bravo

Codirector: _____

Ph.D. Jordi Morató

Codirector: _____

Mg. Ángel Fernandez

Firma del Jurado

Firma del Jurado

Popayán, 26 de enero de 2022

Agradecimientos

Leidy Lorena Carvajal Rincón

En primer lugar le doy gracias a Dios por guiarme en este camino para mejorar mi vida.

A mi papá Carlos Fernando y mi mamá Blanca Nubia, que fueron mi motivación, que confiaron en mis capacidades, me apoyaron en todo momento, con su ejemplo, esfuerzos y sacrificios me formaron y me motivaron constantemente para cumplir mis metas.

A mis hermanos, mi abuela y familiares mas cercanos, por estar presentes en mi crecimiento como profesional.

A mi compañera Karol, por su paciencia, por compartirme sus conocimientos, por ser mi amiga, te agradezco no solo por la ayuda brindada sino por todos los buenos momentos que vivimos.

A mi padrino Javier Burbano, que sin conocerme me brindo su ayuda desinteresada y me dio una voz de aliento para continuar con mis estudios academicos.

A la Familia Burbano Ante y mi pareja Jose E, por acompañarme y apoyarme totalmente en este proceso de aprendizaje.

A mi Tutor Wilber Acuña, por su apoyo y dedicación académica el cual fue fundamental para realizar este proyecto.

A Oscar Amaury por su gran labor como coordinador y a mis maestros por brindarme sus conocimientos.

A mis compañeros mas cercanos de universidad, que hicieron que el proceso de formarnos como profesionales fuera mas ameno.

A la Catedra UNESCO de la sostenibilidad y a Angel Fernandez por recibirme, brindarme una gran experiencia y por su disposición para guiarme en este proyecto.

A la Universidad del Cauca por acogerme y darme la oportunidad de formarme como profesional.

Mil gracias.

Karol Yuliana Rodallega Trochez

Gracias A Dios por permitirme alcanzar cada uno de los logros en mi vida.

A mi primer maestro y amigo, por tener siempre una palabra de ánimo, quien me enseñó a trabajar por mis sueños y tuvo siempre fé en mi. Aunque hoy no pueda abrazarlo físicamente , su amor seguira conmigo todos los dias de mi vida.

A mi ejemplo, mi maravillosa madre y amiga por su gran sacrificio, amor y apoyo. Por estar siempre dandome una palabra de aliento ante las adversidades.

A mis hermanas, Jhobana, Laura y Luisa, quienes son mis fieles compañeras de vida e impulsadoras para alcanzar cada meta impuesta.

A mi familia más cercana que en cada paso de mi vida se encuentran presentes, dandome ánimo durante mis años de formación profesional.

A mis maestros, compañeros y universidad en general por las enseñanzas adquiridas.

A mi compañera de tesis, amiga y confidente del pregrado por estar en cada una de las anécdotas experimentadas en esta etapa.

A mi tutor Wilber Acuña por su esfuerzo y dedicación para culminar de manera exitosa una de las metas planteadas en mi proyecto de vida y Angel Fernandez por su apoyo, paciencia y enseñanza recibida.

A la Catedra UNESCO por brindarme la oportunidad de experimentar cosas nuevas en mi ciclo estudiantil profesional.

Simplemente gracias a cada uno de los mencionados anteriormente por su granito de arena aportado porque me ha permitido crecer personal y profesionalmente.

Resumen

El trabajo de investigación propone una metodología para el diseño de automatización en piso planta que permita a partir de un formalismo matemático describir el proceso de automatización.

El proyecto se centra en realizar una serie de pasos, a partir de los requerimientos y restricciones de diseño para obtener una automatización robusta, flexible y aplicable a diferentes tipos de sistemas. De esta forma, se plantea el proceso de diseño a partir de los requisitos que son representados mediante especificaciones. Según las especificaciones, las descripciones de los productos candidatos son generadas. Las descripciones de los productos se evalúan respecto a los requisitos de diseño prescritos para determinar si el producto diseñado satisface dichas condiciones.

Realizando este ciclo de diseño junto con la representación matemática, se puede representar cualquier sistema de automatización en la etapa de diseño, además de proporcionar una operación lógica que puede realizar la integración y descomposición del sistema que se requiere analizar. El resultado de la propuesta de automatización se representa matemáticamente basado en la teoría axiomática, donde se indica la satisfacción de los requisitos de diseño.

Finalmente, para comprobar la efectividad de la metodología se realiza la implementación en el proyecto AQUARISC, donde se ponen en práctica las descripciones de diseño propuestas y se obtiene como resultado la ecuación característica para el sistema.

Abstract

The research work proposes a methodology for the design of automation in the plant floor that allows, from a mathematical formalism, to describe the automation process. The project focuses on carrying out a series of steps, starting from the design requirements and restrictions, to obtain a robust, flexible automation applicable to different types of systems. The design process is proposed based on the requirements that are represented by specifications. According to the specifications, the descriptions of the candidate products are generated. The products description are evaluated in relation to the designing requirements required to resolve if the product designed satisfies them. By performing this design cycle together with the mathematical representation, any automation system can be represented at the design stage, also providing a logical operation that can perform the integration and decomposition of the system that needs to be analyzed. The result of the automation proposal is represented mathematically based on the axiomatic theory, where the satisfaction of the design requirements is indicated. Finally, to verify the effectiveness of the methodology, the implementation is carried out in the AQUARISC project, where the proposed design descriptions are put into practice and as a result is obtained the characteristic equation for the system.

Keywords: Axiomatic design, formalism, specification design, products, automatic, design requirements.

Índice general

Agradecimientos	IV
Resumen	VI
1. Introducción	1
1.1. Estado del arte	2
1.2. Objetivos	6
1.3. Estructura del documento	6
2. Marco teórico	8
2.1. Diseño de producto	8
2.2. Representación de diseño	8
2.2.1. Representación textual	8
2.2.2. Representación matemática	8
2.2.3. Representación Gráfica	9
2.3. Teoría axiomática del modelado de diseño	9
2.3.1. Operación de estructura	9
2.3.2. Sistema de automatización	10
2.3.3. Definiciones de contexto axiomático	11
2.3.4. Correlación entre conjuntos	12
2.3.5. Definiciones de contexto matemático	12
2.3.6. Diagrama de PERT	12
3. Teoría de Diseño	13

3.1. Análisis de teorías de diseño actuales	13
3.1.1. Matriz de evaluación de alternativas	16
3.2. Planteamiento del proceso de diseño de automatización	16
3.2.1. Proceso de identificación de necesidades	16
3.2.2. Definición de requisitos de automatización	17
3.2.2.1. Requisitos estructurales (R_e):	18
3.2.2.2. Requisitos de rendimiento (R_r):	18
3.2.2.3. Requisitos de proceso (R_p):	19
3.3. Especificaciones de diseño	19
3.4. Proceso de síntesis	20
3.5. Proceso de evaluación	21
3.6. Proceso decisión	22
4. Representación formal de automatización en piso de planta	25
4.1. Representación axiomática	25
4.1.1. Servicios del producto básico	26
4.1.2. Productos de automatización básicos	28
4.1.3. Comportamiento del sistema de automatización con el entorno	29
4.2. Descomposición de las especificaciones de diseño	31
4.3. Combinación de los productos básicos	33
4.4. Implementación de la ecuación	34
4.4.1. Diagrama PERT	34
5. Validación de la metodología propuesta	36
5.1. Planteamiento del problema del caso de estudio	36
5.1.1. Identificación de necesidades	37
5.1.2. Requisitos del sistema	38
5.1.3. Especificaciones de diseño	38
5.2. Representación formal del proceso de la planta piloto	39
5.2.1. Capa 1	39
5.2.1.1. Descomposición en productos básicos de automatización	43

5.2.2. Capa 2	45
5.2.2.1. Descomposición en productos básicos de automatización	50
5.2.3. Capa 3	52
5.2.3.1. Descomposición en productos básicos de automatización	57
5.2.4. Capa 4	59
5.2.4.1. Descomposición en productos básicos de automatización	63
5.2.5. Capa 5	64
5.2.5.1. Descomposición en productos básicos de automatización	66
5.3. Representación formal del sistema de automatización	67
5.3.1. Jerarquía de componentes complejos a básicos	67
5.3.2. Comportamiento de los componentes de automatización.	67
5.4. Combinación de los productos básicos	69
5.4.1. Descripción formal del diseño de la planta piloto	73
5.5. Diagrama PERT	73
5.6. Implementación	74
Bibliografía	84

Índice de figuras

2.1. Sistema de automatización.	10
2.2. Sistema de automatización mapeado en la teoría axiomática.	11
3.1. Fases de proceso de diseño de automatización	17
3.2. Identificación de necesidades de diseño de automatización	18
3.3. Especificaciones de diseño de automatización	19
3.4. Síntesis de diseño de automatización	21
3.5. Evaluación de diseño de automatización	22
3.6. Decisión de diseño de automatización	23
4.1. Jerarquía de componentes complejos a básicos.	26
4.2. Comportamiento de los componentes de automatización	27
4.3. Algoritmo para describir un producto de automatización	31
4.4. Diagrama PERT	35
5.1. Inspección de contenedores artificiales.	38
5.2. Representación del proyecto AQUARISC	67
5.3. Diagrama pert	74
5.4. Diagrama de potencia	76
5.5. PFD AQUARISC	77
5.6. Tablero de automatización industrial	78
5.7. Tablero de automatización industrial en planta	78
5.8. Adquisición y acondicionamiento de datos con Node-Red	79
5.9. Almacenamiento de datos usando nodo InfluxDB	80

5.10. Dashboard de estado de Salidas (Motobombas, electroválvulas)	81
5.11. Prueba de supervisión	81
5.12. Interfaz Ubuntu, con aplicaciones en funcionamiento	82
5.13. Conexión remota via TeamViewer	82

Índice de cuadros

3.1. Análisis de teorías de diseño actuales.	13
3.2. Evaluación de alternativas	16
4.1. Tabla de acción y reacción	27
4.2. Tabla PERT	35
5.1. Requisitos y restricciones del proyecto AQUARISC	38
5.2. Especificaciones de diseño del proyecto AQUARISC	39
5.3. Definir alternativas de solución capa 1	39
5.4. Definir Prestaciones y realizar proceso de evaluación capa 1	40
5.5. Definición de nuevas especificaciones basadas en las necesidades capa1 .	40
5.6. Descripción de subconjuntos capa 1	41
5.7. Descripción solución capa 1	42
5.8. Generación de soluciones de diseño básicas capa 1	43
5.9. Modelo de acción-reacción capa 1	44
5.10. Definir alternativas de solución capa 2	45
5.11. Definir Prestaciones y realizar proceso de evaluación capa 2	45
5.12. Definición de nuevas especificaciones basadas en las necesidades capa 2	47
5.13. Descripción de subconjuntos capa 2	49
5.14. Definir alternativas de solución capa 2	50
5.15. Generación de soluciones de diseño básicas capa 2	50
5.16. Modelo de acción-reacción capa 2	51
5.17. Definir alternativas de solución capa 3	52

5.18. Definir Prestaciones y realizar proceso de evaluación capa 3	52
5.19. Definición de nuevas especificaciones basadas en las necesidades capa 3	53
5.20. Descripción de subconjuntos capa 3	55
5.21. Definir alternativas de solución capa 3	56
5.22. Generación de soluciones de diseño básicas capa 3	57
5.23. Modelo de acción-reacción capa 3	58
5.24. Definir alternativas de solución capa 4	59
5.25. Definir Prestaciones y realizar proceso de evaluación capa 4	59
5.26. Definición de nuevas especificaciones basadas en las necesidades capa 4	60
5.27. Descripción de subconjuntos capa 4	62
5.28. Definir alternativas de solución capa 4	63
5.29. Generación de soluciones de diseño básicas capa 4	63
5.30. Modelo de acción-reacción capa 4	63
5.31. Definir alternativas de solución capa 5	64
5.32. Definir Prestaciones y realizar proceso de evaluación capa 5	64
5.33. Definición de nuevas especificaciones basadas en las necesidades capa 5	65
5.34. Descripción de subconjuntos capa 5	66
5.35. Definir alternativas de solución capa 5	66
5.36. Modelo de acción-reacción capa 5	66
5.37. Tabla de solución de diseños parciales para automatizar	73

Capítulo 1

Introducción

En la actualidad, las diferentes organizaciones se ven obligadas a analizar las necesidades cambiantes de los clientes y desarrollar nuevos productos o mejorar sus versiones actuales, por tanto, el progreso efectivo de estas actividades está directamente relacionado con la función del diseño [1]. Con el surgimiento de las teorías de diseño, se busca definir de manera formal y materializar las soluciones de hardware y software óptimas en productos, ya que en la mayoría de los casos, se establecen soluciones de diseño basándose en prueba y error [2], por tanto, esto es un punto de partida de investigación para la automatización de los procesos en piso de planta.

Entre las teorías de diseño actuales se encuentra el diseño axiomático (DA), el cual es un método formal para desarrollar sistemas mecánicos y modulares requeridos por las organizaciones industriales [3, 4]. Además, mediante los axiomas se proporciona a los diseñadores una base sólida para formalizar problemas de diseño, conceptualizar alternativas de solución, eliminar malas ideas de diseño durante la etapa conceptual, elegir el mejor diseño entre los propuestos y mejorar los existentes [5, 6]. Aunque el DA se ha considerado una técnica valiosa para analizar sistemas y sus requisitos, carece por completo de medios para identificar el orden y tiempo de las soluciones [3]. Por esto, es necesario integrar herramientas como las redes de PERT, que permiten la ejecución de forma estructurada de la solución final de diseño [3, 7].

Dicho lo anterior, la teoría de diseño axiomática resulta apropiada para generar un modelo formal y matemático que describa las etapas de diseño conceptual basadas en la teoría de conjuntos aplicados a la automatización de un sistema. Luego, se representa la estructura sintáctica de los componentes de automatización en piso de planta en evolución jerárquica y en el proceso de diseño dinámico. Es importante conocer el desempeño de este modelo formal, por lo que finalmente se implementará en un estudio de caso en el proyecto AQUARISC, de la Cátedra Unesco de Sostenibilidad de la Universidad Poli-

técnica de Cataluña, el cual es un sistema novedoso en América latina, capaz de realizar el procesamiento de agua residual a partir de humedales artificiales automatizados.

1.1. Estado del arte

A medida que evoluciona la comprensión humana del diseño, la representación utilizada en el diseño también evoluciona [8, 9]. Se necesitan distintas representaciones en las diferentes fases y etapas de desarrollo en el proceso de diseño, para esto, se requiere utilizar una serie de ayudas, herramientas y sistemas de apoyo que permitan llegar a un producto o proceso realizable y esto es lo que generalmente se le considera como métodos de diseño [10]. Los métodos de diseño usualmente ayudan a formalizar y sistematizar las actividades dentro del proceso de diseño y a transferir el conocimiento de diseño, es decir, se extraen los pensamientos del diseñador mediante gráficos y diagramas.

Generalmente, el diseñador tomará los requerimientos del cliente y con base en ello propondrá alternativas para lograr ciertas intenciones de diseño, evaluará estas alternativas de acuerdo con criterios y, en última instancia, aceptará o rechazará cada una durante el proceso de evaluación y la decisión final se registra en forma de documentos de diseño estándar [11]. Desafortunadamente, en la mayoría de las actividades de diseño, el proceso de pensamiento no se registra explícitamente, y poco después de que se completa el proceso, la justificación de las decisiones, como las motivaciones que inician el diseño, se pierden o se olvidan [12]. Esto tiene un impacto severo cuando posteriormente se deben realizar cambios en el diseño original. Se ha argumentado que para poder superar los problemas potenciales durante la modificación y reutilización del diseño, es necesario registrar y representar explícitamente el proceso de pensamiento de este mismo. Existen tres tipos de representación para el proceso del pensamiento de diseño: informal, semiformal y formal.

Las representaciones informales, como grabaciones de audio o video, llamadas telefónicas, correos electrónicos o transcripciones, son fáciles de grabar y comprender, pero son extremadamente difíciles de calcular y razonar. Como caso particular se encuentra la representación textual, que es indispensable en el diseño de modelado y análisis funcional, con esta representación los requisitos de los productos están escritos en formato textual, que si bien se puede aplicar a la descripción de procesos técnicos, es difícil describir la estructura de estos mismos debido a que requieren un nivel más alto de análisis y abstracción [13].

Autores como [8] y [14] establecen que los métodos informales logran tener un poco más de confiabilidad en los diseños de sistemas técnicos al ser integrados con otro método

ya sea formal o semiformal.

Las representaciones semiformales son aquellas donde se tiene cierta uniformidad en las características de un objeto, o un patrón o modelo para realizar el proceso de diseño [8], se utilizan a menudo para representar sistemas de razonamiento de diseño basados en la argumentación, este tipo de representaciones usualmente se muestra en un diagrama, el cual permite tener una visión simplificada y estructurada de conceptos, ideas, construcciones, relaciones, datos estadísticos, etc [15]. las representaciones gráficas se utilizan para transmitir relaciones en forma de diagrama, mediante el uso de símbolos comúnmente entendidos [8]. Se han desarrollado representaciones semi formales como IBIS [16] que surge desde 1970 por Kunz y Rittel [17], donde los diseñadores registran la lógica de su diseño con nodos y arcos, entre sus beneficios está la captura del pensamiento del diseñador y que esto fue base para muchos otras técnicas de modelado de ingeniería como gIBIS, PHI, DRL, QOC, actualmente conocidos como modelos de redes [12]. Sin embargo, este tipo de representación es extremadamente compleja y difícil de analizar, además de que no hay mecanismo para asociar problemas a una decisión de diseño y que se limita a modelar y capturar preguntas de diseño, siendo así poco viable para diseño de automatización.

Por otro lado, se encuentran los estándares que son utilizados generalmente para la representación de procesos de automatización, creados por la International society automation (ISA), donde se aporta información técnica más importante para el campo de la instrumentación, control de procesos y automatización en todo el mundo. Los estándares creados por la ISA más usados para la documentación y diseño de un proceso técnico son los ANSI / ISA 5.1, ANSI / ISA 95 y ANSI / ISA 88 ¹.

El estándar ANSI / ISA 5.1 es uno de los estándares de la ISA más utilizado durante la Ingeniería de diseño de plantas en la realización de planos y documentos; por ejemplo, diagrama de tubería e Instrumentación (PFD), diagramas de Lazo (DLI) y diagramas P&ID; cada uno de estos diagramas representan una forma de analizar y visualizar el proceso, se establecen los lineamientos para representar e identificar los instrumentos o dispositivos y sus funciones de instrumentación y del sistemas [18].

También se encuentra el estándar ANSI / ISA 88 que aborda ampliamente todas las características principales relacionadas con los procesos de fabricación por lotes, describiendo procedimientos y requisitos de equipo durante todas las fases de desarrollo, desde el nivel de empresa hasta la producción [19] los modelos proporcionados son cuatro, estos modelos son lo suficientemente abstractos como para que se puedan aplicar a una amplia variedad de implementaciones de fabricación por lotes, además de que su

¹www.isa.org

representación gráfica se realiza a partir de la elaboración de un gráfico de *grafcet* donde se definen los comportamientos de un sistema lógico, predefinido por sus entradas y salidas [8, 20].

La ANSI / ISA-95 es un estándar internacional para desarrollar interfaces automatizadas entre organizaciones y sus sistemas de control, proponiendo un conjunto de modelos y definiciones para describir las tareas e información de fabricación y producción que deben intercambiarse entre sistemas de información [21], alrededor de este estándar han surgido nuevas propuestas para el intercambio de información utilizando diagramas BPMN (*Business Process Model and Notation*) basados en los modelos de ANSI / ISA-95 [22]. El BPMN contiene actividades, eventos y *getways*, estos grupos se presentan en formas geométricas como cuadrados, rectángulos, rombos, círculos, flechas y líneas. Los cuadrados o rectángulos están conectados por líneas que indican asociación y dirección de recorrido, de tal forma que al finalizar el modelado de este estándar se obtiene como resultado un diagrama de bloques, es decir, una representación pictórica del proceso [23]. En síntesis, estos estándares utilizan métodos gráficos que incluyen diagramas de bloques del sistema, diagramas de flujo de datos, UML ó croquis que son análogos al producto siendo diseñado. La representación gráfica que se utilice depende de la naturaleza del producto [8]. Ahora bien, este tipo de representaciones tiene ciertas limitaciones al momento de hacer un diseño a nivel de piso de planta debido a que en procesos complejos y con un considerable número de instrumentos los diseños generados pueden ser abrumadores visualmente, además de que pueden ser muy laboriosos durante la planeación y el diseño del mismo, debido a que, el modelado de los escenarios para el sistema técnico puede volverse desordenado y poco entendible. De hecho, los investigadores en muchos casos recomiendan incluir hechos básicos, puntuales e información de alto nivel en estos diagramas, dejando por fuera casos que puede dar impacto al sistema técnico [15].

Autores como [12] y [16] afirman que las representaciones de diseño semi formales son herramientas poderosas, pero requieren trabajo, más aún cuando se necesita abarcar muchas actividades de diseño dentro de una sola herramienta. Se puede observar que los procedimientos de modelado antes mencionados siempre dan como resultado una alternativa tras realizar un diagnóstico del proceso en general, pero que la calidad, el nivel de detalle y, por lo tanto, la usabilidad del modelo de producto generado dependen en gran medida de los expertos del dominio y participantes.

Por último se encuentran las representaciones formales las cuales deben cumplir con las condiciones necesarias o con los requisitos establecidos para el diseño, como las reglas integradas en un sistema experto, que pueden ser entendidas por las computadoras. El

propósito de la formalización es convertir lo informal en una estructura formal [15]

Las representaciones formales usualmente se conocen por ser modelos matemáticos que usan notaciones o símbolos para representar los elementos y las relaciones de un sistema, además para expresar su funcionalidad y dependencias, incluyendo en sus lenguajes cálculos aproximados, ecuaciones diferenciales y distribuciones estadísticas. Las representaciones también utilizan teoría de conjuntos o reglas matriciales para describir un sistema [8]. También se encuentran los gráficos de enlaces que representan los sistemas físicos, además vinculan los procesos en un sistema con la comprensión del flujo de energía. [15]

Entre los métodos de diseño populares actuales se encuentra el diseño axiomático (DA) [24] es una teoría de diseño de ingeniería prescriptiva que proporciona una base científica y sistemática para tomar decisiones de diseño [25]. Es un método de modelado de sistemas estructurados como sistemas mecánicos y modulares, entre estas representaciones se encuentra el modelo de matriz de estructura de diseño (DSM) [26] que se ha considerado una buena técnica para el diseño de sistemas, pero que también se utiliza para modelar, sistemas de ingeniería, productos y características, procesos y organizaciones [15]. Mediante los axiomas se da a los diseñadores una base sólida para formalizar problemas de diseño, conceptualizar alternativas de solución, eliminar malas ideas de diseño durante las etapas conceptuales, elegir el mejor diseño entre los propuestos y mejorar los diseños existentes [27,28].

Por otro lado, existen aportes muy importantes como el de Tiacci [29] que propone un formalismo de modelado basado en eventos, un sistema se modela definiendo los cambios que ocurren en el momento de los eventos y la dinámica del sistema se puede describir usando un Gráfico de eventos. La importancia de la simulación Orientada a Objetos está creciendo recientemente bajo el paradigma de Industria 4.0, en el que la optimización vía simulación, simulación en tiempo real, sistemas de decisiones automáticos basados en simulación, análisis de escenarios en línea juegan un papel relevante y que entre sus ventajas permite soportar una metodología de modelado donde los sistemas se describen enlazando componentes de manera análoga a como se enlazan los componentes del sistema, y la dinámica de los sistemas se pueden visualizar en términos de interacciones entre objetos, que tienen su correspondencia física en el mundo real. Sin embargo, su desventaja consiste en requerir una arquitectura mucho más complicada, a su vez, una representación gráfica mucho más compleja del modelo de simulación.

Autores como [30] describen los beneficios de un enfoque integrado ya que estos formalismos de diseño fomentan una comprensión más profunda de la semántica de notación disponible en cualquier técnica. El codesarrollo utilizando múltiples técnicas permite ex-

plotar las fortalezas de las representaciones alternativas del mismo modelo. Por tanto, cabe resaltar que las representaciones formales pueden estar respaldadas por otro tipo de gráficos como IDEF0, metodología MOKA, STEP/DEKLARE, diagramas UML entre otros, que permitan capturar y representar el conocimiento, facilitando el objetivo final para la automatización de un proceso [27].

Dentro de la literatura consultada, no se encontró que en los métodos de diseño formales se representen sistemas de automatización vistos como producto, en general se presentan diseños orientados a productos mecánicos o modulares, y que son limitados a una sola investigación. Por lo tanto, en este trabajo se pretende desarrollar un método de diseño orientado a teorías de diseño para piso de planta que cumpla con las características de flexibilidad, esta situación conduce a un proceso diseño que implique un análisis experimental para definir sistemáticamente todas las estructuras funcionales que se requieren.

1.2. Objetivos

Objetivo general

Proponer un método para el desarrollo de sistemas de automatización de procesos en piso de planta en el contexto de la teoría de diseño de productos.

Objetivos específicos

1. Especificar un formalismo matemático para la descripción de un proceso de automatización en piso de planta en términos de las variables involucradas en el diseño de productos.
2. Determinar las características en términos de requerimientos y restricciones del proceso de diseño de productos, que pueden ser mapeadas a la automatización en piso de planta.
3. Validar la metodología propuesta en una planta piloto del proyecto AQUARISC.

1.3. Estructura del documento

Este trabajo está dividido en 6 capítulos, que siguen una traza lógica y que van desde la base teórica hasta la implementación y validación de la metodología en piso planta, de igual manera se mostrarán las conclusiones y trabajos futuros. En el capítulo II, se

ilustra la teoría desde la problemática del diseño actual, hasta la definición de distintas técnicas utilizadas en el proyecto. En el capítulo III, procedemos a realizar el modelado de diseño conceptual del proceso de automatización, basado en la teoría axiomática y teoría de conjuntos de un producto. En el capítulo IV, se realiza la presentación formal de la metodología propuesta, que va desde la representación axiomática hasta la descomposición y combinación de los productos básicos, de manera que se obtiene un formalismo matemático general para procesos en piso de planta. En el capítulo V, se realiza la aplicación de la metodología a un caso de estudio teniendo en cuenta las restricciones y requerimientos dados por los investigadores del proyecto, de manera que se obtiene como resultado su ecuación formal, la cual es utilizada en la implementación física del sistema y posteriormente su puesta en marcha. En el capítulo VI, se presentan los hallazgos más importantes de la metodología expuesta. Finalmente se plantean algunas propuestas para trabajos futuros.

Capítulo 2

Marco teórico

2.1. Diseño de producto

Es el proceso para la creación de nuevos productos. Es un acercamiento de forma sistemática, para conceptualizar y evaluar ideas convirtiéndolas en productos tangibles, a partir de la innovación en el diseño de productos con soluciones efectivas [31].

2.2. Representación de diseño

Para la definición de un diseño un proceso, maquinaria o producto ya sea nuevo o mejorado, se hace necesario la elaboración de una representación que explique el objetivo, o el procedimiento para llevar a cabo la definición de este [8].

2.2.1. Representación textual

Este tipo de representación es una descripción gramatical, donde se pueden exponer las pautas para el desarrollo de cada una de las fases y etapas del proceso de diseño. Es un método sencillo para representar las especificaciones de diseño, análisis de mercado y aquellas actividades de diseño que requieren un alto nivel de detalle explicativo tal como una descripción de funcionamiento. La información se puede mapear también mediante el uso de tablas [8].

2.2.2. Representación matemática

Se conoce por ser modelos matemáticos aquellos que usan notaciones o símbolos para representar los elementos y las relaciones de un sistema, además para expresar su

funcionalidad y dependencias, incluyendo en sus lenguajes cálculos aproximados, ecuaciones diferenciales y distribuciones estadísticas. Las representaciones también utilizan teoría de conjuntos o reglas matriciales para describir un sistema [14].

2.2.3. Representación Gráfica

Las representaciones gráficas se utilizan para transmitir relaciones en forma de diagrama, mediante el uso de símbolos comúnmente entendidos [15].

