

**SISTEMA DE ACCESO REMOTO A TRAVÉS DE INTERNET A LA
PLANTA SISTEMA A EVENTOS DISCRETOS (SED) DEL LABORATORIO
DE CONTROL DE PROCESOS (LCP) DEL PROGRAMA INGENIERIA EN
AUTOMÁTICA INDUSTRIAL (PIAI)**



**JHESSICA JARAMILLO CAICEDO
VICTOR IVAN JOAQUI TANDIOY**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2013**

**SISTEMA DE ACCESO REMOTO A TRAVÉS DE INTERNET A LA
PLANTA SISTEMA A EVENTOS DISCRETOS (SED) DEL LABORATORIO
DE CONTROL DE PROCESOS (LCP) DEL PROGRAMA INGENIERIA EN
AUTOMÁTICA INDUSTRIAL (PIAI)**



**JHESSICA JARAMILLO CAICEDO
VICTOR IVAN JOAQUI TANDIOY**

Proyecto de grado presentado como requisito para optar al título de
Ingenieros en Automática Industrial

**Director
Ing. Vladimir Trujillo Arias**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
POPAYÁN
2013**

TABLA DE CONTENIDO

RESUMEN.....	1
1. INTRODUCCIÓN	2
1.1. MOTIVACIÓN	2
1.2. OBJETIVOS	3
1.2.1. <i>General</i>	3
1.2.2. <i>Específicos</i>	3
1.3. CONCEPTOS	3
1.3.1. <i>Antecedentes de la tesis</i>	3
1.3.2. <i>Taxonomía</i>	5
1.3.3. <i>Características de un LRR</i>	5
1.3.4. <i>Pasos para la implementación de un LRR</i>	6
1.3.5. <i>Retos actuales de los LRR</i>	7
2. ANÁLISIS DE RECURSOS Y REQUISITOS.....	8
2.1. RECURSOS DEL LCP	8
2.1.1. <i>Recursos Hardware</i>	8
2.1.2. <i>Recursos Software</i>	11
2.2. ANÁLISIS DE LOS REQUISITOS.....	11
2.2.1. <i>Requisitos Hardware y Software</i>	11
2.2.2. <i>Requisitos Funcionales</i>	12
2.2.3. <i>Requisitos no funcionales</i>	14
3. HERRAMIENTAS.....	18
3.1. HERRAMIENTAS PARA ELABORACIÓN DEL SUPERVISOR	18
3.1.1. <i>CRP</i>	18
3.1.2. <i>Grafcet</i>	21
3.1.3. <i>RSLogix 500</i>	22
3.2. HERRAMIENTAS PARA LA SUPERVISIÓN Y COMUNICACIÓN	23
3.2.1. <i>Alternativas</i>	23
3.2.2. <i>Evaluación de las Alternativas Software</i>	28
3.2.3. <i>Selección de la herramienta de supervisión y comunicación</i>	36
3.3. HERRAMIENTAS PARA EL DISEÑO DEL SISTEMA DE ACCESO Y RESERVA. ...	39
3.3.1. <i>Tecnología ASP.NET</i>	39
3.3.2. <i>Entorno de desarrollo</i>	40
3.3.3. <i>Servidores Web</i>	40

3.3.4. Bases de datos.....	40
4. DESARROLLO DEL SISTEMA DE ACCESO REMOTO A TRAVÉS DE INTERNET	42
4.1. ESPECIFICACIONES	42
4.1.1. Modelos de casos de uso	42
4.2. DISEÑO.....	42
4.2.1. Diseño de Entorno Web.....	42
4.3. IMPLEMENTACIÓN	56
4.3.1. Elaboración del supervisor	57
4.3.2. Implementación de LabView.....	65
4.3.3. Arquitectura de la red	84
5. PRUEBAS E INTEGRACIÓN	86
5.1. PRUEBAS DE VALIDACIÓN DEL SUPERVISOR	86
5.2. PRUEBAS PARA EL PANEL REMOTO	86
5.3. PRUEBAS PARA EL SISTEMA DE ACCESO Y RESERVA.....	87
5.4. PRUEBAS DE INTEGRACIÓN	87
5.5. PRUEBAS DE FUNCIONAMIENTO GENERAL.....	88
6. CONCLUSIONES, TRABAJOS FUTUROS Y LIMITACIONES	90
6.1. CONCLUSIONES	90
6.2. TRABAJOS FUTUROS	92
6.3. LIMITACIONES.....	92
REFERENCIAS BIBLIOGRÁFICAS	94

LISTA DE TABLAS

TABLA 1. TAXONOMÍA DE LABORATORIOS.....	5
TABLA 2. PROCEDIMIENTO A EJECUTAR REMOTAMENTE	17
TABLA 3. TAGS DE PRUEBA PARA EL MÓDULO MM&SW	31
TABLA 4. EVALUACIÓN CUANTITATIVA DE LAS ALTERNATIVAS PARA EL PROCESO DE SUPERVISIÓN Y COMUNICACIÓN.	39
TABLA 5. DESCRIPCIÓN DE PUERTOS CONECTADOS AL PLC MICROLOGIX 1500....	57
TABLA 6. ESTADOS Y EVENTOS ASOCIADOS A LA MOTOBOMBA	58
TABLA 7. ESTADOS Y EVENTOS ASOCIADOS A LA ELECTROVÁLVULA DE CARGA DE TK2.....	59
TABLA 8. ESTADOS Y EVENTOS ASOCIADOS A LA TOMA DE DATOS DE TK2.....	60
TABLA 9. ESTADOS Y EVENTOS ASOCIADOS A INICIO 1 PARA TK2	61
TABLA 10. ESTADOS Y EVENTOS ASOCIADOS A INICIO 2 PARA TK2	62
TABLA 11. ESTADOS Y EVENTOS ASOCIADOS A INICIO 3 PARA TK2	62
TABLA 12. TAGS UTILIZADAS EN LA IMPLEMENTACIÓN DEL LLR.....	74

LISTA DE ILUSTRACIONES

FIGURA 1. FUNCIONAMIENTO GENERAL DEL SISTEMA.....	13
FIGURA 2. RELACIÓN DE CONCEPTOS AMEF	15
FIGURA 3. COMPONENTES DE UNA RDP.....	19
FIGURA 4. EJEMPLO DE UNA RDP.....	19
FIGURA 5. TIPOS DE TRANSICIONES EDITADAS EN PIPE EXTENDIDO	19
FIGURA 6. INTERFAZ INICIAL DE CRP	20
FIGURA 7. INTERFAZ PIPE INTEGRADA EN CRP V2.2	21
FIGURA 8. COMPONENTES DE UN GRAFCET	22
FIGURA 9. TRADUCCIÓN DE GRAFCET A LADDER.....	22
FIGURA 10. DIAGRAMA GENERAL DEL MÓDULO DE MONITOREO Y SUPERVISIÓN WEB.	25
FIGURA 11. ESQUEMA DE FUNCIONAMIENTO DE LABVIEW PARA LA INTERACCIÓN USUARIO – RECURSO REMOTO.....	28
FIGURA 12. CARGA DE TAGS AL SERVIDOR OPC KEEPSERVEREX.....	30
FIGURA 13. PRUEBA DE COMUNICACIÓN CLIENTE – SERVIDOR OPC.....	31
FIGURA 14. INTERFAZ DE ACCESO PARA EL DISEÑO DE HMI.....	32
FIGURA 15. INTERFAZ DE USUARIO PARA EL ADMINISTRADOR.....	32
FIGURA 16. CREACIÓN DE UN NUEVO PROYECTO HMI.....	32
FIGURA 17. CONFIGURACIÓN DE LA FUENTE DE DATOS PARA LA INTERFAZ HMI.	33
FIGURA 18. DISEÑO DE LA INTERFAZ DE USUARIO	33
FIGURA 19. INTERFAZ DE LA PÁGINA WEB DISEÑADA EN EL MÓDULO MM& SW.....	33
FIGURA 20. VARIABLES DE PRUEBA EN EL NI OPC SERVERS.....	35
FIGURA 21. VI DE PRUEBAS.	35
FIGURA 22. HERRAMIENTA DE PUBLICACIÓN DE LABVIEW.....	36
FIGURA 23. PÁGINA WEB GENERADA POR LA HERRAMIENTA WEB PUBLISHING TOOL	36
FIGURA 24. ENTORNO DE DESARROLLO DE MICROSOFT VISUAL STUDIO 2010.	40
FIGURA 25. PANTALLA CAMBIAR CONTRASEÑA	44
FIGURA 26. CREACIÓN DE FORMULARIO PARA CAMBIAR CONTRASEÑA	44
FIGURA 27. ESTRUCTURA DE DATOS ASPNET_USERS	45
FIGURA 28. PANTALLA ASIGNACIÓN DE PERMISOS.....	45
FIGURA 29. TABLA DE DATOS PRACTICA	48
FIGURA 30. PANTALLA GESTIONAR RESERVA.....	49
FIGURA 31. PANTALLA EDITAR RESERVA (DE GESTIONAR RESERVA).....	50
FIGURA 32. PANTALLA CREAR NUEVA RESERVA (DE GESTIONAR RESERVA)	50