2.3. Teoría axiomática del modelado de diseño

La teoría axiomática del diseño es una herramienta lógica para representar y razonar sobre estructuras de objetos, los conceptos más básicos del universo y la relación, por ello se utilizan en el modelado de diseño. Definiendo el universo como el conjunto de cosas y fenómenos observados y postulados, un objeto es cualquier cosa que se pueda observar o postular en el universo y una relación es un aspecto o cualidad que conecta dos o más objetos [8].

Con base en esto se definen dos axiomas en la teoría axiomática del modelado del diseño:

- Axioma 1: Todo en el universo es un objeto
- Axioma 2: Hay relaciones entre objetos

En la teoría axiomática podemos definir el concepto y la simbología de relación como:

$$R = A \otimes B, \exists A, B, R, \quad (2.1)$$

donde A se relaciona con B y R es la relación del objeto A al objeto B .

La relación está definida por propiedades básicas como: *idempotente*, *conmutativo*, *transitivo*, y *asociativo* [32].

2.3.1. Operación de estructura

La operación de estructura es denotada por \oplus y se define como la unión de un objeto y la relación del objeto consigo mismo, así:

$$\oplus O = O \cup (O \otimes O), \quad (2.2)$$

donde $\oplus O$ es la estructura del objeto O , y $(O \otimes O)$ es la relación de O consigo mismo. La operación de estructura proporciona el mecanismo de agregación de componentes mostrado en la Figura 4.1, para representar la evolución del objeto en el proceso de diseño de automatización.

Esta es la base del estudio formal de los requisitos del producto ya que forma un sistema de coordenadas para representar varios requisitos. Específicamente, los resultados son útiles para desarrollar un sistema de automatización y respaldar la recopilación y especificación de los requisitos del producto. [14].

2.3.2. Sistema de automatización

Para la obtención de un sistema de automatización es necesario cumplir con el diseño de requerimientos y de proceso.

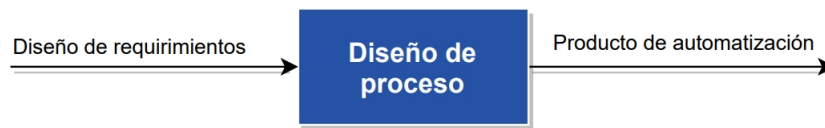


Figura 2.1: Sistema de automatización.

El sistema de automatización, denotado por α , es la estructura del sistema de automatización que incluye tanto el producto S y su entorno E , es decir:

$$\alpha = S \cup E. \quad (2.3)$$

Teniendo en cuenta la operación de estructura se puede representar como:

$$\oplus \alpha = \oplus(E \cup S) = \oplus E \cup \oplus S \cup B. \quad (2.4)$$

El límite entre producto y medio ambiente denotado por B es el conjunto de relaciones entre el producto y su entorno, dado por

$$B = (E \otimes S) \cup (S \otimes E) \quad (2.5)$$

La ecuación (2.5) incluye cuatro componentes: estructura del entorno $\oplus E$, estructura del producto $\oplus S$, relación del medio ambiente al producto $E \otimes S$, y la relación del producto al medio ambiente $S \otimes E$. Gráficamente mapeada en la teoría axiomática se puede representar según la Figura 2.2.

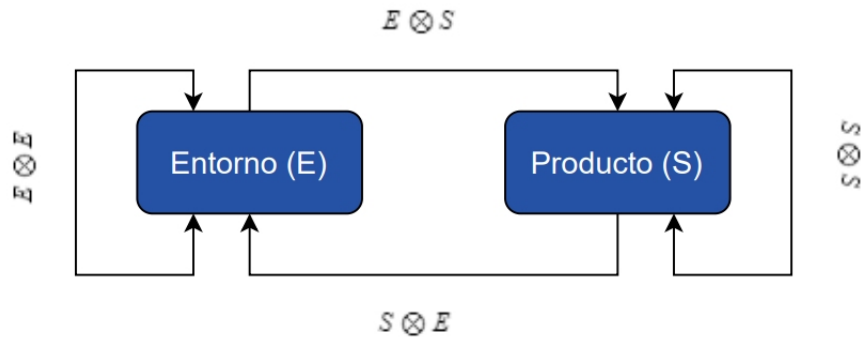


Figura 2.2: Sistema de automatización mapeado en la teoría axiomática.

Estas descripciones se pueden ver como los componentes y como las soluciones de sistemas de automatización en el contexto de planteamiento de diseño.

2.3.3. Definiciones de contexto axiomático

Producto (S): Un producto es un objeto o componente en la naturaleza creado por seres humanos.

Medio ambiente o entorno (E): Es la colección de todos los objetos de la naturaleza excluido el producto.

Sistema de automatización: Puede verse como la unión de un producto (S) y su entorno (E). En este contexto, la estructura de la naturaleza se llama sistema de automatización, denotado por $\oplus\alpha$.

Rendimiento del producto (P): Los rendimientos del producto, denotados por P , son relaciones entre las acciones (A) y las respuestas (R).

$$P \subseteq A \otimes R \quad (2.6)$$

Acción y respuesta: Las acciones, son relaciones del medio ambiente (E) y el producto (S).

$$A \subseteq E \otimes S, \quad (2.7)$$

Las respuestas, son relaciones del producto (S) con el medio ambiente (E).

$$R \subseteq S \otimes E. \quad (2.8)$$

2.3.4. Correlación entre conjuntos

Un conjunto es una colección de elementos con características similares considerada en sí misma como un objeto, ahora bien, existe una clase de correlación que se define en base a la relación entre dos conjuntos X y Z dada por:

$$\forall x \in X, \forall z \in Z (x \approx z \rightarrow z \approx x). \quad (2.9)$$

La clase de correlación del conjunto Z con respecto al conjunto X , denotado como $[Z, X]$, es un subconjunto de X que consta de todos aquellos elementos en x que tienen relación con cada elemento en Z , así:

$$[Z, X]_{\approx} = \{x \in X | z \in Z, x \approx z\}. \quad (2.10)$$

2.3.5. Definiciones de contexto matemático

Relación de recurrencia: Es una ecuación donde cada término de la secuencia es definido como una función de términos anteriores.

Predicado: Un predicado es una fórmula bien formada, de una o más variables definidas en algún dominio específico, que se puede evaluar como verdadero o falso en función de los valores de las variables que ocurren en él.

2.3.6. Diagrama de PERT

La técnica de evaluación y revisión de programas (PERT) es una herramienta estándar para la programación de actividades [33]. PERT es una red de actividades en flecha que representa el comienzo y el final de un proyecto. Para llevar a cabo el proyecto, ciertas actividades deben cumplirse de acuerdo con una secuencia predefinida dada. Esta lógica está representada por un gráfico acíclico dirigido en el que los vértices del gráfico representan los eventos, mientras que las flechas representan las tareas a realizar. Un evento ocurre cuando se han completado todas las actividades anteriores; sólo entonces pueden comenzar las tareas siguientes. El evento se utiliza básicamente para expresar dependencias lógicas entre actividades. En una red PERT, la duración de la actividad se define mediante variables estocásticas que se supone que son independientes entre sí. [34]

Capítulo 3

Teoría de Diseño

3.1. Análisis de teorías de diseño actuales

La Tabla 3.1 presenta un resumen de las teorías de diseño aplicadas actualmente, la cual permitirá guiarnos en la elección de un formalismo, además de ver la utilidad de ellas en la descripción de un producto.

Cuadro 3.1: Análisis de teorías de diseño actuales.

Autor	Formalismo	Concepto	Ventajas	Desventajas
Ching-Hung Lee	Teoría de la resolución inventiva de problemas (TRIZ)	Proporciona una estructura general para la aplicación de la colección de herramientas y técnicas de TRIZ, se rige por ciertas leyes como en la lógica analítica y una forma sistemática de pensar.	Conjunto de herramientas que proporciona un enfoque lógico para desarrollar la creatividad para la innovación y la resolución de problemas inventiva.	Hay pocos estudios que ilustran cómo interpretar y aplicar TRIZ en el diseño de casos empíricos además de ser desafiante en cualquier tipo de práctica.

Continúa en la página siguiente

Cuadro 3.1 – continuación de la página anterior

Autor	Formalismo	Concepto	Ventajas	Desventajas
			Es fundamental el conjunto de soluciones conceptuales a los problemas técnicos. Estas soluciones son una colección de los principios inventivos y las tendencias de la evolución técnica	Existen dificultades asociadas con su aprendizaje debido a que una comprensión completa de TRIZ requiere una inversión sustancial en tiempo y recursos, además dificulta elegir las herramientas adecuadas que a menudo resulta en una pérdida de esfuerzo.
Pries-Heje y Baskerville.	Nexus	Es un conjunto de métodos que permiten construir modelos que conectan numerosas teorías de diseño con soluciones alternativas.	Ayuda a seleccionar el diseño más ideal para el entorno de problemas preciso en cuestión.	Es aplicable a un listado de problemas que puedan solucionar algún problema, sin embargo no está orientado a la descripción de un producto. No se asegura que la elección por el encargado resuelve su problema, ya que parte de la elección se sigue basando en su conocimiento o intuición para elegir.
Elmqvist, M. y Segrestin, B	Teoría C-K	Modela el proceso de diseño a través de la interacción entre dos espacios expandibles: conceptos (C) y conocimiento (K). Donde (C) contiene preposiciones que son interpretables por el diseñador y (K) proposiciones que el diseñador considera verdaderas.	En la fase K es necesariamente multidisciplinaria y eventualmente indicará posibles desarrollos innovadores.	Aunque es una excelente metodología de descripción de ideas en la creación de nuevos productos no introduce un formalismo matemático que pueda ser análogo a la descripción de componentes .
Continúa en la página siguiente				

Cuadro 3.1 – continuación de la página anterior

Autor	Formalismo	Concepto	Ventajas	Desventajas
			En la fase C se organizan subgrupos donde se presenta el trabajo para incrementar el poder creativo de todo el grupo.	
Wenjuan Lia, Zhenghe Songa, C. Steve Suhb	Diseño axiomático	Es una estrategia para el diseño de productos, Proporciona una base teórica de acuerdo con la lógica y el pensamiento racional.	Ofrece herramientas sistémicas y proceso de diseño para completar el diseño.	La teoría AD ayuda a seleccionar un mejor producto, pero no para generar dicho producto. Carece de las descripciones detalladas de los elementos lo que puede causar confusión al diseñador.
Wynn, DC y Clarkson, PJ	Modelo de diseño evolutivo	Los diseñadores trabajan con múltiples representaciones de un diseño emergente.	Se tienen diferentes modelos de producto en los distintos ciclos de diseño, donde se identifican problemas.	Dependiendo del nivel de abstracción será el análisis hasta que se alcance un diseño completamente detallado. Este es un proceso principalmente deductivo, que necesita ser complementado con operaciones lógicas adicionales para manejar un problema.
Wynn, DC y Clarkson, PJ	Teoría del Diseño Universal	Basada en el concepto de espacios de trabajo de diseño (ETD). Cada EDT está limitado por restricciones y relaciones del diseño que se desarrollan a través de cuatro etapas.	El proceso de diseño se desarrolla progresivamente dentro de su EDT mediante tres movimientos escalonados.	El proceso de desarrollo de estas tres dimensiones suelen ser complejas y extensas.

3.1.1. Matriz de evaluación de alternativas

A partir de la Tabla 3.1 se realiza el proceso de selección de la teoría de diseño que más se adapte a los criterios definidos en la Tabla 3.2.

Cuadro 3.2: Evaluación de alternativas

Evaluación de alternativas - Cuantitativo													
Criterios	Valor	TRIZ		Nexus		Teoría C-K		Diseño axiomático		Modelo de diseño evolutivo		Teoría del diseño universal	
Formalismo matemático	5	3	15	0	0	0	0	5	25	0	0	0	0
Formalismo gráfico	3	4	12	5	15	5	15	5	15	5	15	5	15
Flexibilidad de productos	5	4	20	5	25	3	15	4	20	3	15	3	15
Nivel de complejidad	4	5	20	3	12	4	16	4	16	5	20	5	20
Impacto social	4	4	16	3	12	4	16	5	20	4	16	3	12
Calidad de producto resultante	5	4	20	4	20	4	20	4	20	4	20	4	20
Capacidad de innovación	3	4	12	5	15	5	15	4	12	4	12	4	12
TOTAL			115		99		97		128		98		94

3.2. Planteamiento del proceso de diseño de automatización

Una vez elegido el método teórico más adecuado para el estudio de un formalismo que represente la automatización como un producto y que permita la inclusión de sus componentes en piso planta, se continúa realizando el modelado de un proceso de automatización basado en esta teoría y las fases de diseño de un producto.

Las fases del diseño mostrados en la Figura 3.1, son claramente una simplificación de procesos de diseño que incorporan tareas más complejas, estas son las características esenciales para que el proceso de diseño sea exitoso, y que consisten en reconocer las necesidades mediante un proceso de inspección, definir el problema, sintetizar las posibles soluciones, analizar, optimizar las soluciones, y presentar los resultados [35].

3.2.1. Proceso de identificación de necesidades

Las inspecciones, verificaciones o reconocimiento del lugar donde va a ir orientada la solución es muy importante, y es la primer fase que se debe realizar al momento de ejecutar un proceso de diseño de automatización, debido a que esta nos permite a simple vista analizar, detectar problemas, dar sugerencias, que pueden agregar valor o afectar positivamente los ideales del cliente. Este es un proceso de contextualización, por tanto, si esta no ha sido propiamente desarrollada, puede fácilmente dar lugar a malos entendidos y confusiones, así como también a errores de mala interpretación que afectan y

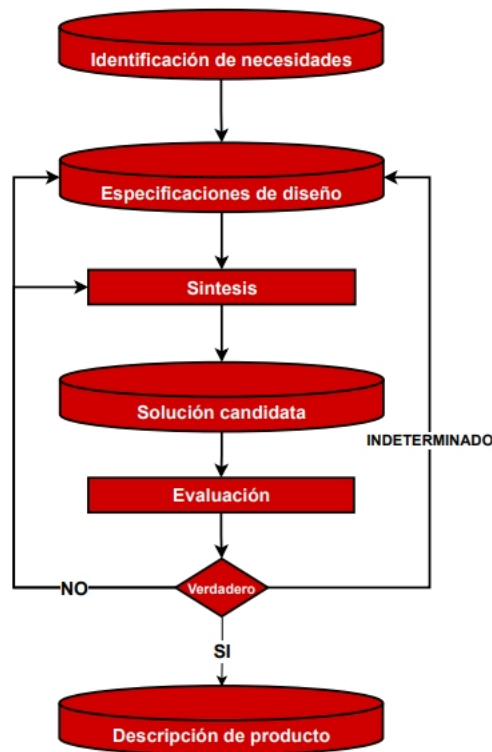


Figura 3.1: Fases de proceso de diseño de automatización

entorpecen desde el comienzo el proceso de diseño de automatización (ver Figura 3.2).

3.2.2. Definición de requisitos de automatización

El diseño es una actividad que inicia con las necesidades del cliente, que surgen con el fin de crear un producto nuevo o mejorar uno ya existente y termina con la descripción de un producto, tal que cumpla con las condiciones a satisfacer. La importancia de tener claro los requisitos radica en que los errores en esta etapa pueden resultar en atributos incorrectos, incompletos, excesivos o en constante cambio durante el proceso de diseño y que traerá en consecuencia soluciones inaceptables o complejas. Es por ello que resulta primordial recopilar la mayor información, además de comprender las necesidades, trabajando con el cliente para definirlas. De igual manera estas necesidades deben ser traducidas a un lenguaje entendible para los diseñadores, por tanto los requisitos se traducen a especificaciones de diseño que se pueden dividir en tres grupos: Requisitos estructurales, de rendimiento y de proceso; descritos a continuación.

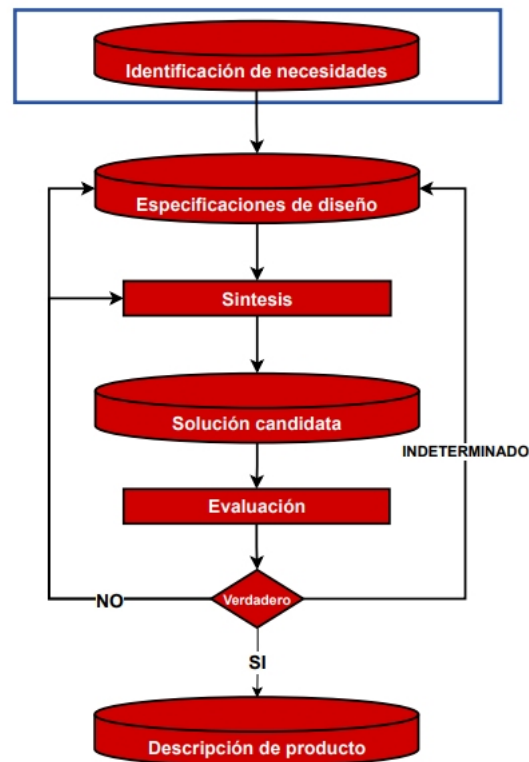


Figura 3.2: Identificación de necesidades de diseño de automatización

3.2.2.1. Requisitos estructurales (R_e):

Son requisitos asociadas directamente a la parte física de la planta y los componentes que la conforman es decir, restricciones sobre las dimensiones, formas, las configuraciones, distribución y materiales de piso planta tales como los instrumentos básicos, ya sean válvulas, motores, materiales como tuberías, tanques, instrumentos de medición y control como transductores, receptores, transmisores, indicadores convertidores, controladores, interruptores y demás configuraciones, convertidores de cables, módulos I/O, electrónica en general, distribución por proceso, distribución por producto, distribución celular, distribución fija, etc. (ver Figura 3.3).

3.2.2.2. Requisitos de rendimiento (R_r):

Son los requisitos asociados a la funcionalidad tales como requisitos de control, supervisión, seguridad, alarmas, capacidad de fabricación, historiadores, redes de comunicación, procesamiento de datos, calidad, requisitos de documentos, etc.

3.2.2.3. Requisitos de proceso (R_p):

Estos requisitos son aquellos que tienen relación directa con las operaciones, tales como las etapas del proceso, rangos de instrumentación, dimensiones, capacidad, balances de masa y energía y parámetros.

3.3. Especificaciones de diseño

Las especificaciones de diseño de automatización son **la elaboración o la formulación de los requisitos**, se expresan como una serie o conjunto de descripciones de productos esperados y sus prestaciones.

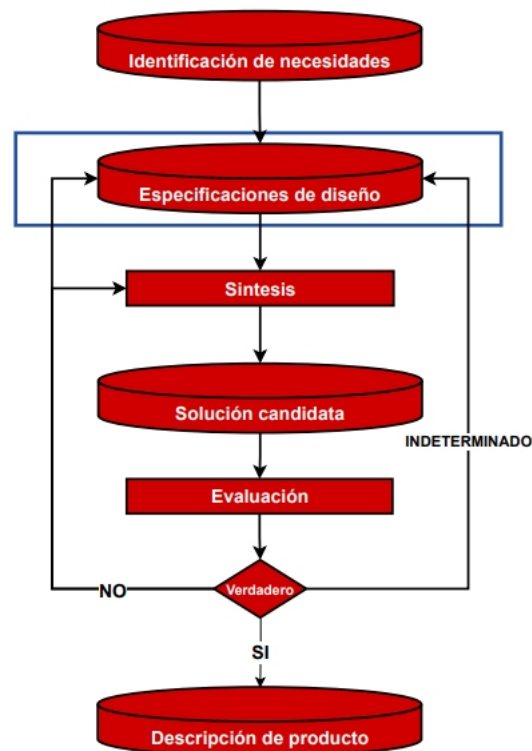


Figura 3.3: Especificaciones de diseño de automatización

Las descripciones de productos (S), son la representación de soluciones de diseño, en el área de automatización estas soluciones comúnmente se describen mediante conceptos del sistema de automatización, configuraciones o esquemas. Y, **las prestaciones (P)** son el rendimiento de los componentes o del sistema de automatización ante las acciones externas que se ejercen sobre éstos, de acuerdo con las leyes del entorno de trabajo que deben cumplir. Denotamos la siguiente ecuación:

$$F_r = S \cup P, \quad (3.1)$$

con,

F_r : Formulación de requisitos de diseño de automatización

S : Descripción del producto

P : Prestaciones de los componentes.

Con esto, podemos definir las especificaciones de diseño de automatización E^d de la siguiente manera:

$$E^d = \lambda(F_r, [F_r]), \quad (3.2)$$

en donde la restricción de la entidad F_r ya sea cuantitativa o cualitativa la representamos como $[F_r]$. La especificación de diseño puede verse entonces como un predicado $\lambda(F_r, [F_r])$ tal que,

$$E^d = \lambda(F_r) \quad (3.3)$$

3.4. Proceso de síntesis

Es el proceso (ver Figura 3.4), en el cual se propone un conjunto de descripciones de soluciones candidatas basadas en especificaciones de diseño. El proceso de búsqueda, denotado por m_a^s , o conocimiento de síntesis son las posibles soluciones de producto ya sea software o hardware y se define como,

$$S = m_a^s(E^d). \quad (3.4)$$

El conjunto de descripciones es igual a la búsqueda que realicemos en las posibles soluciones de automatización evaluadas en los requisitos expresados por el cliente. Esta búsqueda puede ser de conocimiento abstracto, casos de diseño anteriores o cualquier otra información que respalde el proceso.

En este proceso de síntesis pueden existir muchas descripciones de productos para satisfacer los requerimientos. Sin embargo, entre más descripciones de productos existan, más posibilidades de que el diseño final sea óptimo. Cabe destacar que esta etapa da como resultado la descripción del producto candidato que será evaluado posteriormente.

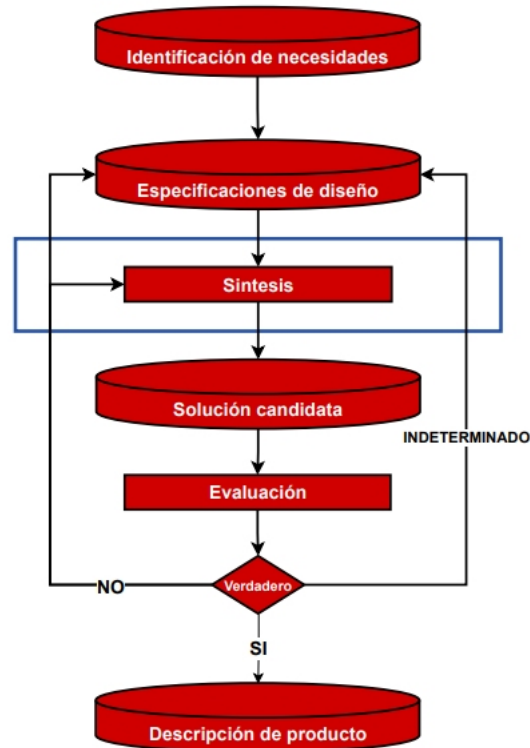


Figura 3.4: Síntesis de diseño de automatización

3.5. Proceso de evaluación

En este proceso se evalúan los productos de automatización candidatos con respecto a las especificaciones de diseño. Es decir, al momento de obtener la descripción, sus propiedades, incluyendo su relación con la parte estructural, el rendimiento y el proceso, se derivan del conocimiento de las propiedades (ver Figura 3.5). Este tipo de conocimiento es denotado como m_a^c , entonces,

$$F_r = m_a^c(S). \quad (3.5)$$

Cabe aclarar que la información relativa a la relación estructural, el rendimiento y el proceso se puede obtener del mismo conocimiento de la propiedad. Este conocimiento se puede generar o existir a partir de conocimientos científicos, experimentos bien diseñados, experiencia por parte del diseñador y otros medios de apoyo. Esta evaluación tiene el fin de seleccionar el mejor producto final que satisface los requisitos de diseño, a nivel de piso de planta a partir de un proceso determinista.

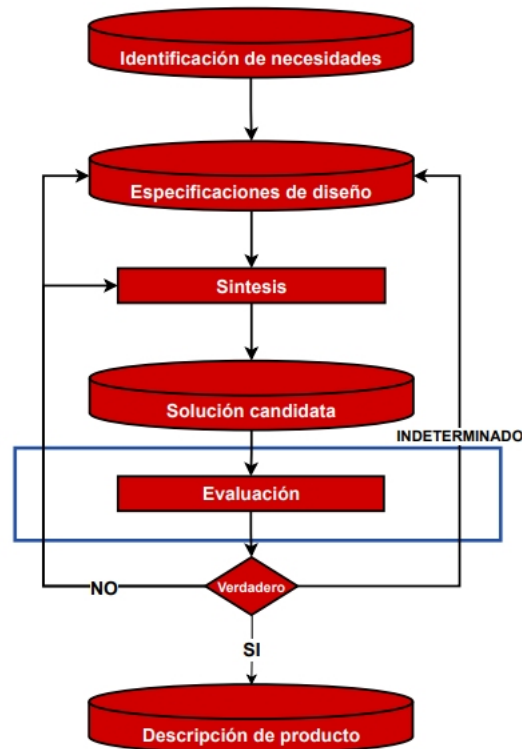


Figura 3.5: Evaluación de diseño de automatización

3.6. Proceso decisión

Desde un punto de vista lógico, siempre existe una forma de determinar si una descripción propuesta satisface las especificaciones de diseño indicadas. Para determinar si es satisfactorio o no el diseño de automatización, se asigna un valor en la siguiente ecuación:

$$\lambda(F_r, [F_r]). \quad (3.6)$$

donde λ es el operador que indica el nivel de satisfacción a los requisitos presentados para realizar el diseño en función de la formulación de requisitos de diseño de automatización (F_r) con respecto a las restricciones cuantitativa o cualitativa $[F_r]$. De esta manera, se realiza la asignación del valor ya sea:

$$\lambda = \begin{cases} \lambda = 1 & \text{cuando } E^d \text{ es satisfactoria} \\ \lambda = 0 & \text{cuando } E^d \text{ es insatisfactoria} \\ \lambda = -1 & \text{cuando } E^d \text{ es indeterminado} \end{cases}$$

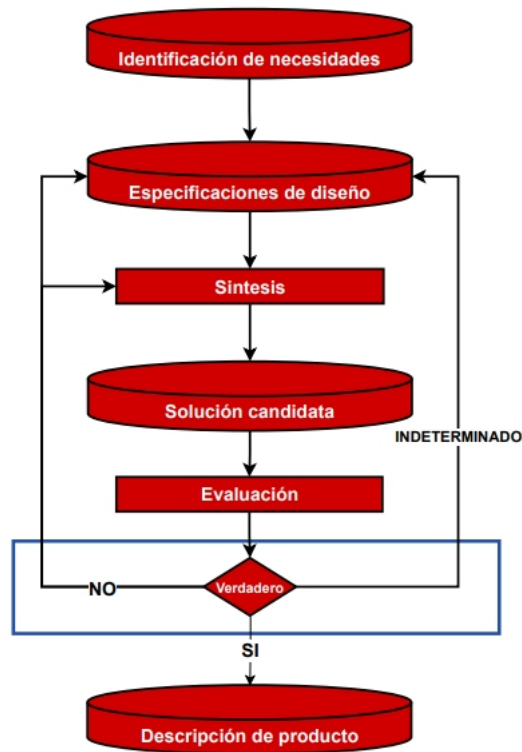


Figura 3.6: Decisión de diseño de automatización

El objetivo del proceso de diseño es generar una descripción del producto para que:

$$E^d = \lambda(F_r) = 1. \quad (3.7)$$

Reemplazando F_r , de la ecuación (3.3), tenemos que:

$$E^d = \lambda(F_r) = \lambda(m_a^c(S)) \quad (3.8)$$

Esto significa que las especificaciones de diseño, van a ser igual a el conocimiento de las propiedades del producto de automatización evaluadas en esa descripción y siendo aceptadas o no según el coeficiente ya sea 0, 1, -1. Por ende, se asume que para que exista una solución válida, el proceso de síntesis y de evaluación deben ser procesos lógicamente verdaderos, debido a que el primer proceso permite ampliar la posibilidades de solución de diseño, mientras que la evaluación permite disminuir este espacio solución. De manera más formal, se puede representar como:

$$E^d \rightarrow S \longleftrightarrow \lambda(F_r) \text{ es verdadero.} \quad (3.9)$$

En este proceso, la interpretación del valor de verdad de requisitos de diseño y deter-

minación del valor de las descripciones de productos son mutuamente dependientes, de aquí también provienen las dificultades del diseño. Sustituyendo la ecuación (3.8) en la ecuación (3.4), obtenemos lo siguiente ecuación:

$$S = (m_a^s)(\lambda')(m_a^c(S)). \quad (3.10)$$

Esta ecuación, es la representación matemática que rige el proceso de diseño de automatización, es una ecuación recursiva y con base en las observaciones anteriores se puede definir como una transformación de los requisitos de diseño en E^d , que satisface las restricciones de acuerdo con el conocimiento de la propiedad m_a^c determinado por la descripción del producto S .

En resumen, para satisfacer las especificaciones de automatización, se generan algunas propuestas que posteriormente se deben justificar. Si la propuesta de diseño satisface las especificaciones entonces podría aceptarse como una solución de diseño, si la solución no es lo suficientemente detallada como para ser evaluada, deberá ser perfeccionada. En un caso contrario, la propuesta de automatización debe ser modificada o descartarse y en este caso se deben recomendar nuevas propuestas o se debe reformular todo el problema del diseño.

Capítulo 4

Representación formal de automatización en piso de planta

4.1. Representación axiomática

Las representaciones se han presentado como el fundamento de la matemática moderna, desde un punto de vista lógico que busca una reconstrucción sistemática de la totalidad de las matemáticas. Estas representaciones permiten la justificación de las teorías matemáticas donde se encuentran métodos como las representaciones gráficas y físicas [36]. De este modo, para la representación axiomática se establece y se describe la evolución del proceso dinámico del diseño, en donde cada componente involucrado incorpore el factor de tiempo. Estos objetos incluyen requisitos de diseño, descripciones de productos y rendimientos del producto. Al representar un producto de automatización, lo primero que se define es el conjunto de productos básicos cuyo rendimiento se puede definir independientemente. En automatización estos productos son las conexiones y elementos básicos de piso planta, por ejemplo: el tipo de cable de conexión (capa física), tipo de lenguaje de programación, protocolo de comunicación, etc. Los cuales son indispensables para el funcionamiento general del sistema de automatización.

De esta manera se puede definir el sistema como un conjunto de productos básicos relacionados por conectores hasta el nivel en el que los componentes son productos estándar (S_a). Esta descomposición se realiza por medio de una representación gráfica, de manera que sirva como justificación a la teoría axiomática y se define en la siguiente ecuación:

$$\begin{aligned} S[0] &= \{S(0, 0, 0)\} \\ S[c] &= \{S(c, i_c, r) | i_c = 1, 2, \dots, m\}, \quad \forall c, 1 \leq c \leq n. \end{aligned} \tag{4.1}$$

Esta ecuación implica la descomposición de las descripciones de producto de diseño donde:

- C : es la capa del árbol y que describe el componente más general. En este sentido la capa cero, es la capa es el sistema de automatización
- r : es la posición relativa de cada nodo en el árbol con referencia a su nodo padre.
- $S[c]$: Es la descripción del producto de la capa del árbol.
- $S(c, i_c, r)$: representa el i_c nodo en la capa del árbol y r es la posición con referencia a su nodo padre.

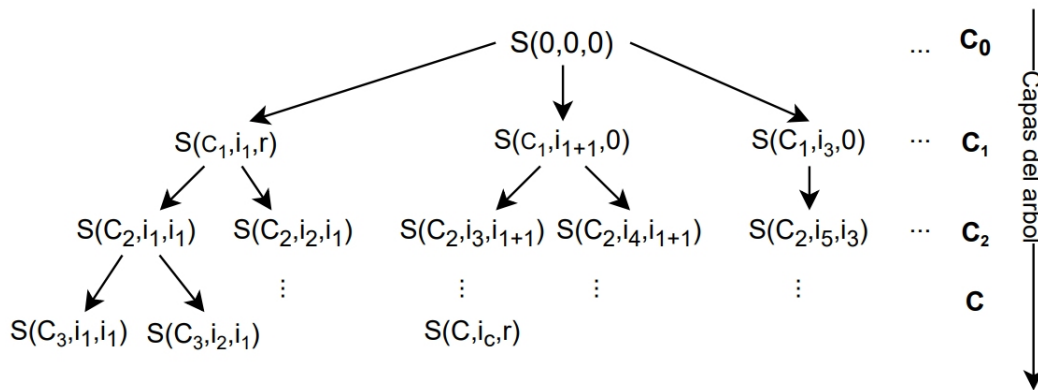


Figura 4.1: Jerarquía de componentes complejos a básicos.

4.1.1. Servicios del producto básico

Los servicios del producto básico son las respuestas de un producto a acciones externas, impuestas de acuerdo con las leyes en el entorno de trabajo del producto. Para definir el servicio del producto básico, es necesario determinar la descripción del producto y de esta manera, se obtiene un **conjunto de rendimientos de productos básicos** (P_a).

Ahora bien, similar a las descripciones de los productos básicos que surgen al realizar la descomposición del sistema de automatización, el rendimiento también se puede descomponer en servicios de componentes básicos. Análogo a la ecuación (??), tenemos la siguiente representación:

$$\begin{aligned}
 P[0] &= \{P(0, 0, 0)\} \\
 P[c] &= \{P(c, i_c, r) | i_c = 1, 2, \dots, m\}, \quad \forall c, 1 \leq c \leq n.
 \end{aligned}
 \tag{4.2}$$

Esta ecuación implica la evolución de las prestaciones de los productos del proceso de diseño, para definir el rendimiento del producto básico se plantea que es necesario cono-

cer, la acción y la respuesta del servicio, que se pueden definir matemáticamente como una relación entre el conjunto de acciones A y el conjunto de respuestas R . Denotando los rendimientos de los productos como P , de esta manera tenemos que:

$$P \subset A \times R. \quad (4.3)$$

Para completar la definición, se analiza el rendimiento del producto por medio del patrón de desempeño de acción y respuesta que se realiza mediante las interacciones mutuas entre los componentes que forman el sistema de automatización mostrados en la Figura 4.2. En un proceso de automatización, cualquier componente puede estar sujeto a las acciones de otros componentes. De esta manera, el desempeño de todo el producto de automatización se logra mediante el desempeño de sus componentes individuales. Estos comportamientos pueden ser de tres tipos, acción, reacción o cambio de estado.

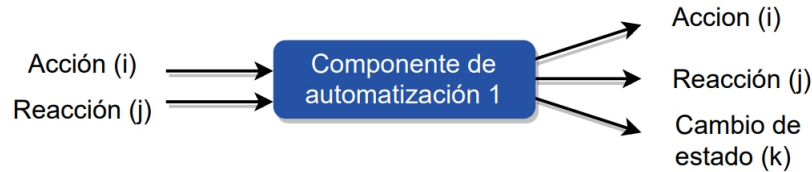


Figura 4.2: Comportamiento de los componentes de automatización

Dicho lo anterior podemos mapear el desempeño del producto en los diferentes niveles de abstracción, como se muestra en la tabla 4.1, donde A, son las acciones, R las reacciones, P es la prestación del componente y S la descripción del producto.

Cuadro 4.1: Tabla de acción y reacción

Acción y Reacción			
Descripción de producto	Acción	Reacción	Rendimiento del producto
S[0]	A[0]	R[0]	P[0]
S[1]	A[1]	R[1]	P[1]
S[2]	A[2]	R[2]	P[2]
S[n]	A[n]	R[n]	P[n]

4.1.2. Productos de automatización básicos

El conjunto de los productos básicos de automatización contienen descripciones y prestaciones bien definidas. Además, de acuerdo con la ecuación (3.5), podemos definir un conjunto de propiedades de conocimiento de los productos básicos tal que:

$$m_a^c \subset S_a P_a, \quad (4.4)$$

donde,

S_a : es un conjunto de descripción de producto básicos

P_a : es un conjunto de rendimiento básicos.

m_a^c : puede ser conocimiento de diseño de automatización, experimentos o experiencia existente por parte del diseñador.

Con base en esto, se supone la existencia de un conjunto de especificaciones de diseño básicos E_a^d , que son restricciones en las descripciones y el rendimiento de productos básicos definidos así:

$$E_a^d = \lambda(Fr, [Fr_a]). \quad (4.5)$$

Ahora bien, **la solución final de diseño de automatización** se genera a partir de las descripciones básicas de automatización, el rendimiento de los productos básicos y los requisitos de diseño básicos, por lo tanto esta solución debe satisfacer la ecuación general de diseño (3.10). En este sentido, con el conocimiento de síntesis y el conocimiento de propiedad básicos m_a^s y m_a^c respectivamente, se obtiene la ecuación que rige el diseño básico como:

$$S_a = (m_a^s)(\lambda')(m_a^c(S_a)). \quad (4.6)$$

Se puede notar que la ecuación (4.6) corresponde al diseño básico de automatización, indicando que es más fácil definir el conocimiento de síntesis y de propiedad en estos niveles, y que se puede definir su comportamiento y sus posibles búsquedas de solución a la descripción básica presentada. De este modo, se propone el siguiente método para resolver la ecuación anterior:

1. Determinar un conjunto de producto básicos de automatización S_a según los requisitos de diseño dados E_a^d y conocimiento de síntesis predefinido m_a^s :

$$S_a = m_a^s(E_a^d), \quad (4.7)$$

es decir, la descripción del producto básico estará dado por el conocimiento de síntesis, que recordando su definición es la búsqueda de las posibles soluciones

que estarán dadas en función de los requerimientos de diseño.

2. Determinar el desempeño de los productos candidatos P_a según la descripción del producto y el conocimiento de propiedad m_a^c

$$P_a = m_a^c(S_a). \quad (4.8)$$

Al momento de definir la descripción del producto básico de automatización en el paso anterior, es posible determinar su desempeño a partir del conocimiento de propiedad el cual indicará el funcionamiento de la descripción para la automatización del sistema.

3. Evaluar los productos candidatos:

$$E_a^d = \lambda'(F_{r_a}). \quad (4.9)$$

Teniendo en cuenta que el proceso de diseño debe satisfacer las especificaciones de diseño es necesario evaluar el producto básico candidato, a partir del operador que indica el nivel de satisfacción, en función de la formulación de los requisitos básicos.

4. Si se satisfacen los requisitos de diseño se finaliza el proceso de diseño básico parcialmente. De lo contrario, se debe repetir el proceso desde el paso 1.

4.1.3. Comportamiento del sistema de automatización con el entorno

El comportamiento del sistema de automatización con el entorno en el que se encuentra es un factor de diseño esencial que influye directamente en el sistema, por ello es necesario tener en cuenta su accionamiento ante posibles sucesos. Se realiza el diseño a partir del conocimiento de la propiedad debido a que dicho conocimiento es un subconjunto de leyes en el entorno del trabajo del producto, es decir, podemos prever qué resultados puede tener cierto diseño básico, ante factores externos al sistema. Matemáticamente, se puede representar como una clase de correlación del conjunto de leyes ambientales con respecto a las descripciones de productos y especificaciones de diseño de la siguiente manera:

$$m_a^c = [S_a \cup [P_a], E]_{\approx}, \quad (4.10)$$

donde:

- E : es el conjunto de leyes en el entorno de trabajo del sistema de automatización.
- $[S_a \cup [P_a], E]_{\approx}$: es una clase de correlación del conjunto $S_a \cup [P_a]$ con respecto al conjunto E .

Para el caso en que m_a^c resulte ser un conjunto vacío, se debe adquirir el conocimiento de desempeño de un producto. Esta es la razón por la que el modelado, el análisis, los experimentos y la creación de prototipos a menudo son necesarios para realizar una tarea de diseño. Luego, al realizar la representación de las especificaciones de diseño, las descripciones de los productos y las prestaciones de los productos, las asignaciones entre ellos se pueden utilizar para describir los procesos de soluciones anteriores.

De esta manera se realiza la continuación del método de automatización a partir de los siguientes pasos :

1. De acuerdo con las especificaciones de producto provisionales generadas S_a , se derivan los rendimientos correlacionados P_a , que permitirán determinar especificaciones de diseño aumentadas $(E_a^d)^+$.

Simbólicamente: $\forall S_a [P_a] \exists m_a^c ((m_a^c = [S_a [P_a], E]_{\approx}) \cap (m_a^c : S_a \rightarrow P_a))$

Es decir, a partir del conocimiento de propiedad generado por las leyes del entorno pueden surgir nuevas especificaciones de diseño aumentadas, que se debe tener en cuenta para realizar el diseño completo y robusto del sistema.

2. Las especificaciones de productos provisionales S_a , y los rendimientos correlacionados P_a , se evalúan contra las especificaciones de diseño aumentadas $(R_a^d)^+$, a continuación, se obtienen el subconjunto de descripciones de producto provisional $S^{[p]}$ y prestaciones de producto provisional $P^{[p]}$.

Simbólicamente: $\exists S^{[p]} \in S_a, \exists P^{[p]} \in P_a (\chi(S^{[p]}) \wedge (\chi(P^{[p]}))$, si $S^{[p]} = \emptyset P^{[p]} = \emptyset$, ir al paso 1

A partir de este paso se realiza una retroalimentación donde se verifica la existencia de una prestación de producto provisional $P^{[p]}$, el cual indica que existe la prestación para la especificación de diseño aumentada, además de esto se debe determinar los subconjuntos de descripciones de productos provisionales $S^{[p]}$, es decir en este punto pueden surgir una o más especificaciones respecto a los requerimientos de diseño aumentados. En caso de no existir se debe realizar nuevamente el proceso de diseño desde el paso indicado.

3. La intersección de las especificaciones del producto S_a y $S^{[p]}$, constituyen la solución de diseño final.

Simbólicamente: $S = S_a \cap S^{[s]} \cap S^{[p]}$ si $S = \emptyset$, ir al paso 1

Cabe señalar que la elección de productos básicos y a su vez, el rendimiento de los productos básicos y los requisitos de diseño es de hecho artificial y depende de la experiencia y el conocimiento de los diseñadores. Por tanto, el diseñador será el encargado de determinar los niveles de complejidad y abstracción de la descripción del producto en la definición del conocimiento de síntesis, donde en realidad se tendrán las posibles soluciones a los requerimientos impuestos por el cliente.



Figura 4.3: Algoritmo para describir un producto de automatización

4.2. Descomposición de las especificaciones de diseño

La descomposición de las especificaciones de diseño se puede asemejar a la operación matemática de la suma, es decir, cuando descomponemos un número en varios sumandos, lo que hacemos es separarlo en otros números que se pueden sumar para obtener el valor original. Siguiendo esta analogía, en el diseño lo que se pueden separar son las especificaciones de diseño originales E^d , en dos partes con respecto a las especificaciones de diseño básicas E_a^d . En el primero se encuentra la intersección del conjunto E^d y el conjunto E_a^d y el segundo es el subconjunto que no tiene ninguna intersección con las especificaciones de diseño básicas. En el último caso, se requiere una especie de estructura de requisitos para transformar las especificaciones en básica. La estructura se

puede deducir sustituyendo las ecuaciones (4.5) y (4.2) en la ecuación (3.2), así,

$$E^d[K] = (F_r(k), [F_r(k)]) \quad (4.11)$$

$$F_r[k] = S[k] \cup P[k]. \quad (4.12)$$

Esta ecuación representa la evolución de las especificaciones de diseño con referencia a los niveles de abstracción y complejidad de las descripciones del producto de automatización desarrolladas por el diseñador. De tal manera que, proporciona una forma de separar los diseños originales en un conjunto de especificaciones básicas de la siguiente manera:

$$E^d = \{e_i^{ad} | e_i^{ad} \in E_a^d, i = 1, 2, 3, \dots\}. \quad (4.13)$$

A partir de esta ecuación las descripciones generadas se van a descomponer gradualmente hasta tener especificaciones básicas, sin embargo en este proceso se pueden generar conflictos, debido a que cada especificaciones tiene prestaciones y pueden afectar la solución dada en los pasos anteriores. Por consiguiente, es posible que se requieran múltiples pasos de descomposición para solucionar dicho problema, por esto, se muestra a continuación una descomposición recursiva, dinámica e incremental:

$$\tau(r_i^{ad}, (E_a^d)^+) = ((E_a^d)^+ / e_i^{ad}) \cup (e_i^{ad}) \downarrow (E_a^d)^+ / e_i^{ad}), \quad e_i^{ad} \in E_a^d, \quad (4.14)$$

donde, τ es el operador de descomposición de las especificaciones de diseño, que descomponen una especificación de diseño aumentada $(E_a^d)^+$ en una especificación de diseño básica (e_i^{ad}) , el símbolo “/”, representa la diferencia de conjuntos y \downarrow , representa los conflictos entre conjuntos.

Esta ecuación muestra que la especificación de diseño E_i^d generada consta de dos partes. En la primera se determinan la diferencia entre los conjuntos de la especificación de diseño básicas e_i^{ad} y de la especificación de diseño aumentadas $(E_a^d)^+$. Mientras que el otro es el conflicto entre la especificación de diseño básicas e_i^{ad} y la diferencia encontrada anteriormente. De esta manera, se puede determinar la diferencia y los tipos de conflictos que tienen los conjuntos mencionados anteriormente, de manera que pueda el diseñador determinar la especificación que más se adapte para el sistema en ese nivel abstracción que se encuentre.

4.3. Combinación de los productos básicos

La combinación de los productos básicos, se realiza a partir de la descomposición de todos los productos o componentes definidos en cada capa de abstracción, y que combinados de forma incremental partiendo desde el producto más básico se genera un producto parcial completo:

$$S_p^0 = \emptyset \quad (4.15)$$

$$S_p^i = \mu (S_a^i, S_a^{i-1}), \quad (4.16)$$

donde μ , es el operador de combinación de productos, y S_p^i , es el producto parcialmente definido.

Cuando un producto básico generado se combina en el producto parcialmente definido, algunas de sus prestaciones pueden entrar en conflicto entre sí. Entonces, el conflicto se convierte en un nuevo requisito el cual debe ser solucionado. Dicho lo anterior, se puede definir un conflicto (\updownarrow), como el rendimiento o los conflictos estructurales entre el producto básico recién generado y el producto parcial existente.

$$r_i^{ad}((E_a^d)^+ / r_i^{ad}) = F_{ra}^i \updownarrow F_{rp}^i, \quad (4.17)$$

$$F_{ra}^i = m_a^c(S_a^i), \quad F_{rp}^i = m_a^c(S_p^i), \quad (40)$$

$$F_{ra}^i = S_a^i \cup P_a^i F_{rp}^i = S_p^i \cup P_p^i. \quad (41)$$

Esta es una elaboración formal del proceso de diseño, el proceso evoluciona al transformar las ideas abstractas, generales y simples de un producto de automatización en descripciones que son mucho más concretas y complejas del producto, hasta que se logre la solución final, manteniendo siempre el marco de requisitos del cliente. En otros términos, la representación axiomática a partir de lo expuesto anteriormente se define de la siguiente manera:

Inicialmente, el sistema de automatización $S = S_p^0 = \emptyset$; es un conjunto vacío, ya que en este punto no existen descripciones de soluciones para el sistema.

1. Obtenemos una primera solución de diseño parcial del sistema de automatización dado por el algoritmo de descripción de productos. $S = S_p^0 \cup S_p^1$;
 $\oplus S = \oplus (S_p^0 \cup S_p^1) = \oplus s_1^1$
2. Obtenemos una segunda solución de diseño parcial del sistema de automatización dado por el paso anterior. $S = S_p^0 \cup S_p^1 \cup S_p^2$

$$\oplus S = \oplus (S_p^0 \cup S_p^1 \cup S_p^2) = \oplus s_1^2 \cup \oplus s_2^2 \cup (s_1^2 \otimes s_2^2) \cup (s_2^2 \otimes s_1^2)$$

En caso de existir una relación, podemos simplificar tal que:

$$(s_1^2 \otimes s_2^2) \cup (s_2^2 \otimes s_1^2) = r_{12}^2$$

Obteniendo la siguiente ecuación:

$$\oplus S = \oplus (S_p^0 \cup S_p^1 \cup S_p^2) = \oplus s_1^2 \cup \oplus s_2^2 \cup r_{12}^2$$

3. Obtenemos una tercera solución de diseño parcial del sistema de automatización dado por los dos pasos anteriores. $S = S_p^0 \cup S_p^1 \cup S_p^2 \cup S_p^3$;

$$\oplus S = \oplus (S_p^0 \cup S_p^1 \cup S_p^2 \cup S_p^3) = \oplus s_1^3 \cup \oplus s_2^3 \cup \oplus s_3^3 \cup r_{21}^3 \cup r_{13}^3$$

Con el fin de ejemplificar la existencia de una relación, decimos que:

$$(s_3^3 \otimes s_1^3) \cup (s_1^3 \otimes s_3^3) = r_{13}^3$$

$$(s_3^3 \otimes s_2^3) \cup (s_2^3 \otimes s_3^3) = 0$$

De manera similar, podemos completar paso a paso el diseño de todo el sistema:

$$\oplus S = \oplus (S_p^0 \cup S_p^1 \cup S_p^2 \cup S_p^3 \dots \cup S_p^n) = \oplus s_1^1 \cup \oplus s_2^1 \cup \oplus s_3^1 \dots \cup \oplus s_n^x \cup r_{21}^x \cup r_{13}^x \dots \cup r_{nm}^x, \quad (4.18)$$

donde,

$$(s_1^1 \otimes s_2^1) \cup (s_2^1 \otimes s_1^1) = r_{12}^1, \quad (4.19)$$

$$(s_3^2 \otimes s_1^2) \cup (s_1^2 \otimes s_3^2) = r_{13}^2, \quad (4.20)$$

$$\vdots \quad (4.21)$$

$$(s_m^x \otimes s_n^x) \cup (s_n^x \otimes s_m^x) = r_{nm}^x. \quad (4.22)$$

4.4. Implementación de la ecuación

La implementación de la ecuación obtenida en (4.18) y (4.19) del sistema de piso de planta, define los procedimientos, software, de documento y actividades de hardware que se deben realizar como parte la solución del producto de automatización, es por ello que se adiciona a esta metodología la implementación de la técnica de revisión y evaluación de programas de la gestión y diseño de proyectos, permitiendo así culminar el proceso de diseño.

4.4.1. Diagrama PERT

El diagrama PERT es un método que permite analizar las tareas involucradas en el proceso y define tiempos para cada tarea, de ahí que sea útil para el desarrollo de la ecuación

del sistema en piso de planta.

Este diagrama recoge las soluciones parciales definidas para cada una de las capas, se establece el orden entre ellas y el tiempo de ejecución que tomara en la implementación, teniendo en cuenta, que al realizar la sumatoria de estas se logra el producto final de automatización, como se observa en el cuadro 4.2.

Cuadro 4.2: Tabla PERT

ID	Solución de diseño parcial S_p^C	Solución de diseño parcial predecesora S_p^C	Duración de la actividad (días)
A	$S_{Programar\ PLC}^1$	—	5
B	$S_{Realizar\ HMI}^4$	A	4
C	$S_{Documentar}^6$	B	3

Teniendo en cuenta la tabla, se realiza la red de PERT definida por el diseñador, de esta manera se ejecutará de forma ordenada cada una de las soluciones de diseño parciales necesarias para automatizar el sistema de piso de planta como se puede observa en la figura 4.4

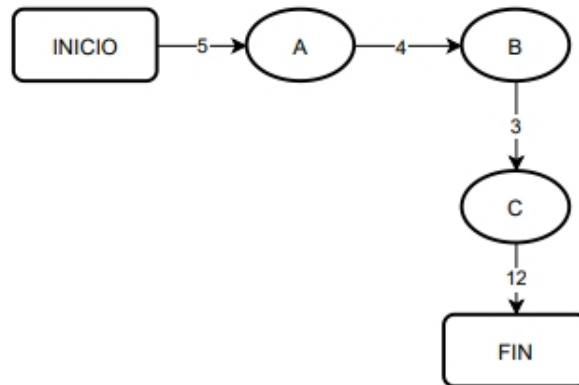


Figura 4.4: Diagrama PERT

En el caso de la figura 4.4 se implementaran 3 soluciones parciales donde se obtendra la programación del PLC, posteriormente se debe programar la pantalla HMI y finalmente se documentará. En cada una de estas descripciones se indica el nivel de abstracción de donde se obtuvo la solución, es decir, la capa definida a partir de la ecuación (??). De esta manera, se obtendra un producto final automatizado según los requerimientos y restricción planteadas por el cliente.

Capítulo 5

Validación de la metodología propuesta

5.1. Planteamiento del problema del caso de estudio

Los responsables de la Cátedra Unesco de Sostenibilidad de la Universidad Politécnica de Cataluña (UPC) desarrollaron la planta piloto de tratamiento de aguas diseñada y utilizada en procesos productivos europeos a través del proyecto LIFE REAGRITECH. Uno de sus principales objetivos es la disminución del consumo de agua para la agricultura y la minimización de los impactos que tienen las zonas vulnerables por nitratos en el medio ambiente.

A partir de este proyecto, se trasladó un piloto en el año 2017 desde Sudanell (Lleida) a Torre Marimon (Caldes de Montbui) en España, para ampliar las condiciones de ensayo y posibles mejoras de automatización del piloto construido, debido a que la tecnología de automatización, a nivel de lógica de control, fue desarrollada e implementada, por encargo, a una empresa privada dedicada a proponer soluciones relacionadas con domótica y automatismos para hogares. Dicha tecnología ha sido desarrollada por la misma empresa, lo que la hace cerrada y poco accesible a la introducción de modificaciones, sin mencionar que la documentación de la misma es escasa o nula. La única información con que se cuenta es un manual de usuario de la interfaz gráfica.

Estas características que si bien, no representan mayor inconveniente en el caso de aplicaciones fijas o definitivas, se convierte en un obstáculo cuando se trata de procesos que exigen flexibilidad, y serán utilizados como herramientas en procesos de investigación en el proyecto AQUARISC, debido a que, al trasladar el piloto construido se presentaron varias problemáticas que impidieron el funcionamiento correcto de la planta, por lo tanto su puesta en marcha presenta cinco situaciones que deberán ser abordadas:

1. La planta se encuentra desmontada y sin funcionar aproximadamente 3 años.
2. Para la puesta en marcha nuevamente en España no se cuenta con planos de construcción (eléctrica, hidráulica, lógica de control, instrumentación, etc).
3. Se desconoce el estado de funcionamiento de las partes y subsistemas de la planta piloto.
4. Se desconoce la existencia de un sistema de tiempo real que responda a las operaciones de la planta piloto y la definición de las variables que influyen en él.

Es por este motivo, que se requiere desarrollar toda la documentación del proceso en cuestión y la puesta en marcha de la planta.

Por otra parte, los miembros del proyecto AQUARISC, tienen definido desarrollar experimentos en la planta piloto. La exigencia en flexibilidad que surge con esta situación conduce a un proceso de sustitución tecnológica a corto plazo, en donde se desea contar con un sistema de automatización que ofrezca prestaciones similares a las que se tenía, pero bajo una implementación abierta, flexible y de bajo costo.

5.1.1. Identificación de necesidades

La primera fase del proceso de diseño se realizó mediante una visita a Torre Marimon, que es un conjunto y finca agrícola de Caldes de Montbui (Barcelona). En el lugar se encuentran ubicados los contenedores artificiales, y allí se llevó a cabo la recolección de información.

Mediante la técnica de observación se identificaron cada uno de los componentes del sistema, lo cual ayudó a estipular actividades fundamentales que deberían realizarse ahí, como el acondicionamiento del lugar y de los instrumentos para que permitiera realizar las posteriores actividades de manera óptima (ver Figura 5.1). Por otro lado, se realizó un registro fotográfico que sirviera de apoyo, teniendo en cuenta que el lugar se encontraba retirado y había que aprovechar toda la información que se pudiera obtener. Finalmente, para lograr entender y reconocer el sistema, mediante el acompañamiento de una investigadora se logró recolectar información de valor y resolver las dudas sobre el funcionamiento del sistema. Este proceso permitió dar inicio a la gestión de documentos definidos por el proceso de diseño, tales como ISA-88, filosofía de funcionamiento del proceso y los diagramas que permitieran plasmar el funcionamiento de la planta y sus necesidades.



(a) Vista interna del proceso.



(b) Vista externa del proceso

Figura 5.1: Inspección de contenedores artificiales.