FIGURA 33. INICIO DE SESIÓN.....	55
FIGURA 34. CREAR UNA NUEVA CUENTA.....	55
FIGURA 35. RECUPERAR CONTRASEÑA.....	56
FIGURA 36. CREACIÓN DE PANTALLA PARA INICIAR SESIÓN	56
FIGURA 37. MODELO DE MOTOBOMBA	59
FIGURA 38. MODELO DE ELECTROVÁLVULA DE CARGA DE TK2.....	59
FIGURA 39. MODELO PARA LA TOMA DE DATOS DE TK2.....	60
FIGURA 40. MODELO PARA INICIO 1	61
FIGURA 41. MODELO PARA INICIO 2	62
FIGURA 42. MODELO PARA INICIO 3	63
FIGURA 43. EVOLUCIÓN DE UNA RDP.....	64
FIGURA 44. ABRIR NI OPC SERVERS	66
FIGURA 45. CREAR NUEVO ARCHIVO (NI OPC SERVERS)	66
FIGURA 46. CONFIGURACIÓN DEL CANAL (NI OPC SEVERS).....	66
FIGURA 47. CONFIGURACIÓN DRIVER (NI OPC SEVERS)	67
FIGURA 48. VERIFICACIÓN DE PUERTO COM.....	67
FIGURA 49. CONFIGURACIÓN DE COMUNICACIONES (NI OPC SEVERS).....	68
FIGURA 50. CONFIGURACIÓN DE OPTIMIZACIÓN DE ESCRITURA (NI OPC SEVERS).....	68
FIGURA 51. CONFIGURACIÓN OPCIONES DE ENLACE (NI OPC SEVERS).....	69
FIGURA 52. RESUMEN DE CONFIGURACIÓN (NI OPC SEVERS).....	69
FIGURA 53. CREACIÓN DEL DEVICE (NI OPC SEVERS)	70
FIGURA 54. SELECCIÓN DEL MODELO DEL PLC (NI OPC SEVERS).....	70
FIGURA 55. ASIGNACIÓN DE ID (NI OPC SEVERS).....	71
FIGURA 56. CONFIGURACIÓN TIEMPO DE ACTUALIZACIÓN (NI OPC SEVERS).....	71
FIGURA 57. OPCIÓN DE RECONECTAR EL DISPOSITIVO (NI OPC SEVERS).....	72
FIGURA 58. CONFIGURACIÓN ADICIONAL (NI OPC SEVERS)	72
FIGURA 59. CONFIGURACIÓN REAL TIME CLOCK (NI OPC SEVERS).....	72
FIGURA 60. RESUMEN CONFIGURACIÓN DEVICE (NI OPC SEVERS).....	73
FIGURA 61. CREACIÓN DE GRUPO E TAGS (NI OPC SEVERS)	73
FIGURA 62. CREACIÓN DE TAG (NI OPC SEVERS)	75
FIGURA 63. ABRIR CLIENTE OPC RÁPIDO (NI OPC SEVERS)	75
FIGURA 64. CLIENTE RÁPIDO (NI OPC SEVERS)	75
FIGURA 65. NUEVO PROYECTO	76
FIGURA 66. CONFIGURACIÓN CLIENTE OPC.....	77
FIGURA 67. IMPORTAR VARIABLES PARA SER USADAS POR EL CLIENTE LABVIEW ...	77
FIGURA 68. CREAR UN VI	78
FIGURA 69. PANEL FRONTAL IMPLEMENTADO	78
FIGURA 70. DIAGRAMA DE BLOQUES IMPLEMENTADO	79
FIGURA 71. EDITAR UNA PALETA DE OPCIONES	80

FIGURA 72. INSERTAR UNA SUBPALETA.....	80
FIGURA 73. EXTENSIÓN DE VI A AGREGAR EN SUB-PALETA.....	80
FIGURA 74. UBICAR ARCHIVO IMAQUSB.LLB.....	81
FIGURA 75. FINALIZACIÓN DE MODIFICACIONES EN LA PALETA.....	81
FIGURA 76. DIAGRAMA DE BLOQUES PARA ADQUISICIÓN DE VIDEO.....	82
FIGURA 77. PANEL FRONTAL CON CÁMARAS ADICIONADAS.....	82
FIGURA 78. SELECCIÓN DE VI PARA PUBLICAR EN INTERNET.....	83
FIGURA 79. EDICIÓN DE CONTENIDO HTML.....	83
FIGURA 80. DIRECCIÓN URL.....	84
FIGURA 81. CONFIRMACIÓN DE URL.....	84
FIGURA 82. ARQUITECTURA DE LA RED.....	85
FIGURA 83.PETICIÓN AL SERVIDOR PARA INVOCAR UN VI.....	88
ILUSTRACIÓN 1. PANEL DE CAMPO DE LA PANTA SED DEL LCP.....	10
ILUSTRACIÓN 2. PANEL DE CONTROL DE LA PLANTA SED DEL LCP.....	10

LISTA DE ANEXOS

ANEXO 1: MANUAL DE USO DEL ADMINISTRADOR

ANEXO 2: MANUAL DE USO DEL ESTUDIANTE

ANEXO 3: APLICACIÓN DE AMEF AL CASO DE ESTUDIO

ANEXO 4: GRAFCET

ANEXO 5: MÉTRICA PARA LAS CARACTERÍSTICAS DE LOS CRITERIOS DE SELECCIÓN.

ANEXO 6: CASOS DE USO SEDLAB

ANEXO 7: VALIDACIÓN DEL SUPERVISOR

GLOSARIO DE TÉRMINOS

AMEF: Análisis modal de fallos y efectos
CRP: Control de Redes de Petri
DCOM: Modelo de Objetos de Componentes Distribuidos
FIET: Facultad de Ingeniería Electrónica y Telecomunicaciones
GRAF CET: Gráfico Funcional de Control de Etapas y Transiciones
HMI: Interfaz Hombre Máquina
HTML: Lenguaje de Marcado de Hipertexto
HTTPS: Protocolo seguro de transferencia de hipertexto
IDE: Entorno de desarrollo integrado
IIS: Internet Information Services
IP: Protocolo de internet
JSON: Notación de Objetos de JavaScripts
LABVIEW: Laboratory Virtual Instrumentation Engineering Workbench
LCP: Laboratorio de Control de Procesos
LMS: Sistemas de Gestión de Aprendizaje
LR: Laboratorio Remoto
LRL: Laboratorio Real con acceso Local
LRR: Laboratorio Real con acceso Remoto
LV: Laboratorio Virtual
LVL: Laboratorio Virtual con acceso Local
LVR: Laboratorio Virtual con acceso Remoto
MM&SW: Módulo de Monitoreo Y Supervisión Web
MYSQL: Lenguaje Estructurado de Consulta
NI: National Instruments
OLE: Incrustación y Enlazado de Objetos
OPC: OLE para Procesos de control
PIAI: Programa de Ingeniería en Automática Industrial
PID: Proporcional Integral Derivativo
PIPE: Plataforma Independiente para la Creación y Análisis de las Redes de Petri
PLC: Controlador Lógico Programable
RENATA: Red Nacional Académica de Tecnología Avanzada
SED: Sistema a Eventos Discretos
SVG: Gráficos Vectoriales Escalables
TIC: Tecnologías de la Información y la Comunicación
TK: Tanque
URL: Localizador Uniforme de Recursos
VI: Instrumento Virtual