5.1.2. Requisitos del sistema

De acuerdo con lo planteado en la metodología propuesta, se presentan los requisitos determinados por los interesados en el proyecto, como se observa en la siguiente tabla:

Cuadro 5.1: Requisitos y restricciones del proyecto AQUARISC

Requisitos estructurales (R_e)	Requisitos de rendimiento (R_r)	Requisitos de proceso (R_p)
Gestionar mantenimiento de componentes físicos	Flexibilidad para el manejo de datos e información.	Implementación de las recetas estipuladas para el proceso.
Desarrollar acondicionamiento hidráulico.	Disponibilidad permanente del sistema energético sostenible.	Implementar diagrama UML para los procesos de software.
-	Adquisición de datos remotamente.	Desarrollar los planos de proceso, eléctricos, hidráulicos, lógica de control e instrumentación.
-	-	Puesta en marcha de la planta

5.1.3. Especificaciones de diseño

De acuerdo a las restricciones formuladas anteriormente para este proyecto, se espera cumplir con las siguientes especificaciones de diseño:

Cuadro 5.2: Especificaciones de diseño del proyecto AQUARISC

Especificaciones de diseño ($E^d = \lambda(F_r, [F_r])$)	Restricciones
Documentar el sistema de automatización para los humedales artificiales.	Hacer uso de herramientas software de acceso abierto.
Implementar herramientas software de control y monitoreo de la planta	Hacer uso de herramientas software de acceso abierto. Guardar las características que tenía el antiguo sistema de interfaz web.
Realizar mantenimiento y reparación de la planta	Reutilizar la máxima cantidad posible de los recursos existentes propende al desarrollo sostenible. Mantener la distribución de piso planta a excepción del panel de módulos y potencia

5.2. Representación formal del proceso de la planta piloto

Teniendo en cuenta que la representación de la metodología depende de los niveles de abstracción que encuentren los diseñadores encargados de la automatización, se plantea realizar la estructuración matemática por capas, de este modo se realiza la identificación de los componentes, acciones, descomposición y combinación de los elementos que se consideran importantes en los humedales de la siguiente manera:

5.2.1. Capa 1

- Definir alternativas de solución

Descripción del producto básico $S_a = m_a^s(E_a^d)$

Cuadro 5.3: Definir alternativas de solución capa 1

Sistema de automatización para piso de plata		
Nombre(E_a^d)	Tipo (m_a^s)	Descripción (S_a)
Software(soft)	Sistemas de Control Supervisor y Adquisición de Datos (SCADA)	$\langle S^1_{Soft(1,1,0)}, SCADA \rangle$
Hardware(hardw)	Tablero de automatización industrial (TAI), sensores, actuadores, Transmisores, indicador, transductor	$\langle S^1_{hardw(1,2,0)}, TAI \rangle$
Documentación(Doc)	Física, digital	$\langle S^1_{doc(1,3,0)}, digital \rangle$

- Definir Prestaciones y realizar proceso de evaluación

Desempeño de producto candidato ($P_a = m_a^c(S_a)$), evaluación de especificaciones ($\lambda' = ((F_r, [F_r])) = 1$)

Cuadro 5.4: Definir Prestaciones y realizar proceso de evaluación capa 1

Sistema de automatización para piso de plata			Evaluación de especificaciones
Descripción (S_a)	Conocimiento (m_a^c)	Prestaciones (P_a)	$E_a^d = \lambda'(F_r^a), F_r^a = SaYPa$
<S1 _{Soft} (1,1,0), SCA-DA>	Conjunto de programas y rutinas que permiten a la instrumentación realizar determinadas tareas.	Manejo de datos e información	1
<S1 _{hardw} (1,2,0), TAI>	Elementos que sirven para medir, convertir, transmitir, controlar o registrar variables de un proceso.	Soporte físico del sistema de automatización	1
<S1 _{doc} (1,3,0), digital>	Es el procesamiento de información que otorgará datos específicos sobre la planta piloto.	Registro de información de hardware y software	1

- Definición de nuevas especificaciones basadas en las necesidades

Determinar especificaciones de diseño aumentadas, teniendo en cuenta el comportamiento del sistema de automatización con el entorno $\forall S_a [P_a] \exists m_a^c ((m_a^c = [S_a \cup [P_a], E]_{\approx}) \cap (m_a^c : S_a \rightarrow P_a))$

Cuadro 5.5: Definición de nuevas especificaciones basadas en las necesidades capa1

Sistema de automatización para piso de plata					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a [P_a], E]_{\approx}$)	$(m_a^c : S_a \rightarrow P_a)$	Nuevas especificación (E_a^d) ⁺
<S1 _{Soft} (1,1,0), SCADA>	Software	Manejo de datos e información	La programación de los equipos y la red permite la intervención y monitoreo a nivel remoto de los investigadores.	Facilita la retroalimentación en tiempo real con los dispositivos de campo (sensores y actuadores), además provee de toda la información que se genera en el proceso productivo como supervisión, control calidad, almacenamiento de datos, etc.	Editor WEB, Editor de nodos, Bus de campo (BC).

continúa en la página siguiente

Cuadro 5.5 – continuación de la página anterior					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a[P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S1 _{hardw(1,2,0)} , TAI>	Hardware	Soporte físico del sistema de automatización	El estado de los dispositivos existente en los humedales dependen de los cambios externos presentes (clima, tiempo de inactividad del sistema y gestión de mantenimiento)	El hardware permite medir, convertir, transmitir, controlar o registrar variables de los humedales con el fin de optimizar los recursos utilizados en este.	Estandarizar información de activos físicos en planta, Red Fotovoltaica.
<S1 _{doc(1,3,0)} , digital>	Documentación	Registro de información de hardware y software	Documentos de comunicación técnica del proceso en caso de posibles cambios a nivel de instrumentación y software en los humedales.	El documento debe contener todos los elementos necesarios para configurar el funcionamiento de los humedales por parte de la cátedra UNESCO	Diagrama de potencia (PW), Manual de usuario.

- Definición de Rendimiento y prestaciones de subconjuntos

Determinar prestaciones y subconjuntos de descripciones de productos provisionales $\exists S^{[s]} \in S_a \exists P^{[p]} \in P_a (\lambda(S^{[s]}) \cap (\lambda(P^{[p]}))$

Cuadro 5.6: Descripción de subconjuntos capa 1

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correlacionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
S1 _{Soft(1,1,0)} , SCADA>	Manejo de datos e información	Editor WEB	<S2 _{EditorWeb(2,1,1)} , Soft>	Permite mostrar información operativa en tiempo real, proporcionar gráficos de procesos visuales que aportan significado, contexto al estado de los instrumentos y los parámetros del proceso.

continúa en la página siguiente

Cuadro 5.6 – continuación de la página anterior

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correlacionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
		Editor de lógica control	<S2 _{EditorProgral} (2,2,1), Soft>	
		Editor de nodos	<S2 _{EditorNodos} (2,3,1), Soft>	
		Bus de campo	<S2 _{BusCampo} (2,4,1), Soft>	
<S1 _{hardw} (1,2,0), TAI>	Soporte físico del sistema de automatización	Estandarizar información de activos físicos en planta	<S2 _{Estandar} (2,5,2), Hardw>	Permite conocer el ciclo de vida en el que se encuentran los dispositivos.
		Red fotovoltaica	<S2 _{RedFotovoltaica} (2,6,2), Hardw>	
<S1 _{doc} (1,3,0), digital>	Registro de información de hardware y software	Diagrama de potencia.	<S2 _{DiagramaPW} (2,7,3), Doc>	Documento técnico destinado a dar asistencia donde se identifiquen la instrumentación del proceso así como las instalaciones eléctrica

- Descripción de solución

Solución de diseño final $S = S_a \cup S^{[p]}$

Cuadro 5.7: Descripción solución capa 1

Sistema de automatización para piso de plata		
Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S1 _{Soft} (1,1,0), SCADA>	<S2 _{EditorWeb} (2,1,1), Soft >	<S1 _{Soft} (1,1,0), SCADA>
	<S2 _{EditorProgral} (2,2,1), Soft>	
	<S2 _{EditorNodos} (2,3,1), Soft>	
	<S2 _{BusCampo} (2,4,1), Soft>	
<S1 _{hardw} (1,2,0), TAI>	<S2 _{Estandar} (2,5,2), Hardw>	<S1 _{hardw} (1,2,0), TAI>
	<S2 _{RedFotovoltaica} (2,6,2), Hardw>	
<S1 _{doc} (1,3,0), digital>	<S2 _{DiagramaPW} (2,7,3), Doc>	<S1 _{doc} (1,3,0), digital>
	<S2 _{ManualUsuario} (2,8,3), Doc>	

5.2.1.1. Descomposición en productos básicos de automatización

- Generación de soluciones de diseño primitivas

Cuadro 5.8: Generación de soluciones de diseño básicas capa 1

Sistema de automatización para piso de plata				
Descripción Solución	Especificación de diseño (E_a^d)'	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos ($(E_a^d)'/e_i^{ad}$)	Descomposición básica en otra especificación de diseño ($e_i^{ad}, (E_a^d)'$)
<S1 _{Soft} (1,1,0), SCADA>	Editor Web	Definir plataforma WEB (Definirsoftweb)	Generación de una interfaz web que sea de guía al contexto de los humedales	<S2 _{Definirsoftweb} (2,1,1), Editor Web >
<S1 _{Soft} (1,1,0), SCADA>	Editor de programación (Editor-Progra)	Definir software para la lógica de humedales (Definirsoftcontrol)	Se requiere un software que permita realizar ajustes sobre el comportamiento del sistema	<S2 _{Definirsoftcontrol} (2,2,1), EditorProgra>
<S1 _{Soft} (1,1,0), SCADA>	Editor de nodos	Definir software de programación de adquisición (AcquirirDatos)	Toma de datos físicos para que puedan ser manipulados requieren de una programación existente que permita dicha la conexión	<S2 _{AcquirirDatos} (2,3,1), EditorNodos>
<S1 _{Soft} (1,1,0), SCADA>	Bus de campo	Red de información (Comunicación)	La conexión entre los dispositivos físicos y su información requieren de la definición de un bus de campo	<S2 _{Comunicacin} (2,4,1), BusCampo>
<S1 _{hardw} (1,2,0), TAI>	Estandarizar información de activos físicos en planta (Estándar)	Estructurar los activos físicos de la planta (EstructActivos)	Una inspección del estado de los dispositivos conlleva a tomar decisiones importantes y entre ellas la realización de nuevas compras.	<S2 _{EstructActivos} (2,5,2), Estándar>
<S1 _{hardw} (1,2,0), TAI>	Red fotovoltaica (RedFotovoltaica)	Ejecutar mantenimiento (Mantenimiento)	La energía eléctrica proviene de un sistema fotovoltaico. Realizar restauración a las partes necesarias permite ponerlo en funcionamiento.	<S2 _{Mantenimiento} (2,6,2), RedFotovoltaica>
<S1 _{doc} (1,3,0), digital>	Diagrama de potencia (DiagramaPW)	Elegir de software para realizar el diagrama (DefinirsoftPW)	El diagrama es la representación gráfica de los circuitos del sistema. El software de acceso abierto debe permitir crear circuitos eléctricos de automatización y de control.	<S2 _{DefinirSoftPW} (2,7,3), DiagramaPW>

continúa en la página siguiente

Cuadro 5.8 – continuación de la página anterior

Descripción Solución	Especificación de diseño ($E_a^{d'}$)	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos ($(E_a^{d'})/e_i^{ad}$)	Descomposición básica en otra especificación de diseño ($e_i^{ad}, (E_a^{d'})$)
<S1 _{doc(1,3,0)} , digital>	Manual de usuario (ManualUsuario)	Realizar el manual de programación (docprograma)	Un manual proporciona información técnica escrita como la programación del sistema general.	<S2 _{docprograma(2,8,3)} , ManualUsuario>

- Modelo de acción-reacción

Cuadro 5.9: Modelo de acción-reacción capa 1

Sistema de automatización para piso de plata			
Descripción del producto (S[J])	Acción (A[J])	Respuesta (R[J])	Rendimiento del producto (P[J])
<S _{Soft(1,1,0)} , SCADA>	A[1,1,0]= Programación de funcionamiento	R[1,1,0]= Envío de información	P[1,1,0]= Manejo de datos e información
<S _{hardw(1,2,0)} , TAI>	A[1,2,0]= Recepción de información	R[1,2,0]= Cambio de estado del dispositivo	P[1,2,0]= Soporte físico del sistema de automatización
<S1 _{doc(1,3,0)} , digital>	A[1,3,0]= Adquisición de información	R[1,3,0]= Presentación de información	P[1,3,0]= Registro de información de hardware y software

5.2.2. Capa 2

- Definir alternativas de solución

Descripción del producto básico $S_a = m_a^s(E_a^d)$

Cuadro 5.10: Definir alternativas de solución capa 2

Sistema de automatización para piso de plata		
Nombre (E_a^d)	Tipo (m_a^s)	Descripción (S_a)
Definir plataforma WEB (Definirsoft-web)	ATOM, LOGO WEB EDITOR (LWE), Intellution Dynamics, RSView32.	<S2 _{definirsoftweb(2,1,1)} ,LWE>
Definir software para la lógica de humedales (definirsoftcontrol)	Rslogix 5000, Rslogix 500, LOGOsoft!Comfort(LOGOsoft), studio 5000	<S2 _{definirsoftcontrol(2,2,1)} ,LOGOsoft>
Definir software de programación de adquisición (AquirirDatos)	Node-Red, software de aplicación de adquisición de datos, DAQ	<S2 _{AquirirDatos(2,3,1)} ,Node-red>
Red de información (Comunicación)	Bus de campo	<S2 _{Comunicacin(2,4,1)} , BusCampo>
Estructurar los activos físicos de la planta (EstructActivos)	ISA-88	<S2 _{EstructActivos(2,5,2)} , , ISA-88>
Ejecutar mantenimiento.	Mantenimiento correctivo, mantenimiento preventivo	<S2 _{Mantenimiento(2,6,2)} ,MCorrectivo >
Elegir de software para realizar el diagrama (definirsoftPW)	QElectrotech, Autocad Electrical, Solidwork Electrical, Visio	<S2 _{definirSoftPW(2,7,3)} ,QElectrotech>
Realizar el manual de programación (docprograma)	Latex, Microsoft Office	<S2 _{docprograma(2,8,3)} ,Latex>

- Definir Prestaciones y realizar proceso de evaluación

Desempeño de producto candidato ($P_a = m_a^c(S_a)$), evaluación de especificaciones ($\lambda' = ((F_r, [F_r])) = 1$)

Cuadro 5.11: Definir Prestaciones y realizar proceso de evaluación capa 2

Sistema de automatización para piso de plata			Evaluación de especificaciones
Descripción (S_a)	Conocimiento(m_a^c)	Prestaciones(P_a)	$E_a^d = \lambda'(F_r^a)$
<S2 _{Definirsoftweb(2,1,1)} ,LWE>	Es un editor web gratuito que hace fácil definir y diseñar las páginas web para Smartphone, tabletas y PC.	Visualizar los posibles cambios de estado de la instrumentación.	1

<S2 _{Definirsoftcontrol(2,2,1)} , LOGOsoft>	Es un software que permite la programación para el PLC LOGO, además permite añadir funcionalidades como la operación simple en modo red y la configuración automática de la comunicación con una pantalla en la visualización de red	Permite la implementación de las recetas estipuladas para el proceso	1
<S2 _{AquirirDatos(2,3,1)} , Node-red>	El editor de flujo es una herramienta de programación que se utiliza para conectar dispositivos de hardware	Mapeo de las señales generadas por los equipos de planta.	1
<S2 _{Comunicacin(2,4,1)} , Bus-Campo>	Medio de comunicación que permite el intercambio de datos entre los controladores lógicos programables y los ordenadores.	Permite la interconexión y flexibilidad para enviar y recuperar información.	1
<S2 _{EstructActivos(2,5,2)} , ISA-88>	Explicar de manera detallada y ordenada los equipos y procedimientos presentes en el sistema.	Permite describir equipos y procedimientos del sistema.	1
<S2 _{Mantenimiento(2,6,2)} , MCorrectivo >	Cuando un sistema presenta alguna falla por diversas razones, se tiene que hacer una corrección del sistema (cambio de componentes, piezas, etc)	Permite poner en funcionamiento nuevamente el sistema de red fotovoltaica.	1
<S2 _{definirSoftPW(2,7,3)} , QElectrotech>	Herramienta de representación para las conexiones físicas de los elementos de planta.	Presentar de manera organizada el cableado de los elementos de planta que conforman el sistema.	1
<S2 _{docprograma(2,8,3)} , Latex>	Sistema de composición de texto	Backup documentado que permita tener un respaldo software	1
<S2 _{definirSoftPW(2,7,3)} , QElectrotech>	Herramienta de representación para las conexiones físicas de los elementos de planta.	Presentar de manera organizada el cableado de los elementos de planta que conforman el sistema.	1
<S2 _{docprograma(2,8,3)} , Latex>	Sistema de composición de texto	Backup documentado que permita tener un respaldo software	1

- Definición de nuevas especificaciones basadas en las necesidades

Determinar especificaciones de diseño aumentadas, teniendo en cuenta el comportamiento del sistema de automatización con el entorno $\forall S_a [P_a] \exists m_a^c ((m_a^c = [S_a \cup [P_a], E]_{\approx}) \cap (m_a^c : S_a \rightarrow P_a))$

Cuadro 5.12: Definición de nuevas especificaciones basadas en las necesidades capa 2

Sistema de automatización para piso de plata					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a, P_a, E]_{\approx}$)	Nuevas especificaciones (E_a^d) ⁺	
<S2 _{SCADA} ASUPERV(2.1.1), LWE>	Definir plataforma WEB (Definir software web)	Organización de las páginas web (Organización Web)	Visualizar los posibles cambios de estado de la instrumentación.	Permite tener un manejo automático de manera remota del proceso	LWE es un software gratis y permite visualizar los estados de la instrumentación.
<S2 _{SCADA} CONTROL(2.2.1), Lazo Abierto>	Definir software para la lógica de humedales (Definir software control)	Lenguaje de programación (Lenguaje Progra)	Permite la implementación de las recetas estipuladas para el proceso	La programación en un software libre permite a los investigadores cambiar o aumentar los parámetros al sistema.	LOGOsoft es un software gratis y permite implementar la lógica establecida por los investigadores.
<S2 _{SCADA} ADDDATOS(2.3.1), NodeRed>	Definir software de programación de adquisición (Adquirir Datos)	Bases de datos temporal	Mapeo de las señales generadas por los equipos de planta.	La lectura de datos dependen de los nodos de programación estructurados los cuales deben ser almacenados en una base de datos temporal.	La integración de los datos se logra al conectar dispositivos de hardware y software del humedal.
<S2 _{SCADA} COMUN(2.4.1), Modbus>	Red de información (Comunicación)	Ethernet	Permite la interconexión y flexibilidad para enviar y recuperar información.	La conexión física debe ser coincidente con la interfaz eléctrica del equipo E/S.	Permite la comunicación física entre maestro y esclavos.
<S2 _{Equipos} (2.4.2), ISA-88>	Estructurar los activos físicos de la planta (EstructurActivos)	Tipos de marquilla (TiposMarquillado)	Permite describir equipos y procedimientos del sistema.	Buena práctica de automatización con lenguaje estándar que facilita la interpretación para los investigadores del proyecto.	Presenta los modos para proceder o llevar a cabo ciertas tareas determinadas por los escenarios.
<S2 _{Inst} Diagram(2.5.3), QElectroTech>	Ejecutar mantenimiento.	Estudio de dimensionamiento	Permite poner en funcionamiento nuevamente el sistema de red fotovoltaica.	Permite verificar la capacidad de los paneles con la planta	Presenta el cuadro de cargas con cada uno de los componentes.

continúa en la página siguiente

Cuadro 5.12 – continuación de la página anterior

Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a, P_a, E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S2_ManualU_ser(2.6.3), Latex>	Elegir de software para realizar el diagrama (definir softPW)	Elaborar esquemas gráficos que representen el proceso de forma fácil y estandarizados (estándar)	Presentar de manera organizada el cableado de los elementos de planta que conforman el sistema.	Permite a los investigadores crear y modificar los circuitos eléctricos de los humedales.	Representación de las conexiones físicas de los elementos de planta.
<S2_ManualU_ser(2.6.3), Latex>	Realizar el manual de programación (docprograma)	Documento sencillo y entendible	Backup documentado que permita tener un respaldo software	Documento de comunicación técnica del proceso en caso de cambio de personal.	Brinda información sobre las características de los equipos, y sus variables de medición.

- Definición de Rendimiento y prestaciones de subconjuntos

Determinar prestaciones y subconjuntos de descripciones de productos provisiona-

$$\text{les } \exists S^{[s]} \in S_a \exists P^{[p]} \in P_a (\lambda'(S^{[s]}) \cap (\lambda'(P^{[p]})))$$

Cuadro 5.13: Descripción de subconjuntos capa 2

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correccionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
<S2 _{Definirsoftweb} (2,1,1), LWE>	Visualizar los posibles cambios de estado de la instrumentación.	Organización de las páginas web (OrganWeb)	<S2 _{OrganWeb} (2,2,1), Definirsoftweb>	Muestra y determina la ejecución de las diferentes escenas dadas por los investigadores del proyecto.
<S2 _{Definirsoftcontrol} (2,2,1), LOGOsoft>	Instrucciones con el fin de controlar el comportamiento físico del sistema.	Lenguaje de programación (LenguajeProgra)	<S2 _{LenguajeProgra} (2,2,1), Definirsoftcontrol>	Los lenguajes de programación formal permite organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático
<S2 _{AquirirDatos} (2,3,1), Node-red>	Mapeo aleatorio de las señales generadas por los equipos de planta.	Bases de datos temporal (Bases-DatosTem)	<S2 _{BasesDatosTem} (2,3,1), AquirirDatos>	Almacenar y evaluar datos de sensores o protocolos con marcas temporales durante un período de tiempo determinado.
<S2 _{Comunicacin} (2,4,1), BusCampos>	Permite la interconexión y flexibilidad para enviar y recuperar información.	Ethernet (Ethernet)	<S2 _{Ethernet} (2,4,1), Comunicación>	Manejar grandes cantidades de datos a velocidades de 10 Mbps o 100 Mbps.
<S2 _{EstructActivos} (2,5,2), ISA-88>	Permite describir equipos y procedimientos del sistema.	Tipos de marquilla (TiposMarquillado)	<S2 _{CableadoEquipos} (2,5,2), EstructActivos>	Verificar el cableado del sistema para funcionar adecuadamente según los activos del mismo.
<S2 _{Mantenimiento} (2,6,2), MCorrectivo >	Permite poner en funcionamiento nuevamente el sistema de red fotovoltaica.	Estudio de dimensionamiento (EstudioDimens)	<S2 _{EstudioDimens} (2,6,2), Mantenimiento>	Permite tener un estudio del gasto de cargas energéticas de cada uno de los elementos de planta.
<S2 _{DefinirSoftPW} (2,7,3), QElectrotech>	Presentar de manera organizada el cableado de los elementos de planta que conforman el sistema.	Elaborar esquemas gráficos que representen el proceso de forma fácil y estandarizados (StdDiagProces)	<S2 _{StdDiagProces} (2,7,3), definirSoftPW>	El cableado en el tablero eléctrico de sus elementos asociado permite la puesta en marcha del sistema.
<S2 _{Docprograma} (2,8,3), La-tex>	Backup documentado que permita tener un respaldo software	Documento sencillo y entendible (Docsencillo)	<S2 _{Docsencillo} (2,8,3), docprograma>	Dar claridad del proceso de programación en caso de ser requerido

- Descripción de solución

Solución de diseño final $S = S_a \cup S^{[p]}$

Cuadro 5.14: Definir alternativas de solución capa 2

Sistema de automatización para piso de plata		
Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S2 _{definirsoftweb} (2,1,1),LWE>	<S2 _{OrganWeb} (2,2,1), Definirsoftweb>	<S2 _{definirsoftweb} (2,1,1),LWE>
<S2 _{definirsoftcontrol} (2,2,1), LOGOsoft>	LenguajeProgra(2,2,1), Definirsoftcontrol>	<S2 _{definirsoftcontrol} (2,2,1), LOGOsoft>
<S2 _{AquirirDatos} (2,3,1), Node-red>	<S2 _{BasesDatosTem} (2,3,1), AquirirDatos>	<S2 _{AquirirDatos} (2,3,1), Node-red>
<S2 _{Comunicacin} (2,4,1), BusCampo>	<S2 _{Ethernet} (2,4,1), Comunicación>	<S2 _{Comunicacin} (2,4,1), BusCampo>
<S2 _{EstructActivos} (2,5,2), ISA-88>	<S2 _{CableadoEquipos} (2,5,2), EstructActivos>	<S2 _{EstructActivos} (2,5,2), ISA-88>
<S2 _{Mantenimiento} (2,6,2), MCorrectivo >	<S2 _{EstudioDimens} (2,6,2),Mantenimiento>	<S2 _{Mantenimiento} (2,6,2),MCorrectivo >
<S2 _{definirSoftPW} (2,7,3),QElectrotech>	<S2 _{StdDiagProces} (2,7,3), definirSoftPW>	<S2 _{definirSoftPW} (2,7,3),QElectrotech>
<S2 _{docprograma} (2,8,3),Latex>	<S2 _{Doescencillo} (2,8,3), docprograma>	<S2 _{docprograma} (2,8,3),Latex>

5.2.2.1. Descomposición en productos básicos de automatización

- Generación de soluciones de diseño primitivas

Cuadro 5.15: Generación de soluciones de diseño básicas capa 2

Sistema de automatización para piso de plata				
Descripción Solución	Especificación de diseño (E_a^d)'	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos $((E_a^d)' / e_i^{ad})$	Descomposición básica en otra especificación de diseño $(e_i^{ad}, (E_a^d)')$
<S2 _{Definirsoftweb} (2,1,1),LWE>	Organización de las páginas web (OrganWeb)	Realizar la programación de la página web (ProgramacionWeb)	La navegabilidad en la plataforma web es indispensable para un buen manejo del piso de planta.	<S3 _{ProgramacionWeb} (3,1,1), OrganWeb>
<S2 _{Definirsoftcontrol} (2,2,1), LOGOsoft>	Lenguaje de programación (LenguajeProgra)	Implementar la receta (Implementareceta)	La implementación de la receta o escenas contiene elementos poco entendibles para el lenguaje de programación.	<S3 _{Implementareceta} (3,2,2), LenguajeProgra>
<S2 _{AquirirDatos} (2,3,1), Node-red>	Bases de datos temporal (BasesDatosTem)	Almacenamiento de datos (AlmacenaDatos)	Los datos de series de tiempo son mediciones que se monitorean, recopilan o agregan durante un período de tiempo.	<S3 _{AlmacenaDatos} (3,3,3), BasesDatosTem>
<S2 _{Comunicacin} (2,4,1), BusCampo>	Ethernet (Ethernet)	Elegir Protocolo de red (EleccProtocolo)	Estándar que define la sintaxis, semántica y sincronización de la comunicación, ethernet define las características de cableado y señalización.	<S3 _{EleccProtocolo} (3,4,4), Ethernet>
<S2 _{EstructActivos} (2,5,2), ISA-88>	Tipos de marquilla (TiposMarquillado)	Identificar cableado de instrumentos(IDcable)	El tipo de marquillado depende del tipo de cable para esto la identificación es un trabajo físico que consiste se debe reconocer cada uno de los elementos de planta.	<S3 _{IDCable} (3,5,5), TiposMarquillados>

continúa en la página siguiente

Cuadro 5.15 – continuación de la página anterior

Descripción Solución	Especificación de diseño $(E_a^d)'$	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos $((E_a^d)' / e_i^{ad})$	Descomposición básica en otra especificación de diseño $(e_i^{ad}, (E_a^d)')$
<S2 _{Mantenimiento} (2,6,2), MCorrectivo >	Estudio de dimensionamiento (EstudioDimens)	Realizar conexión a tablero de automatización (ConexTableroRed)	El estudio permite confirmar la capacidad del sistema de red. Realizar la conexión permite alimentar constantemente los diferentes componentes.	<S3 _{ConexTableroRed} (3,6,5), EstudioDimens>
<S2 _{definirSoftPW} (2,7,3), QElectrotech>	Elaborar esquemas gráficos que representen el proceso de forma fácil y estandarizados (StdDiagProces)	Realizar diagramas de instrumentación (DiagramInstru)	Estandarizar un proceso permite que la identificación de los instrumentos sea más fácil mientras que el diagrama de instrumentación sin un estándar puede ser leído de distintas formas o incluso no entenderse.	<S3 _{DiagramInstru} (3,7,6), StdDiagProces>
<S2 _{docprograma} (2,8,3), Latex>	Documento sencillo y entendible (Docsencillo)	Realizar el manual de red (DocRed)	La documentación de programación presenta las reglas de control para que el sistema funcione y la documentación de red data los medios, tecnologías y protocolos.	<S3 _{DocRed} (3,8,7), Docsencillo>

- Modelo de acción-reacción

Cuadro 5.16: Modelo de acción-reacción capa 2

Sistema de automatización para piso de plata			
Descripción del producto (S[])	Acción (A[])	Respuesta (R[])	Rendimiento del producto (P[])
<S2 _{Definirsoftweb} (2,1,1), LWE>	A[2,1,1]= Visualizar las escenas	R[2,1,1]=Cambio de estado del comportamiento	P[2,1,1]=Visualizar los posibles cambios de estado de la instrumentación.
<S2 _{Definirsoftcontrol} (2,2,1), LOGOsoft>	A[2,2,1]=Programar recetas	R[2,2,1]= Controlar el comportamiento del sistema	P[2,2,1]=Instrucciones con el fin de controlar el comportamiento físico del sistema.
<S2 _{AquirirDatos} (2,3,1), Node-red>	A[2,3,1]=Recepción de datos aleatoriamente	R[2,3,1]=Disponibilidad de datos	P[2,3,1]=Mapeo aleatorio de las señales generadas por los equipos de planta.
<S2 _{Comunicacin} (2,4,1), Bus-Campo>	A[2,4,1]=Establecer conexión entre hardware y software	R[2,4,1]=Mantener conexión Hardware-software.	P[2,4,1]=Permite la interconexión y flexibilidad para enviar y recuperar información.
<S2 _{EstructActivos} (2,5,2), ISA-88>	A[2,5,2]=Recolección de información	R[2,5,2]=Disposición de información estandarizada	P[2,5,2]=Permite describir equipos y procedimientos del sistema.
<S2 _{Mantenimiento} (2,6,2), MCorrectivo >	A[2,6,2]=Revisión de sistema de red fotovoltaica	R[2,6,2]=Acondicionamiento de red fotovoltaica	P[2,6,2]=Permite poner en funcionamiento nuevamente el sistema de red fotovoltaica.
<S2 _{definirSoftPW} (2,7,3), QElectrotech>	A[2,7,3]=Modelar diagrama de potencia	R[2,7,3]=Disposición de diagrama de potencia	P[2,7,3]=Presentar de manera organizada el cableado de los elementos de planta que conforman el sistema.

continúa en la página siguiente

Cuadro 5.16 – continuación de la página anterior

Descripción del producto (S[])	Acción (A[])	Respuesta (R[])	Rendimiento del producto (P[])
<S2 _{docprograma(2,8,3)} , Latex>	A[2,8,3]=Adquisición de Información	R[2,8,3]=Presentación de información	P[2,8,3]=Backup documentado que permita tener un respaldo software.