RESUMEN

El presente proyecto, tiene como finalidad diseñar e implementar un Laboratorio Remoto Real (LRR) para la Planta Sistema a Eventos Discretos en el Laboratorio de Control de Procesos del Programa de Ingeniería en Automática Industrial de la Universidad del Cauca, en ella se realizan, entre otras, prácticas de simulación de procesos de bebidas carbonatadas y elaboración de jugos cítricos. Con este proyecto se pretende dar al estudiante, la posibilidad de acceder, controlar y monitorear, a través de una interfaz de usuario, las variables de la planta, apoyado en un sistema de video, que permita la visualización del laboratorio desde cualquier punto con acceso a internet.

En consecuencia, se busca implementar un sistema capaz de comunicar al usuario final con un dispositivo remoto que controla el funcionamiento de la planta, en este caso, un PLC MicroLogix 1500 instalado en un panel de control deberá cumplir esa función. Además de la ejecución del acceso remoto al recurso físico, se debe garantizar el correcto funcionamiento del mismo mediante el código que se descarga al PLC.

Por último, es necesario llevar a cabo el diseño y desarrollo del ambiente web que será el encargado de interactuar directamente con el usuario final y cumplir funciones de acceso autorizado y reserva de horarios para el uso del LRR.

En el capítulo 1 se hace una introducción a los conceptos generales relacionados con los LRR y se exponen los objetivos que se desean cumplir con esta implementación. En el capítulo 2 se realiza un análisis de recursos y requisitos con el fin de determinar los componentes hardware y software necesarios para el funcionamiento de esta herramienta tecnológica en la planta SED del LCP. En el capítulo 3 se realiza la selección de las herramientas según las necesidades específicas de este proyecto. En el capítulo 4 se lleva a cabo la implementación del LRR haciendo uso de las herramientas seleccionadas en el capítulo anterior y se muestra la arquitectura final del sistema desarrollado. En el capítulo 5 se muestran los resultados de las pruebas realizadas al sistema para validar el funcionamiento de cada uno de sus componentes.

Finalmente en el capítulo de conclusiones y recomendaciones se consignan las experiencias del desarrollo del proyecto y las expectativas para trabajos futuros.

1. INTRODUCCIÓN

El desarrollo de las redes de comunicaciones, la aparición de Internet y en general, la elevada disponibilidad de herramientas para la investigación, motiva, cada vez mas, a los estudiantes a utilizar los recursos a través de la web; y es precisamente, debido a la evolución de las TIC que se ha provocado un cambio en las metodologías de aprendizaje.

Una de las herramientas software más utilizadas por las instituciones educativas, son los LMS, sin embargo éstos se basan en la enseñanza teórica; para abordar la enseñanza práctica se proponen actividades de investigación experimental a distancia [1] mediante los LVR o LRR, en este sentido, la aplicación de las TIC en los laboratorios, tiene como objetivo facilitar y complementar los conceptos dictados en aulas de clase para promover el trabajo autónomo mediante el uso de equipos remotos.

1.1. Motivación

Este proyecto está dirigido, principalmente, a los estudiantes y docentes del PIAI de la Universidad del Cauca, para que puedan realizar experimentos de forma remota en una planta real, la Planta SED, con lo cual se espera obtener principalmente, los siguientes beneficios [2]:

- Permitir a los profesores una mejor demostración de los conceptos vistos en las aulas de clase, simplemente conectándose mediante internet y ejecutando los experimentos.
- Apoyar los cursos de aprendizaje a distancia.
- Permitir el acceso a la planta SED las 24 horas del día 7 días a la semana.
- Dar la sensación de realidad de los sistemas con los que trabajan los alumnos, lo cual se logra con la aplicación de la tele-presencia [3]
- Libertad al alumno de elegir cuándo y a qué hora realizar la práctica y distribuir su tiempo adaptando el laboratorio a sus actividades. [1]
- Extrapolar estos beneficios a otras universidades para apoyar y fortalecer la investigación colaborativa y cooperativa entre diferentes grupos de investigación [4]

1.2. Objetivos

Los objetivos nombrados a continuación, serán el eje de todo el desarrollo del LRR, y cada implementación buscara su cumplimiento.