5.2.3. Capa 3

- Definir alternativas de solución

Descripción del producto básico $S_a = m_a^s(E_a^d)$

Cuadro 5.17: Definir alternativas de solución capa 3

Sistema de automatización para piso de plata		
Nombre (E_a^d)	Tipo (m_a^s)	Descripción (S_a)
Realizar la programación de la página web (ProgramacionWeb)	Front end, Back end	<S3 _{ProgramacionWeb(3,1,1)} , Front end>
Implementar la receta (Implementa-Receta)	Diagrama de bloques de funciones (DiagBloq), Ladder, C, Java, Python	<S3 _{ImplementaReceta(3,2,2)} , DiagBloq>
Almacenamiento de datos (AlmacenaDatos)	InfluxDB, Prometeo, TimeScaleDB, Grafito, QuestDB, OpenTSDB	<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>
Elegir Protocolo de red (EleccProtocolo)	Modbus RTU, Modbus TCP/IP, Modbus ASCII, Modbus Plus	<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>
Identificar cableado de instrumentos (IDcable)	Cable con tubo (manguito) termorretráctil, cable con inserto no adhesivo, marcado con etiquetas adhesivas autolaminadas de envolver, etiquetado de fibra óptica.	<S3 _{IDCable(3,5,5)} , Etiqueta Auto-adhesiva>
Realizar conexión a tablero de automatización (ConexTableroRed)	Foto, video	<S3 _{ConexTableroRed(3,6,5)} , Foto>
Realizar diagramas de instrumentación (DiagramInstru)	ISA-5.1	<S3 _{estndar(3,7,6)} , ISA-5.1>
Realizar el manual de red (DocRed)	Latex, Microsoft Office	<S3 _{DocRed(3,8,7)} , Latex>

- Definir Prestaciones y realizar proceso de evaluación

Desempeño de producto candidato ($P_a = m_a^c(S_a)$), evaluación de especificaciones

($\lambda' = ((F_r, [F_r])) = 1$)

Cuadro 5.18: Definir Prestaciones y realizar proceso de evaluación capa 3

Sistema de automatización para piso de plata			Evaluación de especificaciones
Descripción (S_a)	Conocimiento (m_a^c)	Prestaciones (P_a)	$E_a^d = \lambda'(F_r^a)$
<S3 _{ProgramacionWeb(3,1,1)} , Front end>	En esta parte del software es lo que interactúa con el cliente.	Permite a los investigadores navegar por las diferentes páginas mientras lo deseen	1
<S3 _{ImplementaReceta(3,2,2)} , DiagBloq>	Por exigencia de los investigadores se plantea necesario elegir el lenguaje de programación en diagrama de bloques ya que existen UDFs que solo pueden ser creados y manipulados en este tipo de lenguaje.	Lenguaje gráfico que permite construir procedimientos complejos mediante la unión de bloques funcionales conectados por líneas de conexión	1

<S3 _{AlmacenaDatos} (3,3,3), InfluxDB>	InfluxDB está especialmente diseñado para almacenar de manera eficiente una cantidad importante de información, además se pueden definir periodos de retención.	Es un conjunto de valores y mediciones de sensores que se toman durante un período de tiempo para ser mapeadas en tiempo real.	1
<S3 _{EleccProtocolo} (3,4,4), Modbus TCP/IP>	Es necesario el uso de un protocolo compatible con las características del dispositivo LOGO de Siemens el cual se intuye por el uso de su interfaz Logo Web	Permite la conexión del dispositivo E/S y el PC para poder adquirir los datos mediante el protocolo.	1
<S3 _{IDCable} (3,5,5), Etiqueta Autoadhesiva>	Debido a las condiciones iniciales de los humedales el marcado e identificación de cables es una necesidad para proceder con la adecuación de un panel de control.	Es necesario realizar una identificación del cableado de la instrumentación con marcas para su conexión en tablero.	1
<S3 _{ConexTableroRed} (3,6,5), Foto>	La energía renovable debe llegar al panel de control garantizando la red para todos los dispositivos estipulados.	Es necesario tener evidencia de la correcta adquisición de la red fotovoltaica.	1
<S3 _{estndar} (3,7,6), ISA-5.1>	ISA 5.1 es un estándar que define herramientas que permiten entender, interpretar y aplicar la simbología e identificación de la instrumentación.	Emplea un sistema especial de símbolos con el objeto de transmitir de forma más fácil y específica la información.	1
<S3 _{DocRed} (3,8,7), Latex>	Sistema de composición de texto.	Backup documentado que permita tener un respaldo software.	1

- Definición de nuevas especificaciones basadas en las necesidades

Determinar especificaciones de diseño aumentadas, teniendo en cuenta el comportamiento del sistema de automatización con el entorno $\forall S_a[P_a]\exists m_a^c((m_a^c = [S_a \cup [P_a], E]_{\approx}) \cap (m_a^c : S_a \rightarrow P_a))$

Cuadro 5.19: Definición de nuevas especificaciones basadas en las necesidades capa 3

Sistema de automatización para piso de plata					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a[P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S3 _{ProgramacionWeb} (3,1,1), Front end>	Realizar la programación de la página web (ProgramacionWeb)	Montar la página en el PLC	Permite a los investigadores navegar por las diferentes páginas mientras lo deseen	El entorno de programación de las páginas es sencillo por esto se puede realizar un diseño editable.	La página web está diseñada para ser fácilmente utilizable y también atractiva.
<S3 _{ImplementaReceta} (3,2,2), DiagBloq>	Implementar la receta (Implementareceta)	Subir programa al PLC	Lenguaje gráfico que permite construir procedimientos complejos mediante la unión de bloques funcionales conectados por líneas de conexión .	Los investigadores pueden tener una programación visualmente no tan abrumadora y entendible.	La implementación de las recetas en diagrama de bloques permite construir procedimientos complejos en bloques.

continúa en la página siguiente

Cuadro 5.19 – continuación de la página anterior

Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a[P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>	Almacenamiento de datos (AlmacenaDatos)	Visualización de datos métricos	Es un conjunto de valores y mediciones de sensores que se toman durante un período de tiempo para ser mapeadas en tiempo real.	Es una aplicación de acceso libre, entre sus ventajas ofrece gran velocidad de introducción de datos ideal para monitoreo en tiempo real.	InfluxDB almacena sus datos con el fin de enviarlos a una interfaz que los muestre al usuario.
<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>	Elegir Protocolo de red (EleccProtocolo)	Configuración de dispositivo receptor MTU (DisReceptor)	Permite la conexión del dispositivo E/S y el PC para poder adquirir los datos mediante el protocolo.	El PLC logo admite los protocolos industriales de comunicación Modbus TCP/IP y Ethernet lo que hace que el proceso de comunicación sea muy sencillo ya que solo requiere de un cable ethernet.	El protocolo Modbus es el encargado de enviar la información por un canal (Ethernet) hasta su receptor.
<S3 _{IDCable(3,5,5)} , Etiqueta Autoadhesiva>	Identificar cableado de instrumentos (IDcable)	Conexión de tablero con instrumentos (conexiTAI)	Es necesario realizar una identificación del cableado de la instrumentación con marcas para su conexión en tablero.	La identificación de cable permite que los investigadores puedan cambiar los instrumentos con su cable correspondiente al momento de detectar alguna falla en el sistema.	La identificación del cableado con etiquetas adhesivas es un método fácil, económico y muy usado para estos sistemas.
<S3 _{ConexTableroRed(3,6,5)} , Foto>	Realizar conexión a tablero de automatización (ConexTableroRed)	Prueba de conexión	Es necesario tener evidencia de la correcta adquisición de la red fotovoltaica.	Permite evidenciar si el sistema de potencia está disponible para los elementos físicos y para los investigadores.	Las conexiones eléctricas proveen la energía para los actuadores, sensores, sistemas de control y red de comunicación.
<S3 _{estndar(3,7,6)} , ISA-5.1>	Realizar diagramas de instrumentación (DiagramInstru)	Revisión de diagramas	Emplea un sistema especial de símbolos con el objeto de transmitir de forma más fácil y específica la información.	El estándar ISA 5.1 permite una representación de la instrumentación industrial de manera que sea leída fácilmente por los investigadores.	El estándar ISA 5.1 emplea símbolos de manera que se facilite la información, la cual es indispensable en el diseño de sistemas

continúa en la página siguiente

Cuadro 5.19 – continuación de la página anterior

Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a[P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S3 _{DocRed(3,8,7)} , Latex>	Realizar el manual de red (DocRed)	Unificar los manuales en uno solo	Backup documentado que permita tener un respaldo software.	Documento de comunicación técnica del proceso en caso de cambio de personal el cual debe ser explicado de manera fácil, concisa y gráfica para su manejo.	Brinda información sobre las características sobre la red de comunicación y sus variables de medición

- Definición de Rendimiento y prestaciones de subconjuntos

Determinar prestaciones y subconjuntos de descripciones de productos provisionales

$$\text{les } \exists S^{[s]} \in S_a \exists P^{[p]} \in P_a (\chi(S^{[s]}) \cap (\chi(P^{[p]})))$$

Cuadro 5.20: Descripción de subconjuntos capa 3

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correlacionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
<S3 _{ProgramacionWeb(3,1,1)} , Front end>	Permite a los investigadores navegar por las diferentes páginas mientras lo deseen	Montar la página en el PLC (MontarPagina)	<S3 _{MontarPagina(3,1,1)} , ProgramacionWeb>	La visualización del front end se realiza a partir del montaje de esta en el PLC.
<S3 _{ImplementaReceta(3,2,2)} , DiagBloq>	Lenguaje gráfico que permite construir procedimientos complejos mediante la unión de bloques funcionales conectados por líneas de conexión .	Subir programa al PLC (SubirPrograma)	<S3 _{SubirPrograma(3,2,2)} , Implementareceta>	Para el funcionamiento verificar el correcto funcionamiento de la instrumentación es necesario subir la programación de las recetas al PLC
<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>	Es un conjunto de valores y mediciones de sensores que se toman durante un período de tiempo para ser mapeadas en tiempo real.	Visualización de datos métricos (VisualizarDatos)	<S3 _{VisualizarDatos(3,3,3)} , AlmacenaD>	Presentar los datos de monitoreo de una manera más fácil y agradable de usar por los investigadores.
<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>	Permite la conexión del dispositivo E/S y el PC para poder adquirir los datos mediante el protocolo.	Configuración de dispositivo receptor MTU (DisReceptor)	<S3 _{DisReceptor(3,4,4)} , EleccProtocolo>	El protocolo debe prestar la arquitectura cliente/ servidor, y permitir la conexión de los dispositivos, gratuita y fácil de implementar .

continúa en la página siguiente

Cuadro 5.20 – continuación de la página anterior

Sistema de automatización para piso de planta				
Descripción (S_a)	Rendimientos correccionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
<S3 _{IDCable(3,5,5)} , Etiqueta Autoadhesiva>	Es necesario realizar una identificación del cableado de la instrumentación con marcas para su conexión en tablero.	Conexión de tablero con instrumentos (Conexi-CablesTAI)	<S3 _{ConexionCablesTAI(3,5,5)} , IDCable>	Etiqueta de señalización que permite reconocer el inicio y destino de un cable, utiliza nomenclatura técnica.
<S3 _{ConexTableroRed(3,6,5)} , Foto>	Es necesario tener evidencia de la correcta adquisición de la red fotovoltaica.	Prueba de conexión (TestConexionRed)	<S3 _{TestConexionRed(3,5,5)} , ConexTableroRed >	La prueba de funcionamiento del sistema de potencia se evidencia mediante el funcionamiento de todos los componentes en campo.
<S3 _{estndar(3,7,6)} , ISA-5.1>	Emplea un sistema especial de símbolos con el objeto de transmitir de forma más fácil y específica la información.	Revisión de diagramas (Revisar-Diagram)	<S4 _{RevisarDiagram(4,5,6)} , estandar>	Minimizar errores en la aplicación del estándar ISA-5.1 para la entrega al cliente.
<S3 _{DocRed(3,8,7)} , Latex>	Backup documentado que permita tener un respaldo software.	Unificar los manuales en uno solo (UnirManuales)	<S3 _{UnirManuales(3,7,7)} , DocRed>	Entregar un manual de usuario completo de la automatización en piso de planta del proyecto

- Descripción de solución

Solución de diseño final $S = S_a \cup S^{[p]}$

Cuadro 5.21: Definir alternativas de solución capa 3

Sistema de automatización para piso de planta		
Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S3 _{ProgramacionWeb(3,1,1)} , Front end>	<S3 _{MontarPagina(3,1,1)} , ProgramacionWeb>	<S3 _{ProgramacionWeb(3,1,1)} , Front end>
<S3 _{ImplementaReceta(3,2,2)} , Diag-Bloq>	<S3 _{SubirPrograma(3,2,2)} , Implementareceta>	<S3 _{ImplementaReceta(3,2,2)} , Diag-Bloq>
<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>	<S3 _{VisualizarDatos(3,3,3)} , AlmacenaD>	<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>
<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>	<S3 _{DisReceptor(3,4,4)} , EleccProtocolo>	<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>

continúa en la página siguiente

Cuadro 5.21 – continuación de la página anterior

Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S3 _{IDCable(3,5,5)} , Etiqueta Auto-adhesiva>	<S3 _{ConexionCablesTAI(3,5,5)} , ID-Cable>	<S3 _{IDCable(3,5,5)} , Etiqueta Auto-adhesiva>
<S3 _{ConexTableroRed(3,6,5)} , Foto>	<S3 _{TestConexionRed(3,5,5)} , Conex-TableroRed >	<S3 _{ConexTableroRed(3,6,5)} , Foto>
<S3 _{estndar(3,7,6)} , ISA-5.1>	<S4 _{RevisarDiagram(4,5,6)} , estan-dar>	<S3 _{estndar(3,7,6)} , ISA-5.1>
<S3 _{DocRed(3,8,7)} , Latex>	<S3 _{UnirManuales(3,7,7)} , DocRed>	<S3 _{DocRed(3,8,7)} , Latex>

5.2.3.1. Descomposición en productos básicos de automatización

- Generación de soluciones de diseño primitivas

Cuadro 5.22: Generación de soluciones de diseño básicas capa 3

Sistema de automatización para piso de plata				
Descripción Solución	Especificación de diseño (E_a^d)'	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos $((E_a^d)' / e_i^{ad})$	Descomposición básica en otra especificación de diseño $(e_i^{ad}, (E_a^d)')$
<S3 _{ProgramacionWeb(3,1,1)} , Front end>	Montar la página en el PLC (MontarPagina)	Realizar pruebas de supervisión (PruebSuper)	La plataforma web funciona solamente si se tiene conexión ethernet en el PLC LOGO por esto para las pruebas de supervisión se toma como evidencia de funcionamiento del programa	<S4 _{PruebSuper(4,1,1)} , MontarPagina>
<S3 _{ImplementaReceta(3,2,2)} , DiagBloq>	Subir programa al PLC (SubirPrograma)	Realizar pruebas de control (PruebControl)	Subir el programa al PLC permite que la planta funcione según lo planteado por los investigadores, sin embargo, para verificarlo es necesario realizar pruebas de control	<S4 _{PruebControl(4,2,2)} , SubirPrograma>
<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>	Visualización de datos métricos (VisualizarDatos)	Presentar datos de monitoreo (PresentarDat)	Influx, envía los datos para que puedan ser consultados por una herramienta capaz de visualizarla	<S4 _{PresentarDat(4,3,3)} , VisualizarDatos>
<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>	Configuración de dispositivo receptor MTU (DisReceptor)	Instalar sistema operativo (SisOperativo)	El PC que es el dispositivo receptor requiere de un sistema operativo abierto para su funcionamiento	<S4 _{SisOperativo(4,4,4)} , DisReceptor>

continúa en la página siguiente

Cuadro 5.22 – continuación de la página anterior

Descripción Solución	Especificación de diseño $(E_a^d)'$	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos $((E_a^d)' / e_i^{ad})$	Descomposición básica en otra especificación de diseño $(e_i^{ad}, (E_a^d)')$
<S3 _{IDCable(3,5,5)} , Etiqueta Autoadhesiva>	Conexión de tablero con instrumentos (ConexiCablesTAI)	Realizar pruebas de conexión del tablero (TestComponentes)	La conexión de tablero se realiza con el fin de integrar todos los elementos sin embargo se hace necesaria una prueba de funcionamiento que compruebe el funcionamiento de estos.	<S4 _{TestComponentes(4,5,5)} , ConexiCablesTAI>
<S3 _{ConexTableroRed(3,6,5)} , Foto>	Revisión de diagramas (RevisarDiagram)	Realizar diagrama secuencial (EsquemaSecu)	Unir los manuales de programación y red permite que en un solo documento se encuentre la automatización sin embargo es necesario realizar un esquema secuencial que permita el entendimiento de este documento.	<S4 _{EsquemaSecu(4,6,8)} , UnirManuales>

- Modelo de acción-reacción

Cuadro 5.23: Modelo de acción-reacción capa 3

Sistema de automatización para piso de plata			
Descripción del producto (S [])	Acción (A [])	Respuesta (R [])	Rendimiento del producto (P [])
<S3 _{ProgramacionWeb(3,1,1)} , Front end>	A[3,1,1]= Visualizar las escenas de funcionamiento del sistema	R[3,1,1]= Facilidad en el manejo de las escenas	Permite a los investigadores navegar por las diferentes páginas mientras lo deseen
<S3 _{ImplementaReceta(3,2,2)} , DiagBloq>	A[3,2,2]=Lenguaje en bloques funcionales	R[3,2,2]=Elaboración de procedimientos complejos	Lenguaje gráfico que permite construir procedimientos complejos mediante la unión de bloques funcionales conectados por líneas de conexión .
<S3 _{AlmacenaDatos(3,3,3)} , InfluxDB>	A[3,3,3]=Almacenamiento de datos	R[3,3,3]=medición de sensores tomadas en un periodo de tiempo en tiempo real	Es un conjunto de valores y mediciones de sensores que se toman durante un período de tiempo para ser mapeadas en tiempo real.
<S3 _{EleccProtocolo(3,4,4)} , Modbus TCP/IP>	A[3,4,4]=Conectar Dispositivos E/S y PC	R[3,4,4]=Adquisición de datos mediante el protocolo Modbus	Permite la conexión del dispositivo E/S y el PC para poder adquirir los datos mediante el protocolo.
<S3 _{IDCable(3,5,5)} , Etiqueta Autoadhesiva>	A[3,5,5]= Elaborar etiquetas adhesivas	R[3,5,5]=Identificación del cableado de instrumentación con marquilla	Es necesario realizar una identificación del cableado de la instrumentación con marcas para su conexión en tablero.
<S3 _{ConexTableroRed(3,6,5)} , Foto>	A[3,6,6]=Verificar la implementación de la red fotovoltaica	R[3,6,6]= Registro fotográfico de la implementación	Es necesario tener evidencia de la correcta adquisición de la red fotovoltaica.
<S3 _{estndar(3,7,6)} , ISA-5.1>	A[3,7,7]= Realizar diagrama de instrumentación	R[3,7,7]= Estandarizar los diagramas en su simbología	Emplea un sistema especial de símbolos con el objeto de transmitir de forma más fácil y específica la información.
<S3 _{DocRed(3,8,7)} , Latex>	A[3,8,8]=Adquisición de Información	R[3,8,8]=Presentacion de informacion	Backup documentado que permita tener un respaldo software.

5.2.4. Capa 4

- Definir alternativas de solución

Descripción del producto básico $S_a = m_a^s(E_a^d)$

Cuadro 5.24: Definir alternativas de solución capa 4

Sistema de automatización para piso de plata		
Nombre (E_a^d)	Tipo (m_a^s)	Descripción (S_a)
Realizar pruebas de supervisión (PruebSuper)	Registro fotográfico (Foto), registro en video (Vid)	<S4 _{PruebSuper(4,1,1)} ,Foto>
Realizar pruebas de control (Prueb-Super)	Registro fotográfico (Foto), registro en video (Vid)	<S4 _{PruebControl(4,2,2)} ,Foto>
Presentar datos de monitoreo (PresentarDat)	Grafana, kibana, Zabbix, tableau...	<S4 _{PresentarDat(4,3,3)} ,Grafana>
Instalar sistema operativo (SisOperativo)	Linux Ubuntu, Inferno, Plan 9, FreeBSD, ReactOS, FreeDOS, Haiku, illumos.	<S4 _{SisOperativo(4,4,4)} , LinuxUbuntu>
Realizar pruebas de conexión del tablero (TestComponentes)	Registro fotográfico (Foto), registro en video (Vid)	<S4 _{TestComponentes(4,5,5)} ,Foto>
Realizar diagrama secuencial (EsquemaSecu)	UML	<S4 _{EsquemaSecu(4,6,8)} ,UML>

- Definir Prestaciones y realizar proceso de evaluación

Desempeño de producto candidato ($P_a = m_a^c(S_a)$), evaluación de especificaciones

($\lambda' = ((F_r, [F_r])) = 1$)

Cuadro 5.25: Definir Prestaciones y realizar proceso de evaluación capa 4

Sistema de automatización para piso de plata			Evaluación de especificaciones
Descripción(S_a)	Conocimiento(m_a^c)	Prestaciones(P_a)	$E_a^d = \lambda'(F_r^a), F_r^a = S_a Y P_a$
<S4 _{PruebSuper(4,1,1)} , Foto>	Evidencia que proporciona la información necesaria para la confirmar el cumplimiento de supervisión	Verificación de los detalles en supervisión WEB	1
<S4 _{PruebControl(4,2,2)} , Foto>	Evidencia que proporciona la información necesaria para la confirmar el cumplimiento de control	Verificación de los detalles de programación de funcionamiento en las recetas.	1
<S4 _{PresentarDat(4,3,3)} , Grafana>	Grafana es una herramienta de código abierto para el análisis y visualización de métricas.	Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo.	1
<S4 _{SisOperativo(4,4,4)} , LinuxUbuntu>	Linux es un sistema operativo, de código abierto y desarrollado por una comunidad, para computadoras, servidores, mainframes, dispositivos móviles y dispositivos embebidos	Linux sirve para hacer funcionar todo el hardware de un PC, además controlar superordenadores o servidores	1
<S4 _{TestComponentes(4,5,5)} , Foto>	Evidencia que proporciona la información necesaria para la confirmar el cumplimiento de la conexión de tablero	Verificación de los detalles de instalación de tablero de conexión.	1

continúa en la página siguiente

Cuadro 5.25 – continuación de la página anterior

Descripción(S_a)	Conocimiento(m_a^c)	Prestaciones(P_a)	$E_a^d =$ $\lambda'(F_r^a), F_r^a =$ $S_a Y P_a$
<S4 _{EsquemaSecu(4,6,8)} , UML>	Es una representación visual de objetos, estados y procesos dentro de un sistema. Por un lado, el lenguaje de modelado puede servir de modelo de los humedales y garantizar así una arquitectura de información estructurada secuencial	Específica de manera que defina el sistema, para detallar los artefactos para documentar la programación y red.	1

- Definición de nuevas especificaciones basadas en las necesidades

Determinar especificaciones de diseño aumentadas, teniendo en cuenta el comportamiento del sistema de automatización con el entorno $\forall S_a [P_a] \exists m_a^c ((m_a^c = [S_a \cup [P_a], E]_{\approx}) \cap (m_a^c : S_a \rightarrow P_a))$

Cuadro 5.26: Definición de nuevas especificaciones basadas en las necesidades capa 4

Sistema de automatización para piso de plata					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a [P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S4 _{PruebSuper(4,1,1)} , Foto>	Realizar pruebas de supervisión (PruebSuper)	Informe	Verificación de los detalles en supervisión WEB	La verificación de los detalles de supervisión WEB a partir de las fotos permite entender al investigador el funcionamiento de activación, navegabilidad y tareas de ejecución de las escenas planteadas por la UPC.	El registro fotográfico permite realizar la una verificación del software de supervisión del proyecto AQUARISC
<S4 _{PruebControl(4,2,2)} , Foto>	Realizar pruebas de control (PruebSuper)	Informe	Verificación de los detalles de programación de funcionamiento en las retetas.	La verificación de los detalles de la lógica de control a partir de las fotos permite entender al investigador el movimiento de agua que se realiza en los humedales además de tener un respaldo visual en caso de pérdida o daño de la programación.	El registro fotográfico permite realizar la una verificación del software de control del proyecto AQUARISC

continúa en la página siguiente

Cuadro 5.26 – continuación de la página anterior

Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a[P_a], E]_{\approx}$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S4 _{PresentarDat(4,3,3)} , Grafana>	Presentar datos de monitoreo (PresentarDat)	Creación de dashboard de Sensores y actuadores, estados de escenas.	Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo.	Facilitar a los investigadores de la cátedra, monitorear el estado de los dispositivos remotamente.	Los cuadros de mando gráficos permiten visualizar cada una de las categorías del sistema. (CreaDashboard)
<S4 _{SisOperativo(4,4,4)} , LinuxUbuntu>	Sistema operativo (SisOperativo)	Conexión Remota (ConexionRemota)	Linux sirve para hacer funcionar todo el hardware de un PC, además controlar superordenadores o servidores	Ubuntu permite ser personalizado para la configuración de todos los programas requeridos por el sistema. Por lo tanto debe incluir el programa de conexión remota que permite a los investigadores acceder a la información	Mediante sus distribuciones permite adaptarse al sistema operativo más conveniente.
<S4 _{TestComponentes(4,5,5)} , Foto>	Pruebas de funcionamiento (TestComponentes)	Informe	Verificación de los detalles de instalación de tablero de conexión.	La verificación de los detalles de instalación en el tablero de conexión a partir de las fotos permite que para realizar posibles cambios por mal funcionamiento, o un cambio de cableado para que se haga de manera fácil.	El registro fotográfico permite tener una verificación de los detalles de la instalación del tablero.
<S4 _{EsquemaSecu(4,6,8)} , UML>	Esquema secuencial (EsqSecuen)	Definir etapas de la secuencia (DefinSecu)	Específica de manera que defina el sistema, para detallar los artefactos para documentar la programación y red.	El esquema secuencial de la programación y red de los humedales artificiales permite describirlos de forma gráfica y sencilla para el investigador interesado en conocer este sector.	El UML es un esquema que describe el proceso de programación y red del sistema.