1.2.1. General

Implementar un sistema que permita acceder, controlar y manipular la planta de SED del LCP, a través de internet.

1.2.2. Específicos

- Diseñar e implementar una interfaz de usuario que permita el monitoreo control y supervisión de la planta de SED (caso de estudio) de forma remota.
- Permitir a los estudiantes una tele-presencia en la planta caso de estudio del LCP mediante el uso de las TIC
- Implementar un sistema de acceso y reserva para la gestión de horarios vía web basados en la disponibilidad de los equipos, mediante autenticación previa.
- Garantizar la integridad física de la planta caso de estudio, especificando las consideraciones necesarias para su correcto uso.

1.3. Conceptos

En este capítulo se abordan definiciones formales con el objetivo de ampliar la información al lector sobre otras alternativas a la docencia presencial, como lo son los LR, los cuales son una opción extensible incluso a otras disciplinas diferentes a la ingeniería.

1.3.1. Antecedentes de la tesis

En la actualidad, se pueden encontrar un gran número de propuestas educativas cuyo principal objetivo es dar solución a los problemas que se presentan al momento de impartir y adquirir conocimientos teóricos o prácticos. De este amplio número de propuestas que surgen como posibles soluciones, cabe resaltar la acogida que ha tenido por parte de Instituciones Educativas, el uso de plataformas de aprendizaje basadas en las TIC. Al tiempo que las plataformas educativas evolucionaron, surgieron soluciones orientadas a fortalecer el conocimiento práctico impartido en las diferentes

áreas, estas soluciones tecnológicas conocidas como LR brindan una alternativa que apunta al trabajo virtual, institucional y multidisciplinario.

Los desarrollos alrededor de los LR son bastante amplios, pero en su gran mayoría han sido planteados como una ayuda pedagógica que brinda nuevas posibilidades dentro de los diferentes campos de estudio que comprende la ingeniería [5,6]. Actualmente es posible encontrar la implementación de LR en proyectos relacionados especialmente con sistemas de control, robótica o mecatrónica como en [7,8], los cuales cuentan con interfaces gráficas que permiten al usuario el acceso, manipulación, configuración y visualización de la información proveniente directamente de los procesos a través de páginas web o recursos desarrollados en distintos lenguajes de programación.

En el contexto internacional la implementación de este tipo de tecnologías han permitido validar su funcionalidad y aplicación respecto a las expectativas planteadas alrededor de las mismas. En este sentido proyectos como el descrito en [9], en el cual diferentes universidades españolas unieron esfuerzos para compartir recursos de experimentación a través de internet en un contexto de educación flexible, llevan a pensar que este tipo de proyectos son viables dentro de una institución educativa, al tiempo que permiten tener un punto de referencia en esta clase de experiencias para adelantar iniciativas con miras al mejoramiento de la calidad del aprendizaje. De estudios similares como el presentado en [10], surgen propuestas relacionadas que sugieren un conjunto de pautas a seguir en el proceso de diseño e implementación de los laboratorios virtuales y/o remotos, siendo este segundo el objeto de estudio de este trabajo. Algunos de los principales aportes en cuanto a las experiencias obtenidas con el uso de estas herramientas se pueden consultar en [11,12] donde es posible hacer un análisis concerniente a los alcances obtenidos con la experimentación desde los diferentes puntos de vista, tanto del alumno como del profesor.

En Colombia se han desarrollado estudios relacionados, los cuales han permitido un acercamiento más claro a las metodologías planteadas por las TIC en el marco de las nuevas propuestas para la educación. Casos de estudio como [13], donde su aplicación permite a los estudiantes llevar a cabo prácticas de laboratorio relacionadas con Ingeniería Sísmica y Dinámica Estructural, utilizando para ello un conjunto de módulos físicos que permiten reproducir diferentes movimientos como sismos, impulsos y vibraciones armónicas sobre un conjunto de estructuras modelo. La educación a distancia en el área de la automatización, no ha sido ajena a la aplicación de este tipo de desarrollos [14]. Así como también se debe rescatar la colaboración y participación entre Instituciones Universitarias con intereses comunes como se evidencia en [4], cuyo proyecto adelantado por la Universidad del Valle y la Universidad del Quindío, consistió en implementar

un laboratorio distribuido de acceso remoto a través de RENATA, para el control y supervisión de robots.