- Definición de Rendimiento y prestaciones de subconjuntos

Determinar prestaciones y subconjuntos de descripciones de productos provisionales

$$\exists S^{[s]} \in S_a \exists P^{[p]} \in P_a (\mathcal{X}(S^{[s]}) \cap (\mathcal{X}(P^{[p]})))$$

Cuadro 5.27: Descripción de subconjuntos capa 4

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correccionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
<S4 _{PruebSuper} (4,1,1), Foto>	Verificación de los detalles en supervisión WEB	Informe	<S4 _{Informe} (4,1,1), PruebSuperv>	El documento permite para ilustrar y soportar la entrega de los detalles realizados en la interfaz WEB implementada,
<S4 _{PruebControl} (4,2,2), Foto>	Verificación de los detalles de programación de funcionamiento en las recetas.	Informe	<S4 _{Informe} (4,2,2), PruebControl>	El documento permite para ilustrar y soportar la entrega de los detalles realizados en la programación, así como su optimización de codificación
<S4 _{PresentarDat} (4,3,3), Grafana>	Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo.	Creación de dashboard de Sensores y actuadores, estados de escenas.	<S4 _{CreaDashboard} (4,3,3), PresentarDat>	La clasificación de los tableros permite administrar la información de una manera más organizada. (CreaDashboard)
<S4 _{SisOperativo} (4,4,4), Linux>	Linux sirve para hacer funcionar todo el hardware de un PC, además controlar superordenadores o servidores	Conexión Remota (ConexionRemota)	<S4 _{ConexionRemota} (4,4,4), SistOperativo>	Es la conexión a distancia entre el dos o más equipos, donde uno de ellos permite acceder al otro como si se estuviera trabajando directamente en frente de este.
<S4 _{TestComponentes} (4,5,5), Foto>	Verificación de los detalles de instalación de tablero de conexión.	Informe	<S4 _{Fotografias} (4,5,5), TestComponentes>	Las evidencias fotográficas del tablero permiten mantener un registro que compruebe que el trabajo realizado para la cátedra.
<S4 _{EsquemaSecu} (4,6,8), UML>	Especifica de manera que defina el sistema, para detallar los artefactos para documentar la programación y red.	Definir etapas de la secuencia (DefinSecu)	<S4 _{DefinSecu} (4,6,7), EsqSecuen>	un diagrama secuencial permite tener un gráfico para visualizar, especificar, construir y documentar facilitando el entendimiento secuencial del proceso programación y red

- Descripción de solución

Solución de diseño final $S = S_a \cup S^{[p]}$

Cuadro 5.28: Definir alternativas de solución capa 4

Sistema de automatización para piso de plata		
Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S4 _{PruebSuper} (4,1,1), Foto>	<S4 _{Informe} (4,1,1), PruebSuperv>	<S4 _{PruebSuper} (4,1,1), Foto>
<S4 _{PruebControl} (4,2,2), Foto>	<S4 _{Informe} (4,2,2), PruebControl>	<S4 _{PruebControl} (4,2,2), Foto>
<S4 _{PresentarDat} (4,3,3), Grafana>	<S4 _{CreaDashboard} (4,3,3), PresentarDat>	<S4 _{PresentarDat} (4,3,3), Grafana>
<S4 _{SisOperativo} (4,4,4), LinuxUbuntu>	<S4 _{ConexionRemota} (4,4,4), SistOperativo>	<S4 _{SisOperativo} (4,4,4), LinuxUbuntu>
<S4 _{TestComponentes} (4,5,5), Foto>	<S4 _{Fotografias} (4,5,5), TestComponentes>	<S4 _{TestComponentes} (4,5,5), Foto>
<S4 _{EsquemaSecu} (4,6,8), UML>	<S4 _{DefinSecu} (4,6,7), EsqSecuen>	<S4 _{EsquemaSecu} (4,6,8), UML>

5.2.4.1. Descomposición en productos básicos de automatización

- Generación de soluciones de diseño primitivas

Cuadro 5.29: Generación de soluciones de diseño básicas capa 4

Sistema de automatización para piso de plata				
Descripción Solución	Especificación de diseño (E_a^d)'	Especificación de diseño primitiva e_i^{ad}	Diferencia de conjuntos ($(E_a^d)' / e_i^{ad}$)	Descomposición básica en otra especificación de diseño ($e_i^{ad}, (E_a^d)'$)
<S4 _{CreaDashboard} (4,3,3), Grafana>	Creación de dashboard de Sensores y actuadores, estados de escenas. (CreaDashboard)	Realizar la representación de datos (RepresDat)	La creación de los dashboard se realiza por medio de consultas a la BDT, una vez creadas y organizadas se puede lograr la representación de estos.	<S5 _{RepresDat} (5,3,3), CreaDashboard>
<S4 _{SisOperativo} (4,4,4), LinuxUbuntu>	Conexión Remota (ConexionRemota)	Definir método de Conexión Remota (MConexionRemota)	La conexión remota es primordial para que los investigadores puedan acceder remotamente por ello es importante definirla	<S5 _{MConexionRemota} (5,2,4), SisOperativo>

- Modelo de acción-reacción

Cuadro 5.30: Modelo de acción-reacción capa 4

Sistema de automatización para piso de plata			
Descripción del producto (S [])	Acción (A [])	Respuesta (R [])	Rendimiento del producto (P [])
<S4 _{PruebSuper} (4,1,1), Foto>	A[4,1,1]= Verificar software de supervisión WEB	R[4,1,1]= Registro fotográfico de la interfaz web	Verificación de los detalles en supervisión WEB

continúa en la página siguiente

Cuadro 5.30 – continuación de la página anterior

Descripción del producto (S[])	Acción (A[])	Respuesta (R[])	Rendimiento del producto (P[])
<S4 _{PruebControl} (4,2,2), Foto>	A[4,2,2]=Verificar el la implementación de lazo de control	R[4,2,2]= Registro fotográfico de la programación	Verificación de los detalles de programación de funcionamiento en las recetas.
<S4 _{PresentarDat} (4,3,3), Grafana>	A[4,3,3]=Datos de fuentes múltiples como las bases de datos de series de tiempo	R[4,3,3]=Crear cuadros y gráficos	Permite crear cuadros de mando y gráficos a partir de múltiples fuentes, incluidas bases de datos de series de tiempo
<S4 _{SisOperativo} (4,4,4), Linux>	A[4,4,4]=Instalación del sistema operativo	R[4,4,4]= Linux permite funcionar y controlar hardware de un pc o servidores	Linux sirve para hacer funcionar todo el hardware de un PC, además controlar superordenadores o servidores
<S4 _{TestComponentes} (4,5,5), Foto>	A[4,5,5]= Verificar Conexión de tablero	R[4,5,5]= Registro fotográfico del tablero	Verificación de los detalles de instalación de tablero de conexión
<S4 _{EsquemaSecu} (4,6,8), UML>	A[4,6,8]=Realizar una secuencia de los procesos de red y programación	R[4,6,8]=Diagrama UML de red y programación	Específica de manera que defina el sistema, para detallar los artefactos para documentar la programación y red.

5.2.5. Capa 5

- Definir alternativas de solución

Descripción del producto básico $S_a = m_a^s(E_a^d)$

Cuadro 5.31: Definir alternativas de solución capa 5

Sistema de automatización para piso de plata		
Nombre (E_a^d)	Tipo (m_a^s)	Descripción (S_a)
Realizar la representación de datos (RepresDat)	Gráfico, lista de alerta, texto, tabla, mapa de calor, lista dashboard ...	<S5 _{RepresDat} (5,1,3), Tabla >
Definir método de Conexión Remota (MConexRemota)	Enrutamiento, GoToMyPC, TeamViewer, Escritorio remoto, Anydesk, Enlace por VPN, Creación de servidor remoto.	<S5 _{MConexRemota} (5,2,4), TeamViewer>

- Definir Prestaciones y realizar proceso de evaluación

Desempeño de producto candidato ($P_a = m_a^c(S_a)$), evaluación de especificaciones

($\lambda' = ((F_r, [F_r])) = 1$)

Cuadro 5.32: Definir Prestaciones y realizar proceso de evaluación capa 5

Sistema de automatización para piso de plata			Evaluación de especificaciones
Descripción (S_a)	Conocimiento(m_a^c)	Prestaciones(P_a)	$E_a^d = \lambda'(F_r^a)$

<S5 _{RepresDat(5,1,3)} , Tabla >	Con la representación mediante tabla se visualizará si el componente está accionado o no.	En el tablero se podrán mostrar los estados asociados a los dispositivos	1
<S5 _{MConexRemota(5,2,4)} , TeamViewwe>	Es un software que permite conectarse remotamente y controlar el escritorio.	Compartir y controlar el escritorio de la planta y realizar transferencia de archivos desde cualquier lugar donde se encuentren los investigadores	1

- Definición de nuevas especificaciones basadas en las necesidades

Determinar especificaciones de diseño aumentadas, teniendo en cuenta el comportamiento del sistema de automatización con el entorno $\forall S_a [P_a] \exists m_a^c ((m_a^c = [S_a \cup [P_a], E] \approx) \cap (m_a^c : S_a \rightarrow P_a))$

Cuadro 5.33: Definición de nuevas especificaciones basadas en las necesidades capa 5

Sistema de automatización para piso de plata					
Descripción (S_a)	Nombre (E_a^d)	Prestaciones (P_a)	Conocimiento del entorno ($m_a^c = [S_a [P_a], E] \approx$)	($m_a^c : S_a \rightarrow P_a$)	Nuevas especificación (E_a^d) ⁺
<S5 _{RepresDat(5,1,3)} , Tabla >	Representación de datos (RepresDat)	En el tablero se podrán mostrar los estados asociados a los dispositivos	Con las tablas se puede mostrar el estado de todos los sensores, actuadores, y el estado de las escenas.	El registro en las tablas permite visualizar los datos de manera organizada y fácil de interpretar para ser mostrados remotamente.	Informe
<S5 _{MConexRemota(5,2,4)} , TeamViewwe>	Definir método de Conexión Remota (MConexRemota)	Compartir y controlar el escritorio de la planta y realizar transferencia de archivos desde cualquier lugar donde se encuentren los investigadores	Con la conexión remota se garantiza que los investigadores no tengan que desplazarse hasta la planta para saber cómo está funcionando o para ejecutar cambios.	La conexión remota permite acceder a la planta en cualquier momento lo que proporciona disponibilidad siempre y cuando exista una conexión por internet.	Configuración IP

- Definición de Rendimiento y prestaciones de subconjuntos

Determinar prestaciones y subconjuntos de descripciones de productos provisionales $\exists S^{[s]} \in S_a \exists P^{[p]} \in P_a (\chi(S^{[s]}) \cap (\chi(P^{[p]})))$

Cuadro 5.34: Descripción de subconjuntos capa 5

Sistema de automatización para piso de plata				
Descripción (S_a)	Rendimientos correccionado (P_a)	Nuevas especificaciones de diseño (E_a^d) ⁺	Subconjuntos de descripciones de producto ($S^{[p]}$)	Prestaciones de producto ($P^{[p]}$)
<S5 _{RepresDat(5,1,3)} , Tabla >	En el tablero se podrán mostrar los estados asociados a los dispositivos.	Informe	<S5 _{Informe(5,1,3)} , RepresDat >	El documento permite mostrar el proceso realizado en el monitoreo de datos.
<S5 _{MConexRemota(5,2,4)} , TeamViewwe>	Compartir y controlar el escritorio de la planta y realizar transferencia de archivos desde cualquier lugar donde se encuentren los investigadores	Configurar IP	<S5 _{ConfiguracinIP(5,1,3)} , MConexRemota >	La configuración IP y la contraseña permite que solamente los investigadores puedan acceder al control remoto de la planta.

- Descripción de solución

Solución de diseño final $S = S_a \cup S^{[p]}$

Cuadro 5.35: Definir alternativas de solución capa 5

Sistema de automatización para piso de plata		
Descripción(S_a)	Subconjunto de descripciones de productos ($S^{[P]}$)	Solución
<S5 _{RepresDat(5,1,3)} , Tabla >	<S5 _{Informe(5,1,3)} , RepresDat >	<S5 _{RepresDat(5,1,3)} , Tabla >
<S5 _{MConexRemota(5,2,4)} , TeamViewwe>	<S5 _{ConfiguracinIP(5,1,3)} , MConexRemota >	<S5 _{MConexRemota(5,2,4)} , TeamViewwe>

5.2.5.1. Descomposición en productos básicos de automatización

- Modelo de acción-reacción

Cuadro 5.36: Modelo de acción-reacción capa 5

Sistema de automatización para piso de plata			
Descripción del producto ($S[]$)	Acción ($A[]$)	Respuesta ($R[]$)	Rendimiento del producto ($P[]$)
<S5 _{RepresDat(5,1,3)} , Tabla >	A[5,3,3]=Representar los datos	R[5,3,3]= Mostrar el estado y nombres de los dispositivos.	P[5,1,3]=En el tablero se podrán mostrar los estados asociados a los dispositivos
<S5 _{MConexRemota(5,2,4)} , TeamViewwe>	A[5,4,4]= Establecer conexión remota	R[5,4,4]= Mantener control del dispositivo en planta	P[5,1,3]=Compartir y controlar el escritorio de la planta y realizar transferencia de archivos desde cualquier lugar donde se encuentren los investigadores

5.3. Representación formal del sistema de automatización

A partir de las descripciones obtenidas en el procedimiento anterior a nivel software, hardware y de documentación en la planta piloto de AQUARISC, se realiza la representación gráfica de la siguiente manera:

5.3.1. Jerarquía de componentes complejos a básicos

Como se mencionó en el capítulo anterior la presentación de los componentes se puede realizar a partir de la ecuación 4.1. La figura 5.2 se muestra la relación vertical que tienen

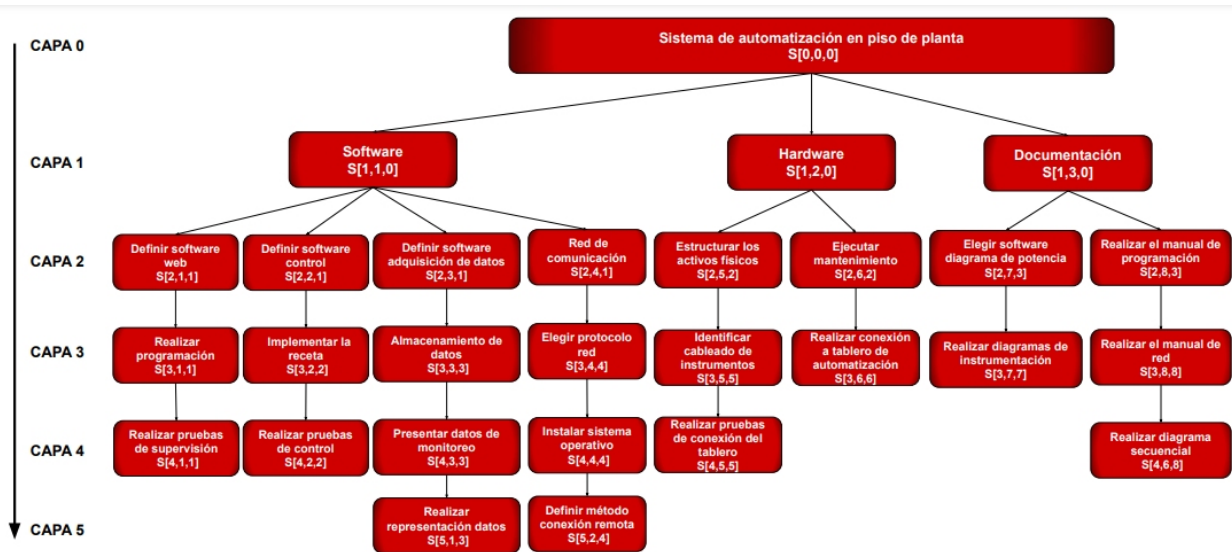


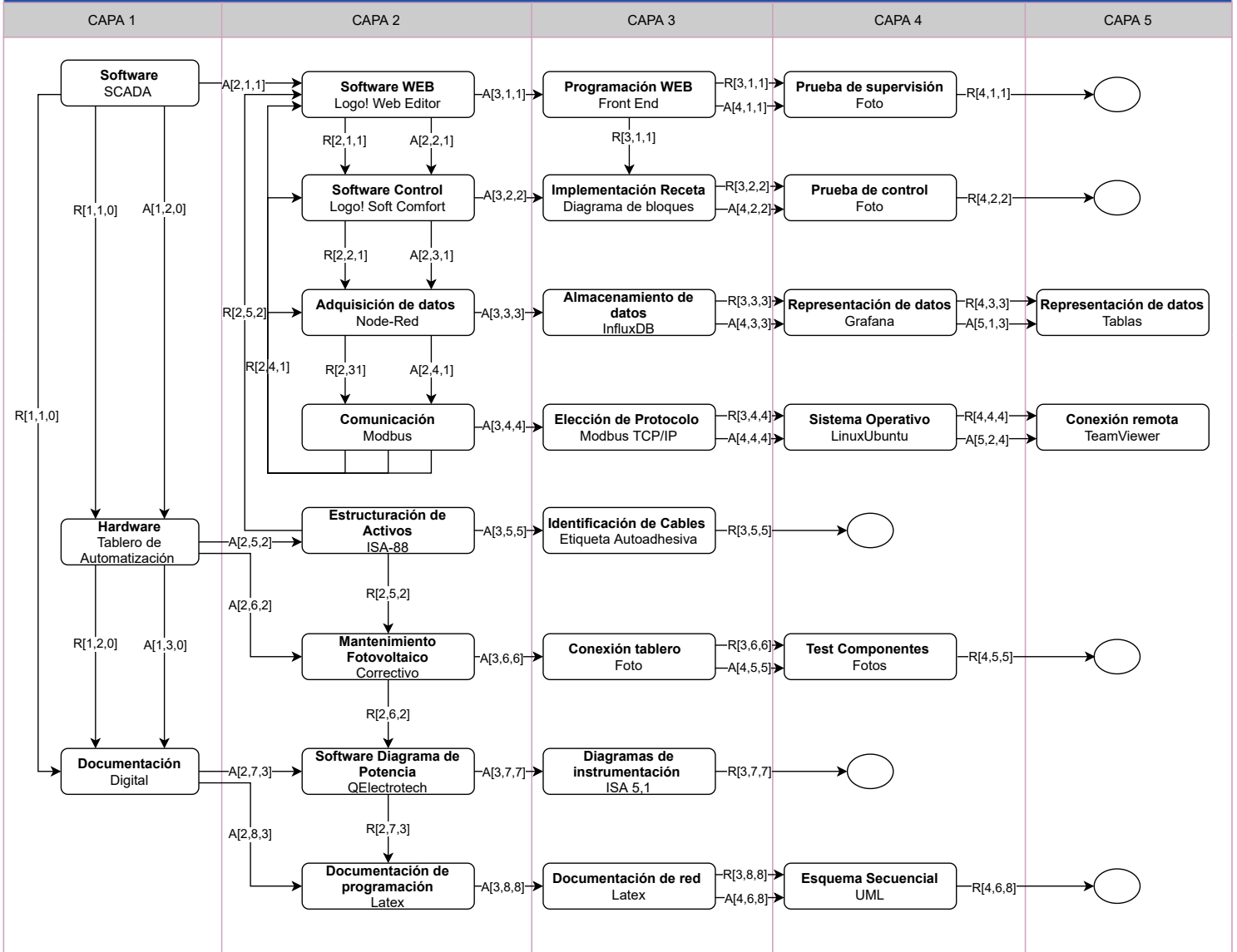
Figura 5.2: Representación del proyecto AQUARISC

cada uno de los componentes identificados para realizar el diseño de automatización a nivel de piso de planta de la planta. Esta figura nos permite fundamentar, verificar y visualizar, de manera fácil y rápida el resultado de las ecuaciones aplicadas hasta este punto.

5.3.2. Comportamiento de los componentes de automatización.

La representación gráfica del comportamiento de los componentes no solo permite determinar las entradas y salidas que tiene el sistema, sino que también establece una relación horizontal de los componentes de la siguiente manera:

Automatización de proyecto AQUARISC Humedales Artificiales



5.4. Combinación de los productos básicos

Recordando que la suma o unión de los productos básicos de automatización que se encontraron a partir de la descripciones de soluciones parciales dadas en la ecuación 4.10 por cada capa, se debe realizar la combinación por niveles de abstracción para formar la solución final, de la siguiente manera:

■ Capa 1

S =Sistema parcial de automatización de piso de planta

S_1 =Capa 1

$s_1^1 = \langle S_1_{Soft(1,1,0)}, SCADA \rangle$

$s_2^1 = \langle S_1_{hardw(1,2,0)}, TAI \rangle$

$s_3^1 = S_1_{doc(1,3,0), digital} \rangle$

El sistema de los humedales artificiales $S=S_0=\emptyset$ es un conjunto vacío.

Paso 1: obtenemos un sistema parcial $S=S_0 \cup S_1$;

$$\oplus S = \oplus (S_0 \cup S_1) = \oplus s_1^1 \cup \oplus s_2^1 \cup \oplus s_3^1 \cup (s_1^1 \otimes s_2^1) \cup (s_2^1 \otimes s_1^1) \cup (s_3^1 \otimes s_1^1) \cup (s_1^1 \otimes s_3^1) \cup (s_2^1 \otimes s_3^1) \cup (s_3^1 \otimes s_2^1)$$

Podemos simplificar tal que :

$$(s_1^1 \otimes s_2^1) \cup (s_2^1 \otimes s_1^1) = r_{12}^1$$

$$(s_3^1 \otimes s_1^1) \cup (s_1^1 \otimes s_3^1) = r_{13}^1$$

$$(s_2^1 \otimes s_3^1) \cup (s_3^1 \otimes s_2^1) = r_{23}^1$$

Obteniendo la siguiente ecuación:

$$\oplus S = \oplus (S_0 \cup S_1) = \oplus s_1^1 \cup \oplus s_2^1 \cup \oplus s_3^1 \cup r_{12}^1 \cup r_{13}^1 \cup r_{23}^1$$

■ Capa 2

S_2 =Capa 2

$S_1^2 = \langle S_2_{Definirsoftweb(2,1,1)}, LWE \rangle$

$S_2^2 = \langle S_2_{Definirsoftcontrol(2,2,1)}, LOGOsoft \rangle$

$S_3^2 = \langle S_2_{AquirirDatos(2,3,1)}, Node-red \rangle$

$S_4^2 = \langle S_2_{Comunicacin(2,4,1)}, BusCampo \rangle$

$S_5^2 = \langle S_2_{InventarioAct(2,5,2)}, ISA-88 \rangle$

$S_6^2 = \langle S_2_{Mantenimiento(2,6,2)}, MCorrectivo \rangle$

$S_7^2 = \langle S_2_{definirSoftPW(2,7,3)}, QElectrotech \rangle$

$S_8^2 = \langle S_2_{docprograma(2,8,3)}, Latex \rangle$.

Paso 2: Obtenemos un segundo sistema parcial

$$\begin{aligned} \oplus S = \oplus(S2) = & \oplus s_1^2 \cup \oplus s_2^2 \cup \oplus s_3^2 \cup \oplus s_4^2 \cup \oplus s_5^2 \cup \oplus s_6^2 \cup \oplus s_7^2 \cup \oplus s_8^2 \cup (s_1^2 \otimes s_2^2) \cup (s_2^2 \otimes s_1^2) \cup (s_3^2 \otimes \\ & s_1^2) \cup (s_1^2 \otimes s_3^2) \cup (s_2^2 \otimes s_3^2) \cup (s_3^2 \otimes s_2^2) \cup (s_1^2 \otimes s_4^2) \cup (s_4^2 \otimes s_1^2) \cup (s_2^2 \otimes s_4^2) \cup (s_4^2 \otimes s_2^2) \cup (s_3^2 \otimes s_4^2) \cup (s_4^2 \otimes \\ & s_3^2) \cup (s_1^2 \otimes s_5^2) \cup (s_5^2 \otimes s_1^2) \cup (s_2^2 \otimes s_5^2) \cup (s_5^2 \otimes s_2^2) \cup (s_3^2 \otimes s_5^2) \cup (s_5^2 \otimes s_3^2) \cup (s_4^2 \otimes s_5^2) \cup (s_5^2 \otimes s_4^2) \cup (s_1^2 \otimes \\ & s_6^2) \cup (s_6^2 \otimes s_1^2) \cup (s_2^2 \otimes s_6^2) \cup (s_6^2 \otimes s_2^2) \cup (s_3^2 \otimes s_6^2) \cup (s_6^2 \otimes s_3^2) \cup (s_4^2 \otimes s_6^2) \cup (s_6^2 \otimes s_4^2) \cup (s_5^2 \otimes s_6^2) \cup (s_6^2 \otimes \\ & s_5^2) \cup (s_1^2 \otimes s_7^2) \cup (s_7^2 \otimes s_1^2) \cup (s_2^2 \otimes s_7^2) \cup (s_7^2 \otimes s_2^2) \cup (s_3^2 \otimes s_7^2) \cup (s_7^2 \otimes s_3^2) \cup (s_4^2 \otimes s_7^2) \cup (s_7^2 \otimes s_4^2) \cup \\ & (s_5^2 \otimes s_7^2) \cup (s_7^2 \otimes s_5^2) \cup (s_6^2 \otimes s_7^2) \cup (s_7^2 \otimes s_6^2) \cup (s_1^2 \otimes s_8^2) \cup (s_8^2 \otimes s_1^2) \cup (s_2^2 \otimes s_8^2) \cup (s_8^2 \otimes s_2^2) \cup (s_3^2 \otimes s_8^2) \cup \\ & (s_8^2 \otimes s_3^2) \cup (s_4^2 \otimes s_8^2) \cup (s_8^2 \otimes s_4^2) \cup (s_5^2 \otimes s_8^2) \cup (s_8^2 \otimes s_5^2) \cup (s_6^2 \otimes s_8^2) \cup (s_8^2 \otimes s_6^2) \cup (s_7^2 \otimes s_8^2) \cup (s_8^2 \otimes s_7^2) \end{aligned}$$

Podemos simplificar tal que :

$$\begin{aligned} (s_1^2 \otimes s_2^2) \cup (s_2^2 \otimes s_1^2) &= r_{12}^2 & (s_5^2 \otimes s_6^2) \cup (s_6^2 \otimes s_5^2) &= r_{56}^2 \\ (s_3^2 \otimes s_1^2) \cup (s_1^2 \otimes s_3^2) &= 0 & (s_1^2 \otimes s_7^2) \cup (s_7^2 \otimes s_1^2) &= 0 \\ (s_2^2 \otimes s_3^2) \cup (s_3^2 \otimes s_2^2) &= r_{23}^2 & (s_7^2 \otimes s_2^2) \cup (s_2^2 \otimes s_7^2) &= 0 \\ (s_1^2 \otimes s_4^2) \cup (s_4^2 \otimes s_1^2) &= r_{41}^2 & (s_3^2 \otimes s_7^2) \cup (s_7^2 \otimes s_3^2) &= 0 \\ (s_2^2 \otimes s_4^2) \cup (s_4^2 \otimes s_2^2) &= r_{42}^2 & (s_4^2 \otimes s_7^2) \cup (s_7^2 \otimes s_4^2) &= 0 \\ (s_3^2 \otimes s_4^2) \cup (s_4^2 \otimes s_3^2) &= r_{43}^2 & (s_5^2 \otimes s_7^2) \cup (s_7^2 \otimes s_5^2) &= 0 \\ (s_1^2 \otimes s_5^2) \cup (s_5^2 \otimes s_1^2) &= r_{51}^2 & (s_6^2 \otimes s_7^2) \cup (s_7^2 \otimes s_6^2) &= r_{67}^2 \\ (s_2^2 \otimes s_5^2) \cup (s_5^2 \otimes s_2^2) &= 0 & (s_1^2 \otimes s_8^2) \cup (s_8^2 \otimes s_1^2) &= 0 \\ (s_3^2 \otimes s_5^2) \cup (s_5^2 \otimes s_3^2) &= 0 & (s_2^2 \otimes s_8^2) \cup (s_8^2 \otimes s_2^2) &= 0 \\ (s_4^2 \otimes s_5^2) \cup (s_5^2 \otimes s_4^2) &= 0 & (s_3^2 \otimes s_8^2) \cup (s_8^2 \otimes s_3^2) &= 0 \\ (s_1^2 \otimes s_6^2) \cup (s_6^2 \otimes s_1^2) &= 0 & (s_4^2 \otimes s_8^2) \cup (s_8^2 \otimes s_4^2) &= 0 \\ (s_2^2 \otimes s_6^2) \cup (s_6^2 \otimes s_2^2) &= 0 & (s_5^2 \otimes s_8^2) \cup (s_8^2 \otimes s_5^2) &= 0 \\ (s_3^2 \otimes s_6^2) \cup (s_6^2 \otimes s_3^2) &= 0 & (s_6^2 \otimes s_8^2) \cup (s_8^2 \otimes s_6^2) &= 0 \\ (s_4^2 \otimes s_6^2) \cup (s_6^2 \otimes s_4^2) &= 0 & (s_7^2 \otimes s_8^2) \cup (s_8^2 \otimes s_7^2) &= r_{78}^2 \end{aligned}$$

Obteniendo la siguiente ecuación:

$$\begin{aligned} \oplus(S) = \oplus(S2) = & \oplus s_1^2 \cup \oplus s_2^2 \cup \oplus s_3^2 \cup \oplus s_4^2 \cup \oplus s_5^2 \cup \oplus s_6^2 \cup \oplus s_7^2 \cup \oplus s_8^2 \cup r_{12}^2 \cup r_{23}^2 \\ & \cup r_{41}^2 \cup r_{42}^2 \cup r_{43}^2 \cup r_{51}^2 \cup r_{56}^2 \cup r_{67}^2 \cup r_{78}^2 \end{aligned}$$

■ Capa 3

S3=Capa 3

$S_1^3 = \langle S3_{ProgramacionWeb(3,1,1)}, \text{Front end} \rangle$

$S_2^3 = \langle S3_{Implementareceta(3,2,2)}, \text{DiagBloq} \rangle$

$S_3^3 = \langle S3_{AlmacenaDatos(3,3,3)}, \text{InfluxDB} \rangle$

$S_4^3 = \langle S3_{EleccProtocolo(3,4,4)}, \text{Modbus TCP/IP} \rangle$

$S_5^3 = \langle S3_{IDCable(3,5,5)}, \text{Etiqueta Autoadhesiva} \rangle$

$S_6^3 = \langle S3_{ConexTableroRed(3,6,5)}, \text{Foto} \rangle$

$S_7^2 = \langle S3_{estndar(3,7,6)}, \text{ISA-5.1} \rangle$

$$S_8^3 = \langle S3_{DocRed(3,8,7)}, \text{Latex} \rangle.$$

Paso 3: Obtenemos un tercer sistema parcial

$$\begin{aligned} \oplus S = \oplus (S3) = & \oplus s_1^3 \cup \oplus s_2^3 \cup \oplus s_3^3 \cup \oplus s_4^3 \cup \oplus s_5^3 \cup \oplus s_6^3 \cup \oplus s_7^3 \cup \oplus s_8^3 \cup (s_1^3 \otimes s_2^3) \cup \\ & (s_2^3 \otimes s_1^3) \cup (s_3^3 \otimes s_1^3) \cup (s_1^3 \otimes s_3^3) \cup (s_2^3 \otimes s_3^3) \cup (s_3^3 \otimes s_2^3) \cup (s_1^3 \otimes s_4^3) \cup (s_4^3 \otimes s_1^3) \cup (s_2^3 \\ & \otimes s_4^3) \cup (s_4^3 \otimes s_2^3) \cup (s_3^3 \otimes s_4^3) \cup (s_4^3 \otimes s_3^3) \cup (s_1^3 \otimes s_5^3) \cup (s_5^3 \otimes s_1^3) \cup (s_2^3 \otimes s_5^3) \cup (s_5^3 \otimes \\ & s_2^3) \cup (s_3^3 \otimes s_5^3) \cup (s_5^3 \otimes s_3^3) \cup (s_4^3 \otimes s_5^3) \cup (s_5^3 \otimes s_4^3) \cup (s_1^3 \otimes s_6^3) \cup (s_6^3 \otimes s_1^3) \cup (s_2^3 \otimes s_6^3) \\ & \cup (s_6^3 \otimes s_2^3) \cup (s_3^3 \otimes s_6^3) \cup (s_6^3 \otimes s_3^3) \cup (s_4^3 \otimes s_6^3) \cup (s_6^3 \otimes s_4^3) \cup (s_5^3 \otimes s_6^3) \cup (s_6^3 \otimes s_5^3) \cup \\ & (s_1^3 \otimes s_7^3) \cup (s_7^3 \otimes s_1^3) \cup (s_7^3 \otimes s_2^3) \cup (s_2^3 \otimes s_7^3) \cup (s_3^3 \otimes s_7^3) \cup (s_7^3 \otimes s_3^3) \cup (s_4^3 \otimes s_7^3) \cup (s_7^3 \\ & \otimes s_4^3) \cup (s_5^3 \otimes s_7^3) \cup (s_7^3 \otimes s_5^3) \cup (s_6^3 \otimes s_7^3) \cup (s_7^3 \otimes s_6^3) \cup (s_1^3 \otimes s_8^3) \cup (s_8^3 \otimes s_1^3) \cup (s_2^3 \otimes \\ & s_8^3) \cup (s_8^3 \otimes s_2^3) \cup (s_3^3 \otimes s_8^3) \cup (s_8^3 \otimes s_3^3) \cup (s_4^3 \otimes s_8^3) \cup (s_8^3 \otimes s_4^3) \cup (s_5^3 \otimes s_8^3) \cup (s_8^3 \otimes s_5^3) \\ & \cup (s_6^3 \otimes s_8^3) \cup (s_8^3 \otimes s_6^3) \cup (s_7^3 \otimes s_8^3) \cup (s_8^3 \otimes s_7^3) \end{aligned}$$

Podemos simplificar tal que :

$$\begin{array}{ll} (s_1^3 \otimes s_2^3) \cup (s_2^3 \otimes s_1^3) = r_{12}^3 & (s_5^3 \otimes s_6^3) \cup (s_6^3 \otimes s_5^3) = 0 \\ (s_3^3 \otimes s_1^3) \cup (s_1^3 \otimes s_3^3) = 0 & (s_1^3 \otimes s_7^3) \cup (s_7^3 \otimes s_1^3) = 0 \\ (s_2^3 \otimes s_3^3) \cup (s_3^3 \otimes s_2^3) = 0 & (s_7^3 \otimes s_2^3) \cup (s_2^3 \otimes s_7^3) = 0 \\ (s_1^3 \otimes s_4^3) \cup (s_4^3 \otimes s_1^3) = 0 & (s_3^3 \otimes s_7^3) \cup (s_7^3 \otimes s_3^3) = 0 \\ (s_2^3 \otimes s_4^3) \cup (s_4^3 \otimes s_2^3) = 0 & (s_4^3 \otimes s_7^3) \cup (s_7^3 \otimes s_4^3) = 0 \\ (s_3^3 \otimes s_4^3) \cup (s_4^3 \otimes s_3^3) = 0 & (s_5^3 \otimes s_7^3) \cup (s_7^3 \otimes s_5^3) = 0 \\ (s_1^3 \otimes s_5^3) \cup (s_5^3 \otimes s_1^3) = 0 & (s_6^3 \otimes s_7^3) \cup (s_7^3 \otimes s_6^3) = 0 \\ (s_2^3 \otimes s_5^3) \cup (s_5^3 \otimes s_2^3) = 0 & (s_1^3 \otimes s_8^3) \cup (s_8^3 \otimes s_1^3) = 0 \\ (s_3^3 \otimes s_5^3) \cup (s_5^3 \otimes s_3^3) = 0 & (s_2^3 \otimes s_8^3) \cup (s_8^3 \otimes s_2^3) = 0 \\ (s_4^3 \otimes s_5^3) \cup (s_5^3 \otimes s_4^3) = 0 & (s_3^3 \otimes s_8^3) \cup (s_8^3 \otimes s_3^3) = 0 \\ (s_1^2 \otimes s_6^3) \cup (s_6^2 \otimes s_1^3) = 0 & (s_4^3 \otimes s_8^3) \cup (s_8^3 \otimes s_4^3) = 0 \\ (s_2^3 \otimes s_6^3) \cup (s_6^3 \otimes s_2^3) = 0 & (s_5^3 \otimes s_8^3) \cup (s_8^3 \otimes s_5^3) = 0 \\ (s_3^3 \otimes s_6^3) \cup (s_6^3 \otimes s_3^3) = 0 & (s_6^3 \otimes s_8^3) \cup (s_8^3 \otimes s_6^3) = 0 \\ (s_4^3 \otimes s_6^3) \cup (s_6^3 \otimes s_4^3) = 0 & (s_7^3 \otimes s_8^3) \cup (s_8^3 \otimes s_7^3) = 0 \end{array}$$

Obteniendo la siguiente ecuación:

$$\oplus S = \oplus (S3) = \oplus s_1^3 \cup \oplus s_2^3 \cup \oplus s_3^3 \cup \oplus s_4^3 \cup \oplus s_5^3 \cup \oplus s_6^3 \cup \oplus s_7^3 \cup \oplus s_8^3 \cup r_{12}^3$$

■ Capa 4

$S4 = \text{Capa 4}$

$$S_1^4 = \langle S4_{PruebSuper(4,1,1)}, \text{Foto} \rangle$$

$$S_2^4 = \langle S4_{PruebControl(4,2,2)}, \text{Foto} \rangle$$

$$S_3^4 = \langle S4_{PresentarDat(4,3,3)}, \text{Grafana} \rangle$$

$$S_4^4 = \langle S4_{SisOperativo(4,4,4)}, LinuxUbuntu \rangle$$

$$S_5^4 = \langle S4_{TestComponentes(4,5,5)}, Foto \rangle$$

$$S_6^4 = \langle S4_{EsquemaSecu(4,6,8)}, UML \rangle$$

Paso 4: Obtenemos un cuarto sistema parcial

$$\begin{aligned} \oplus S = \oplus (S4) = & \oplus s_1^4 \cup \oplus s_2^4 \cup \oplus s_3^4 \cup \oplus s_4^4 \cup \oplus s_5^4 \cup \oplus s_6^4 \cup \oplus s_7^4 \cup (s_1^4 \otimes s_2^4) \cup (s_2^4 \otimes s_1^4) \cup (s_3^4 \otimes s_1^4) \cup \\ & (s_1^4 \otimes s_3^4) \cup (s_2^4 \otimes s_3^4) \cup (s_3^4 \otimes s_2^4) \cup (s_1^4 \otimes s_4^4) \cup (s_4^4 \otimes s_1^4) \cup (s_2^4 \otimes s_4^4) \cup (s_4^4 \otimes s_2^4) \cup (s_3^4 \otimes s_4^4) \cup (s_4^4 \otimes s_3^4) \cup \\ & (s_1^4 \otimes s_5^4) \cup (s_5^4 \otimes s_1^4) \cup (s_2^4 \otimes s_5^4) \cup (s_5^4 \otimes s_2^4) \cup (s_3^4 \otimes s_5^4) \cup (s_5^4 \otimes s_3^4) \cup (s_4^4 \otimes s_5^4) \cup (s_5^4 \otimes s_4^4) \cup (s_1^4 \otimes s_6^4) \cup \\ & (s_6^4 \otimes s_1^4) \cup (s_2^4 \otimes s_6^4) \cup (s_6^4 \otimes s_2^4) \cup (s_3^4 \otimes s_6^4) \cup (s_6^4 \otimes s_3^4) \cup (s_4^4 \otimes s_6^4) \cup (s_6^4 \otimes s_4^4) \cup (s_5^4 \otimes s_6^4) \cup (s_6^4 \otimes s_5^4) \end{aligned}$$

Podemos simplificar tal que :

$$(s_1^4 \otimes s_2^4) \cup (s_2^4 \otimes s_1^4) = 0$$

$$(s_3^4 \otimes s_5^4) \cup (s_5^4 \otimes s_3^4) = 0$$

$$(s_1^4 \otimes s_3^4) \cup (s_3^4 \otimes s_1^4) = 0$$

$$(s_4^4 \otimes s_5^4) \cup (s_5^4 \otimes s_4^4) = 0$$

$$(s_1^4 \otimes s_4^4) \cup (s_4^4 \otimes s_1^4) = 0$$

$$(s_1^4 \otimes s_6^4) \cup (s_6^4 \otimes s_1^4) = 0$$

$$(s_1^4 \otimes s_5^4) \cup (s_5^4 \otimes s_1^4) = 0$$

$$(s_2^4 \otimes s_6^4) \cup (s_6^4 \otimes s_2^4) = 0$$

$$(s_2^4 \otimes s_3^4) \cup (s_3^4 \otimes s_2^4) = 0$$

$$(s_3^4 \otimes s_6^4) \cup (s_6^4 \otimes s_3^4) = 0$$

$$(s_2^4 \otimes s_4^4) \cup (s_4^4 \otimes s_2^4) = 0$$

$$(s_4^4 \otimes s_6^4) \cup (s_6^4 \otimes s_4^4) = 0$$

$$(s_2^4 \otimes s_5^4) \cup (s_5^4 \otimes s_2^4) = 0$$

$$(s_5^4 \otimes s_6^4) \cup (s_6^4 \otimes s_5^4) = 0$$

$$(s_3^4 \otimes s_4^4) \cup (s_4^4 \otimes s_3^4) = 0$$

Obteniendo la siguiente ecuación:

$$\oplus S = \oplus (S4) = \oplus s_1^4 \cup \oplus s_2^4 \cup \oplus s_3^4 \cup \oplus s_4^4 \cup \oplus s_5^4 \cup \oplus s_6^4$$

■ Capa 5

$S_5 = \text{Capa 5}$

$$S_1^5 = \langle S5_{RepresDat(5,1,3)}, Tabla \rangle$$

$$S_2^5 = \langle S5_{MConexRemota(5,2,4)}, TeamViewwer \rangle$$

Paso 5: Obtenemos un quinto sistema parcial

$$\oplus S = \oplus (S5) = \oplus s_1^5 \cup \oplus s_2^5 \cup (s_1^5 \otimes s_2^5) \cup (s_2^5 \otimes s_1^5)$$

Podemos simplificar tal que :

$$(s_1^5 \otimes s_2^5) \cup (s_2^5 \otimes s_1^5) = 0$$

Obteniendo la siguiente ecuación:

$$\oplus S = \oplus (S5) = \oplus s_1^5 \cup \oplus s_2^5$$

5.4.1. Descripción formal del diseño de la planta piloto

La descripción de producto final, se logra con la unión de las soluciones parciales definidas por cada una de las cinco capas. Con esto, se determina que la ecuación final que rige el diseño de migración del sistema de automatización para los humedales artificiales es el siguiente:

$$\begin{aligned} \oplus(\alpha) = \oplus(S) = \oplus(S0 \cup S1 \cup S2 \cup S3 \cup S4 \cup S5) = & \oplus s_1^1 \cup \oplus s_2^1 \cup \oplus s_3^1 \cup \oplus s_1^2 \cup \oplus s_2^2 \cup \oplus s_3^2 \\ & \cup \oplus s_4^2 \cup \oplus s_5^2 \cup \oplus s_6^2 \cup \oplus s_7^2 \cup \oplus s_8^2 \cup \oplus s_1^3 \cup \oplus s_2^3 \cup \oplus s_3^3 \cup \oplus s_4^3 \cup \oplus s_5^3 \cup \oplus s_6^3 \cup \oplus s_7^3 \cup \\ & \oplus s_8^3 \cup \oplus s_1^4 \cup \oplus s_2^4 \cup \oplus s_3^4 \cup \oplus s_4^4 \cup \oplus s_5^4 \cup \oplus s_6^4 \cup \oplus s_1^5 \cup \oplus s_2^5 \cup r_{12}^1 \cup r_{13}^1 \cup r_{23}^1 \cup r_{12}^2 \cup r_{23}^2 \\ & \cup r_{41}^2 \cup r_{42}^2 \cup r_{43}^2 \cup r_{51}^2 \cup r_{56}^2 \cup r_{67}^2 \cup r_{78}^2 \cup r_{12}^3 \end{aligned}$$

5.5. Diagrama PERT

Para ejecutar la automatización de la planta piloto de AQUARISC, se realiza un diagrama de red con el fin de determinar la secuencia que permita el desarrollo de cada una de las capas y las descripciones formuladas por estas. En la figura 5.3. se observa la serie de actividades que se realizan para la implementación de la automatización.

Cuadro 5.37: Tabla de solución de diseños parciales para automatizar

Sistema de automatización para piso de plata			
ID	Solución de diseño parcial	Solución de diseño parcial predecesora	Duración de la actividad (días)
A	S1 _{Soft(1,1,0)}	–	1
B	S1 _{hardw(1,2,0)}	–	1
C	S1 _{doc(1,3,0)}	–	1
D	S2 _{EstructActivos(2,5,2)}	B	2
E	S2 _{definirSoftPW(2,7,3)}	C	2
F	S3 _{estndar(3,7,6)}	E	2
G	S3 _{IDCable(3,5,5)}	D,F	1
H	S2 _{Mantenimiento(2,6,2)}	G	2
I	S3 _{ConexTableroRed(3,6,5)}	H	3
J	S4 _{TestComponentes(4,5,5)}	I	1
K	S2 _{Definirsoftweb(2,1,1)}	A	1
L	S3 _{ProgramacionWeb(3,1,1)}	K	8
M	S2 _{Definirsoftcontrol(2,2,1)}	A	1
N	S3 _{Implementareceta(3,2,2)}	M	8
N	S2 _{AquirirDatos(2,3,1)}	J,L	5
O	S3 _{AlmacenaDatos(3,3,3)}	N	3
P	S4 _{PresentarDat(4,3,3)}	O	2
Q	S5 _{RepresDat(5,1,3)}	P	2
R	S2 _{Comunicacin(2,4,1)}	M	1
S	S3 _{EleccProtocolo(3,4,4)}	R	2
T	S4 _{SisOperativo(4,4,4)}	S	1
U	S4 _{PruebControl(4,2,2)}	R,N	1

continúa en la página siguiente

Cuadro 5.37 – continuación de la página anterior

ID	Solución de diseño parcial	Solución de diseño parcial predecesora	Duración de la actividad (días)
V	S5 $_{MConexRemota(5,2,4)}$	T	1
W	S4 $_{PruebSuper(4,1,1)}$	L,U,V	1
X	S2 $_{docprograma(2,8,3)}$	W	2
Y	S3 $_{DocRed(3,8,7)}$	Q	2
Z	S4 $_{EsquemaSecu(4,6,8)}$	X,Y	1
Total de días			58

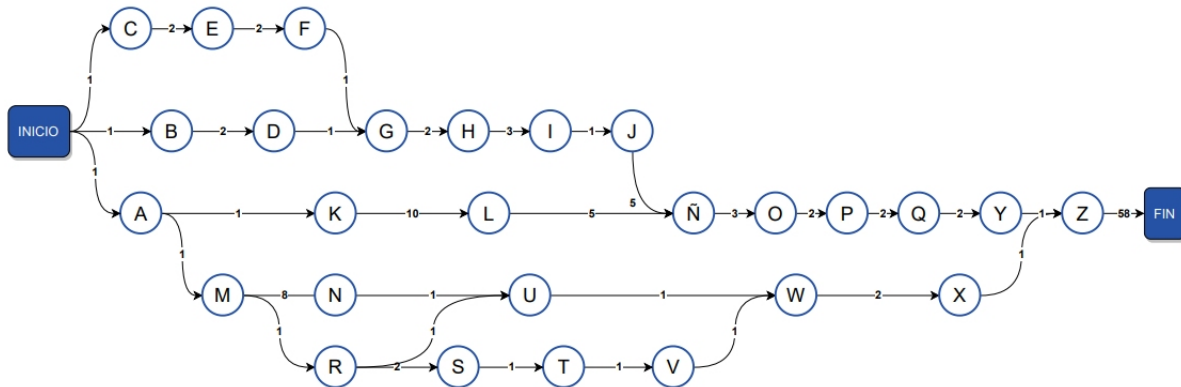


Figura 5.3: Diagrama pert

5.6. Implementación

En la implementación se lleva a cabo el desarrollo del diagrama de red de la siguiente manera, las actividades **A**, **B** y **C**, que corresponden a la primera capa del sistema a desarrollar, es decir es la más general, e inicia con un conjunto vacío donde no se ha realizado ninguna actividad del proceso, pero seguidamente surgen las ideas más grandes sobre el desarrollo, por tanto, se definen los tres componentes más importantes como lo es el software, el hardware y la documentación, en ella se muestra la necesidad de la planta por un sistema SCADA en s_1^1 , un tablero de automatización industrial para la planta piloto de AQUARISC en s_2^1 y realizar la documentación requerida por los investigadores del proyecto para el funcionamiento del piloto en formato digital en s_3^1 . Debido a que el sistema SCADA está relacionado directamente con la instalación del tablero de automatización para funcionar dada en r_{12}^1 , también se considera en r_{13}^1 y r_{23}^1 la necesidad de los componentes anteriores (A y B) para realizar la documentación de forma correcta. De esta manera se daría cumplimiento a la primera capa.

En la actividad **D** se aplica el estándar ISA-88 (s_5^2) de manera que se pueda estructurar los activos existentes en el sistema, así como conocer los procesos y acciones que realiza la instrumentación. Ahora bien, el modelo de receta dada por los investigadores del proyecto

debe ser replicado. En este punto se encuentra que el tablero de automatización industrial es obsoleto y que la programación es realizada a través de lógica cableada. Los clientes del proyecto definieron el uso de un PLC y por tanto se realiza la actividad **E** en donde se define el software ElectroTech (s_7^2) para realizar los planos de potencia como se muestra en la figura 5.4.

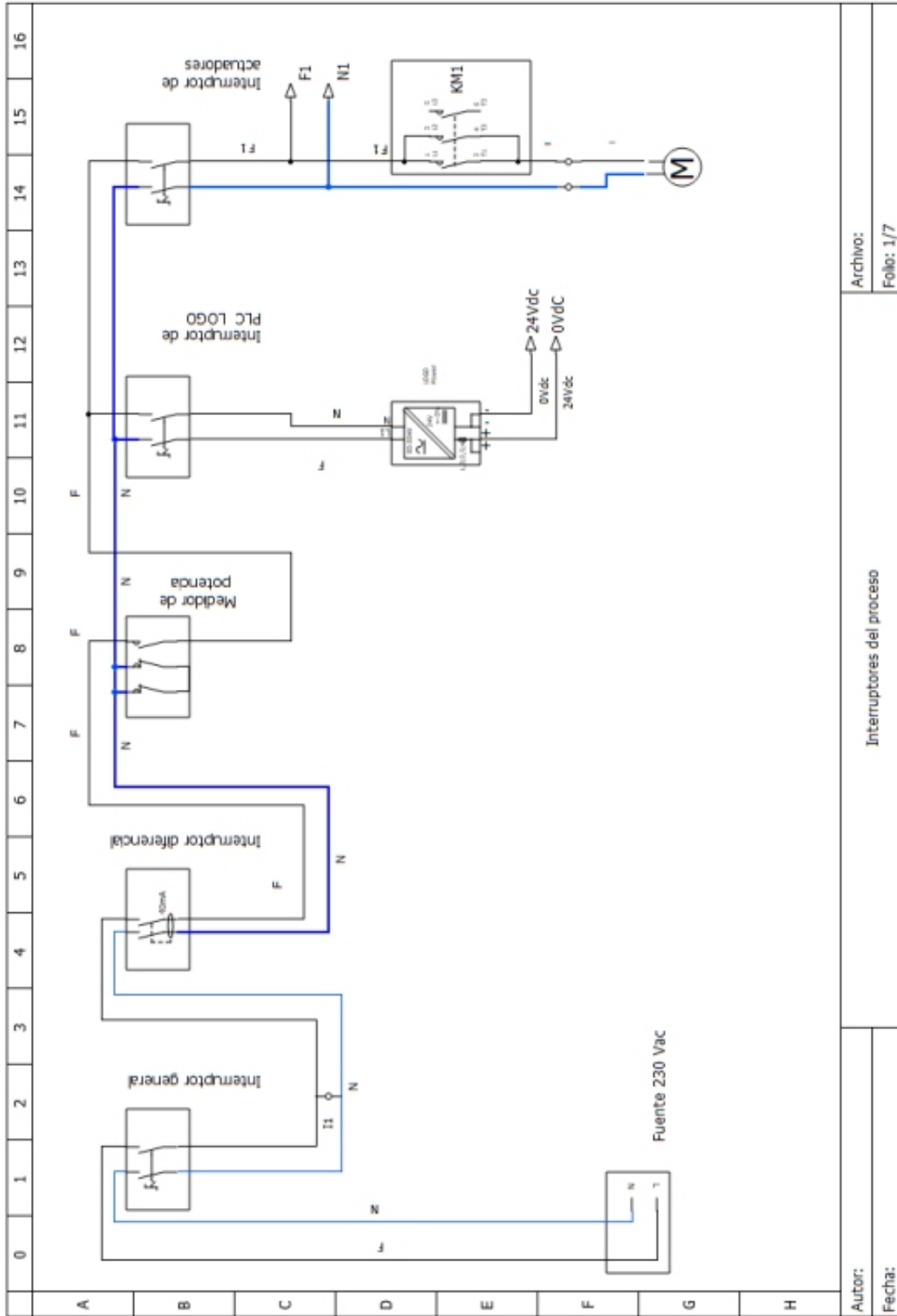


Figura 5.4: Diagrama de potencia

Interrupciones del proceso															
Archivo: 1/7															
Autor:															
Fecha:															

En la actividad **F** (s_7^3) se realiza la documentación de los diagramas de instrumentación a partir de la norma ISA-5.1 donde se identifica posibles lazos de control y el flujo del proceso, en este sentido, se da por finalizada el requisito de desarrollar los planos eléctricos y de instrumentación. Como se muestra en la Figura 5.5

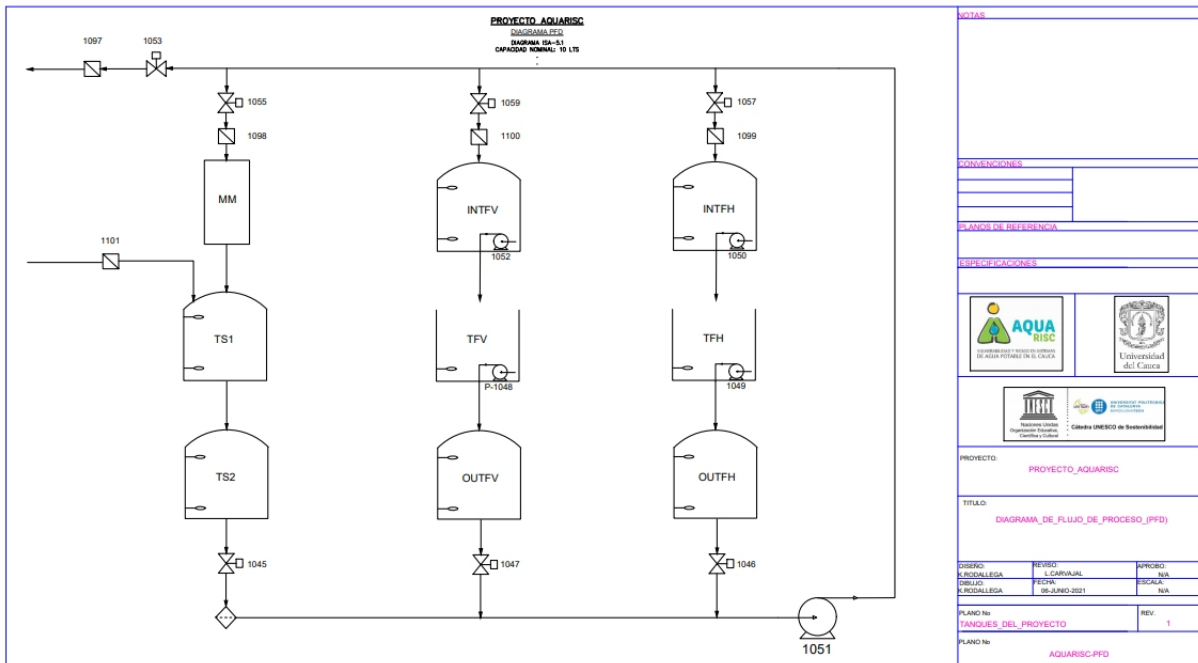


Figura 5.5: PFD AQUARISC

Teniendo los planos de potencia e instrumentación en la actividad **G** (s_5^3) plantea hacer un reconocimiento de cableado de la instrumentación, teniendo en cuenta que las restricciones impuestas son mantener la distribución en piso de planta a excepción del panel de módulos y potencia, además de reutilizar la máxima cantidad posible de los recursos existentes, se decide de esta manera utilizar los mismos cables de los instrumentos para conectarlos e identificarlos con etiquetas adhesivas.

Por otro lado, en la ejecución de **H** (s_6^2) permite el cumplimiento del requisito estructural de gestionar mantenimiento de componentes físicos en donde se realizó el cuadro de cargas para la red fotovoltaica además de encontrar que las baterías existentes no estaban en buen estado, por ende se define el mantenimiento correctivo para el sistema. Posteriormente, al realizar el mantenimiento correctivo y tener los planos eléctricos en la actividad **I** (s_6^3) se realiza la conexión del tablero eléctrico como se evidencia en la Figura 5.6

En la actividad **J** (s_5^4) se instala el tablero de automatización en la planta (Figura 5.7) de manera que la instrumentación pueda ser verificada en su funcionalidad. En este punto, se encontró que una de las válvulas no funcionaba de manera correcta y además, la mo-

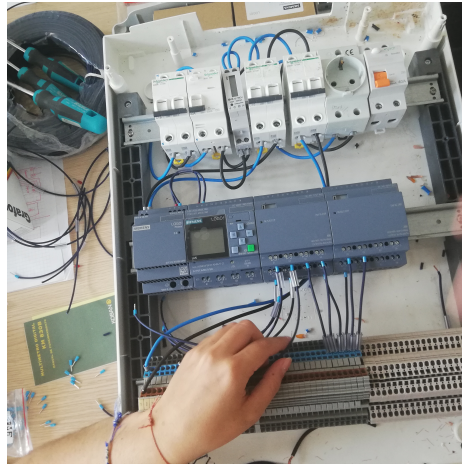


Figura 5.6: Tablero de automatización industrial

tobomba de uno de los tanques estaba dañada por tanto no funcionaba. Sin embargo, se dejó instalado el tablero teniendo en cuenta los espacios correspondientes a estos instrumentos, debido a que por el tiempo de ejecución estimado del proyecto no se alcanzaba a realizar el pedido de la motobomba y el mantenimiento de la válvula.



Figura 5.7: Tablero de automatización industrial en planta

Por otro lado, en \mathbf{K} (s_1^2) se define el software de programación web llamado Logo!Web

editor, debido a las restricciones del cliente donde pide que sean software libres, además de ser un programa que es compatible y fácil de programar con el PLC, el cual permite guardar las características que tenía el antiguo sistema de interfaz web. De esta manera se ejecuta la programación web L (s_1^3) para supervisión del sistema, en donde se ingresan valores de llenado de tanques, tiempos de ejecución, días semanales de ejecución, escenas de circulación de agua, caudal por escena y accionamiento manual.

Del mismo modo, en M (s_2^2) se elige el software de control LOGO soft Comfort, en donde se puede realizar UDF's, los cuales son subrutinas en diagramas de bloques funcionales que permitieron realizar el cálculo del caudal, debido a que los contadores de agua miden un pulso por litro. A partir de esto, se programó las recetas en N (s_2^3) teniendo en cuenta que se debía optimizar el código por medio de subrutinas, además de organizar las entradas y salidas en lugar visible y de fácil entendimiento.

De acuerdo al modelo de datos de Modbus y la memoria variable del PLC LOGO!, se logra el acceso de los datos en Node-Red. Mediante este programa se logra captar la información de los datos del PLC, se ejecuta desde una dirección web para una fácil accesibilidad desde cualquier navegador. Como primera instancia se requiere hacer una petición de datos los cuales fueron, entradas/salidas, análogas/discretas del sistema. Actuando el PLC como servidor y Node-red como cliente. Este es el procedimiento para la actividad \tilde{N} (S_3^2). Figura 5.8.

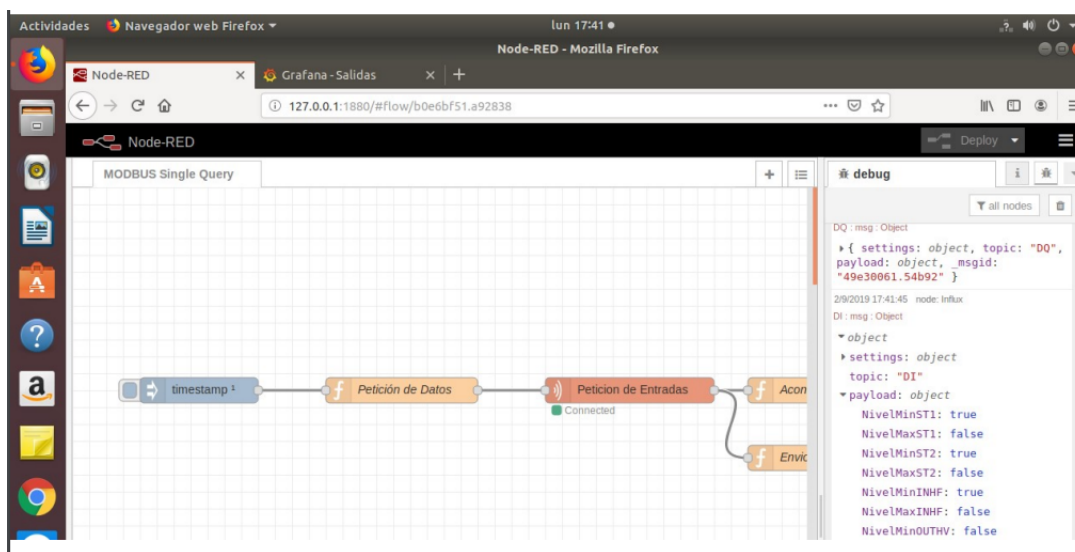


Figura 5.8: Adquisición y acondicionamiento de datos con Node-Red

Posteriormente, se realiza la actividad O (S_3^3), debido a que los datos ya están organizados, se realizó la conexión a influxDB a través del nodo "Debug node", que muestra las

propiedades del mensaje seleccionado (Acondicionamiento de datos) .Se creó una base de datos la cual es llamada para este proyecto "statsdemo", al igual que node-red, esta herramienta se configuró para ser ejecutada automáticamente al encender el equipo, por lo que no requiere ninguna manipulación por parte de los investigadores. Figura 5.8.

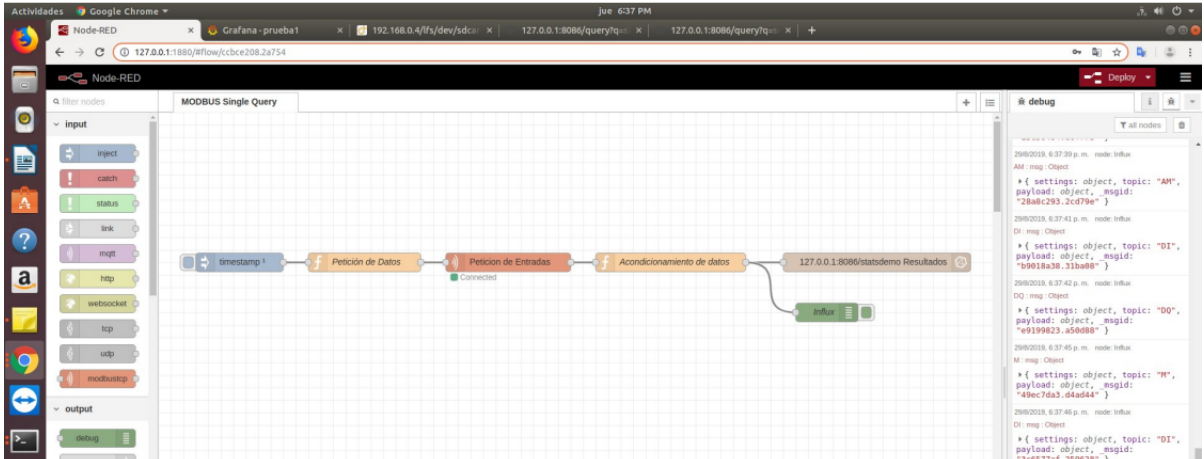


Figura 5.9: Almacenamiento de datos usando nodo InfluxDB

Seguidamente se ejecutó la actividad **P** (S_3^4), para la cual fue necesario la herramienta Grafana, A partir de una serie de datos recolectados por Node-RED e influxDB donde se obtiene un panorama gráfico de la situación del sistema de automatización. Para esta actividad se realizaron diferentes consultas de datos según el tipo de variable, actividad que dio paso a **Q** (S_1^5), la organización de los Dashboard o tableros que mostrarían la información del estado de los dispositivos y las escenas de automatización en tiempo real. Siendo el valor 1, como estado activo y 0, como estado inactivo.

Para la actividad **R** (S_4^2) se estipuló que, en cualquier sistema es necesario establecer el sistema de comunicación, aunque el dispositivo de control se conecta mediante ethernet al cliente, se requirió establecer un protocolo que permitiera la lectura de adquisición de datos y sus actividades posteriores.

Teniendo los software instalados y programados, se realiza la prueba de control en **S** (s_2^4) y posteriormente la prueba de supervisión **T** (s_1^4 , ver figura 5.11) . Donde se pudo observar el funcionamiento correcto de la planta cumpliendo con el requisito de proceso de la puesta en marcha del sistema.

Contando con las formas de comunicación que ofrece el PLC logo, finalmente se estableció el uso del protocolo Modbus TCP/IP. Para esta actividad **U** (S_4^3), se realizó la configuración necesaria. Figura 5.11.

Teniendo en cuenta, las necesidades de los investigadores y del entorno, se instaló el software de acceso libre Ubuntu, y se configuraron allí todas las herramientas nombradas

The dashboard displays two tables of operational data. The first table, 'Resultados.MotobombaGeneral', shows a status of 1.00 for all entries from 2019-08-29 19:52:58 to 19:52:09. The second table, 'Resultados.Electroválvula1045', shows a status of 1.00 for all entries from 2019-08-29 19:52:58 to 19:52:02. Both tables have a 'Time' column and a 'Resultados' column.

Time	Resultados.MotobombaGeneral	Resultados.Motobomba1048
2019-08-29 19:52:58	1.00	0.00
2019-08-29 19:52:51	1.00	0.00
2019-08-29 19:52:44	1.00	0.00
2019-08-29 19:52:37	1.00	0.00
2019-08-29 19:52:30	1.00	0.00
2019-08-29 19:52:23	1.00	0.00
2019-08-29 19:52:16	1.00	0.00
2019-08-29 19:52:09	1.00	0.00

Time	Resultados.Electroválvula1045	Resultados.Electroválvula1046	Resultados.Electroválvula1047
2019-08-29 19:52:58	1.00	1.00	0.00
2019-08-29 19:52:51	1.00	1.00	0.00
2019-08-29 19:52:44	1.00	1.00	0.00
2019-08-29 19:52:37	1.00	1.00	0.00
2019-08-29 19:52:30	1.00	1.00	0.00
2019-08-29 19:52:23	1.00	1.00	0.00
2019-08-29 19:52:16	1.00	1.00	0.00
2019-08-29 19:52:09	1.00	1.00	0.00
2019-08-29 19:52:02	1.00	1.00	0.00

Figura 5.10: Dashboard de estado de Salidas (Motobombas, electroválvulas)

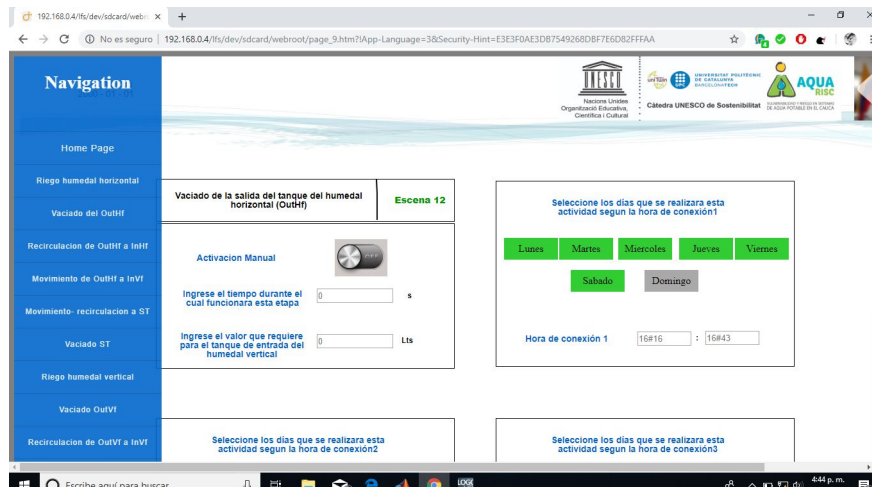


Figura 5.11: Prueba de supervisión

anteriormente, que proporcionan un funcionamiento óptimo a la planta esta es la actividad $V(S_4^4)$. Figura 5.10.

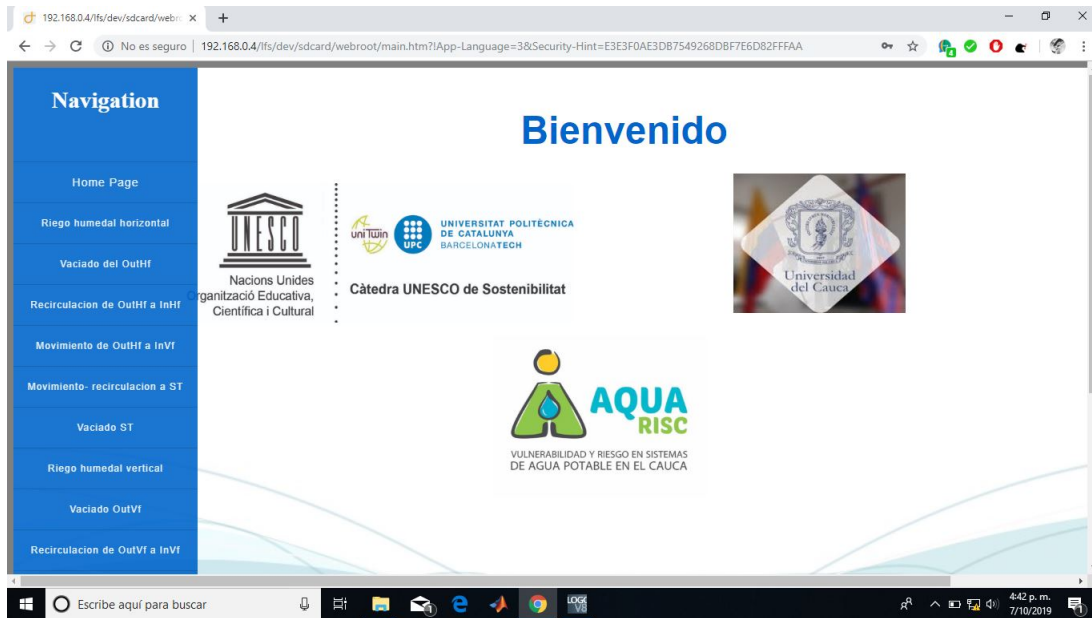


Figura 5.12: Interfaz Ubuntu, con aplicaciones en funcionamiento

Debido a las limitaciones de acceso a la planta, y por ende a su sistema operativo, se lleva a cabo la actividad \mathbf{W} (s_2^5), donde se realizó la instalación temporal Team Viewer, de un escritorio remoto, que brinda a los investigadores la información y manipulación del sistema desde cualquier lugar. Figura 5.12

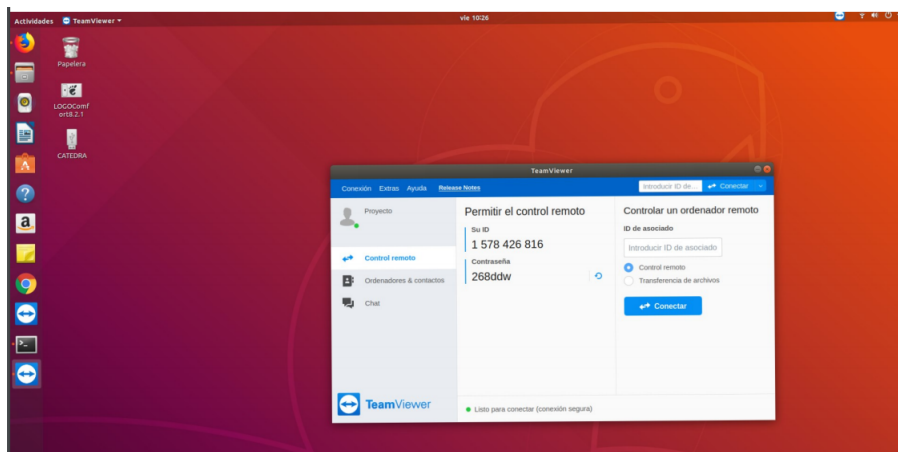


Figura 5.13: Conexión remota via TeamViewer

Para finalizar el diseño de automatización se tiene como requisito la documentación del sistema, por tanto se realiza el manual de usuario con los elementos necesarios para entender la programación (\mathbf{X} (s_8^2)) y posteriormente la documentación de red (\mathbf{Y} (s_8^3)), de esta manera se implementa diagrama UML para los procesos de software \mathbf{Z} . (s_6^4)

Conclusiones y trabajo futuro

Esta metodología se creó para el proceso de diseño o rediseño de sistemas de automatización. Para esto, se utilizaron formalismos matemáticos que respaldan el proceso de diseño. La validación de la metodología se llevó a cabo con la adaptación del producto generado por el diseño axiomático, en la puesta en marcha del sistema de depuración de aguas residuales para el proyecto AQUARISC, a partir de ello se pudo llegar a lo siguiente:

- Al aplicar la metodología propuesta se evidenció que es posible formalizar el proceso de diseño para automatización de sistemas en piso planta, a través de expresiones matemáticas fundamentadas en el diseño conceptual, logrando así mapear de forma fácil y entendible las necesidades del sistema y dar cumplimiento a los requisitos de diseño de automatización en el tiempo establecido.
- En la validación de la metodología se observó que a medida que los sistemas parciales aumentaban de nivel de abstracción, los requerimientos y restricciones impuestas se fueron concretando, asegurando el cumplimiento de los requisitos del sistema de automatización.
- Dentro de la metodología expuesta en el proceso de descomposición, se encuentran los conflictos entre conjuntos de soluciones parciales, asegurando que en caso de presentarse, se deben generar nuevas especificaciones que den solución a estas dificultades. Sin embargo, a raíz de que en el proceso de diseño que se implementó no se tuvieron en cuenta, se considera que es necesario estudiar con más profundidad esta temática a partir de formalismos existentes, enfocados en conflictos entre conjuntos y su adaptación a un sistema de automatización en piso de planta.
- A pesar de que la metodología expuesta demostró ser robusta y flexible en el caso del diseño de sistemas de automatización, su desarrollo es repetitivo y laborioso, por ende se considera como posible extensión o trabajo futuro realizar la optimización mediante una herramienta software, que permita disminuir el tiempo de diseño y aumentar el proceso de análisis y decisiones del mismo.

Bibliografía

- [1] Y. Wang, D. Y. Mo, and M. M. Tseng, "Mapping customer needs to design parameters in the front end of product design by applying deep learning," *CIRP Annals*, vol. 67, no. 1, pp. 145–148, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0007850618300428>
- [2] M. Sawaguchi, S. Ishikawa, and H. Izumi, "Effectiveness of conceptual design process respecting "the axiomatic design theory"," *Procedia Engineering*, vol. 131, pp. 1050–1063, 2015, tRIZ and Knowledge-Based Innovation in Science and Industry. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877705815043143>
- [3] Y. Borgianni and D. T. Matt, "Applications of triz and axiomatic design: A comparison to deduce best practices in industry," *Procedia CIRP*, vol. 39, pp. 91–96, 2016, structured Innovation with TRIZ in Science and Industry: Creating Value for Customers and Society. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827116001864>
- [4] R. jie Lu, Y. xiong Feng, H. Zheng, and J. rong Tan, "A product design based on interaction design and axiomatic design theory," *Procedia CIRP*, vol. 53, pp. 125–129, 2016, the 10th International Conference on Axiomatic Design (ICAD2016). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827116308344>
- [5] Y. Zeng and P. Gu, "A science-based approach to product design theory part i: formulation and formalization of design process," *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 4, pp. 331–339, 1999. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584599000289>
- [6] —, "A science-based approach to product design theory part ii: formulation of design requirements and products," *Robotics and Computer-Integrated Manufacturing*, vol. 15, no. 4, pp. 341–352, 1999. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584599000290>

- [7] A. Mishakova, A. Vakhrushkina, V. Murgul, and T. Sazonova, "Project control based on a mutual application of pert and earned value management methods," *Procedia Engineering*, vol. 165, pp. 1812–1817, 2016, 15th International scientific conference "Underground Urbanisation as a Prerequisite for Sustainable Development" 12-15 September 2016, St. Petersburg, Russia. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877705816342886>
- [8] B. Yan, "A systematic study of design conflict: Modeling, representation, resolution, and application," 2013.
- [9] N. Cross, "Expertise in professional design," 2018.
- [10] N. F. O. Evbuomwan, S. Sivaloganathan, and A. Jebb, "A survey of design philosophies, models, methods and systems," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 210, no. 4, pp. 301–320, 1996.
- [11] Z.-L. Liu, Z. Zhang, and Y. Chen, "A scenario-based approach for requirements management in engineering design," *Concurrent Engineering*, vol. 20, no. 2, pp. 99–109, 2012.
- [12] J. Liu, M. Zhang, and X. Hu, "Dmml: A design thinking process modeling language," in *2011 12th International Conference on Computer-Aided Design and Computer Graphics*, 2011, pp. 44–48.
- [13] G. P. V. Endara and D. M. Z. Armendáriz, "Automatización de procesos de tratamiento de agua y mejora del scada de la planta de tratamiento puengasí epmaps," 2016.
- [14] X. Sun, Y. Zeng, and W. Liu, "Formalization of design chain management using environment-based design (ebd) theory," *Journal of Intelligent Manufacturing*, vol. 24, no. 3, pp. 597–612, 2013.
- [15] M. s. Erden, H. Komoto, T. j. Van beek, V. D'amelio, E. Echavarria, and T. Tomiyama, "A review of function modeling: Approaches and applications," *Ai Edam Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 22, no. 2, pp. 147–169, 2008.
- [16] L. J. Ball, N. J. Lambell, T. C. Ormerod, S. Slavin, and J. A. Mariani, "Representing design rationale to support innovative design reuse: a minimalist approach," *Automation in Construction*, vol. 10, no. 6, pp. 663–674, 2001.

- [17] W. Kunz and H. W. Rittel, *Issues as elements of information systems*. Citeseer, 1970, vol. 131.
- [18] D. Rozo-Ibañez, J. Ruiz-León, D. Guevara-Lozano, and C. R. Vázquez, “Petri net modelling of industrial processes from a pid description,” in *2020 7th International Conference on Control, Decision and Information Technologies (CoDIT)*, vol. 1, 2020, pp. 715–720.
- [19] M. Badell, J. Romero, R. Huertas, and L. Puigjaner, “Planning, scheduling and budgeting value-added chains,” *Computers & Chemical Engineering*, vol. 28, no. 1, pp. 45 – 61, 2004, escape 12. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0098135403001637>
- [20] M. D. Minicis, F. Giordano, F. Poli, and M. M. Schiraldi, “Recipe development process re-design with ansi/isa-88 batch control standard in the pharmaceutical industry;,” *International journal of engineering business management*, vol. 6, p. 16, 2014.
- [21] M. C. Vidoni and A. R. Vecchietti, “An intelligent agent for erp’s data structure analysis based on ansi/isa-95 standard,” *Computers in Industry*, vol. 73, pp. 39 – 50, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0166361515300294>
- [22] I. Harjunkoski, R. Nyström, and A. Horch, “Integration of scheduling and control—theory or practice?” *Computers & Chemical Engineering*, vol. 33, no. 12, pp. 1909 – 1918, 2009, fOCAPO 2008 – Selected Papers from the Fifth International Conference on Foundations of Computer-Aided Process Operations. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0098135409001690>
- [23] G. Polančič, “Bpmn-l: A bpmn extension for modeling of process landscapes,” *Computers in Industry*, vol. 121, p. 103276, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0166361520305108>
- [24] W. Li, Z. Song, C. S. Suh, and E. Mao, “Integration of extenics and axiomatic design theory,” *Procedia Computer Science*, vol. 122, pp. 245–252, 2017, 5th International Conference on Information Technology and Quantitative Management, ITQM 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050917327503>
- [25] J. Thomas and P. Mantri, “Axiomatic design/design patterns mashup: Part 1 (theory),” *Procedia CIRP*, vol. 34, pp. 269–275, 2015, 9th International Conference on Axiomatic Design (ICAD 2015). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827115008501>

- [26] D. Tang, R. Zhu, S. Dai, and G. Zhang, "Enhancing axiomatic design with design structure matrix," *Concurrent Engineering*, vol. 17, no. 2, pp. 129–137, 2009.
- [27] R. Hunter, J. Rios, J. Perez, and A. Vizan, "A functional approach for the formalization of the fixture design process," *International Journal of Machine Tools and Manufacture*, vol. 46, no. 6, pp. 683–697, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0890695505001732>
- [28] J. Shao, F. Lu, C. Zeng, and M. Xu, "Research progress analysis of reliability design method based on axiomatic design theory," *Procedia CIRP*, vol. 53, pp. 107–112, 2016, the 10th International Conference on Axiomatic Design (ICAD2016). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212827116307752>
- [29] L. Tiacci, "Object-oriented event-graph modeling formalism to simulate manufacturing systems in the industry 4.0 era," *Simulation Modelling Practice and Theory*, vol. 99, p. 102027, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1569190X19301583>
- [30] B. von Konsky, M. Robey, and S. Nair, "Integrating design formalisms in software engineering education," in *17th Conference on Software Engineering Education and Training, 2004. Proceedings.*, 2004, pp. 78–83.
- [31] J. C. B. Westermeyer and A. R. Arellano, "Modelo visual para el mapeo y análisis de referentes morfológicos: aplicación educativa en el diseño industrial," *Ingeniare. Revista chilena de ingeniería*, vol. 21, no. 2, pp. 185–195, 2013.
- [32] Y. Zeng, "Axiomatic theory of design modeling," *Journal of Integrated Design & Process Science archive*, vol. 6, no. 3, pp. 1–28, 2002.
- [33] S. Mukherjee and K. Basu, "Solution of interval pert/cpm network problems by a simplified tabular method," *Opsearch*, vol. 48, no. 4, pp. 355–370, 2011.
- [34] M. Hajdu and O. Bokor, "Sensitivity analysis in pert networks: Does activity duration distribution matter?" *Automation in Construction*, vol. 65, pp. 1–8, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0926580516300073>
- [35] K. Sanders, P. V. Farrell, and S. K. Pfatteicher, "Curriculum innovation using job design theory," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 50, no. 6. SAGE Publications Sage CA: Los Angeles, CA, 2006, pp. 779–783.

- [36] C. H. Sánchez-B., “Género y matemáticas. colombianas pioneras en el campo de las matemáticas,” *Revista de la Academia Colombiana de Ciencias Exactas, Físicas y Naturales*, vol. 41, no. 160, pp. 381–392, 2017.