1.3.2. Taxonomía

Como se describe en [2,4] los laboratorios pueden ser clasificados conforme a dos criterios: 1. La forma de acceder al laboratorio, 2. La naturaleza del sistema físico, como se muestra en la Tabla 1. Los LV crean ambientes que simulan el comportamiento real de las plantas, los cuales son una buena opción, en especial si se usan como un complemento para los LR, ya que los estudiantes tienen un acercamiento al laboratorio, antes de realizar un experimento real, de esta manera se previenen daños a la integridad física del LR y se asegura una comprensión previa de los experimentos.

Por otra parte, los LR son una buena elección en cuanto a experimentación, ya que los estudiantes pueden percibir el laboratorio tal y como es, de ésta manera pueden asimilar más fácil los conceptos, en el caso de los LRL, los alumnos pueden usar sus cinco sentidos para percibir la planta física, sin embargo se enfrentan a limitaciones, de tiempo y espacio principalmente, porque los laboratorios están restringidos, normalmente a la presencia del instructor, permaneciendo el laboratorio cerrado e infrautilizado la mayor parte del tiempo; una solución a este inconveniente, es presentada y desarrollada a lo largo de este documento: el desarrollo de un LRR, ya que permite el acceso desde cualquier equipo con internet, con mayor eficacia y menor movimiento de personal llegando hasta estudiantes marginados económica y/o geográficamente y además está equipado con una o más cámaras que brindan al estudiante la posibilidad de tener una tele-presencia en el laboratorio.

Tabla 1. Taxonomía de laboratorios

	Virtual	Real
Local	Laboratorios presenciales con plantas simuladas	Laboratorios reales con plantas reales
Remoto	Laboratorios remotos con plantas simuladas	Tele-operación de una planta real

Fuente: Propia, Marzo de 2013. Basada en [2]

1.3.3. Características de un LRR

Las características de los LRR han sido abordadas en varios textos, sin embargo en [15] se reúnen y resumen las principales, mediante un estudio realizado en el mes de Julio de 2007, en el cual se realizó una encuesta a

distintos desarrolladores en el área de los LR, que consistió en evaluar con un puntaje de 1 a 10 la importancia de cada requerimiento, la investigación realizada arrojó como resultado que los investigadores valoran más la universalidad del LR que la potencia del mismo, es decir, les preocupa más que se pueda acceder al recurso con seguridad y desde cualquier plataforma, que la reducción de la potencia que esto puede conllevar.

A continuación se listan y explican las características objeto de estudio, las cuales serán tenidas en cuenta en gran parte del desarrollo de este proyecto para la consecución de los objetivos planteados:

- **Multiplataforma:** Se puede acceder desde cualquier Sistema Operativo: Windows, Linux, Mac OS, etc.
- **Navegadores:** Se puede acceder desde cualquier navegador sin restricciones de ningún tipo Explorer, Mozilla, Opera, Safari, etc.
- **Seguridad:** Usa https y no necesita permisos en los “firewalls”, solo necesita abiertos los puertos 80 y 443, etc.
- **Intrusividad:** No es necesaria la configuración de ningún tipo en el servidor con el fin de negar permisos al cliente para acceder a los recursos propios de equipo, tales como discos duros y periféricos.
- **Instalación:** No necesita de ninguna instalación previa tipo plug-in, JVM, Flash Player, etc.
- **Interacción:** Tiene implementada la máxima interacción con el usuario.
- **Dispositivos:** Se puede acceder desde diversos dispositivos: PDA, teléfono móvil, etc.
- **Ancho de banda:** Tiene gran eficacia en el uso del ancho de banda.
- **Audio y vídeo:** Cuenta con servicios de audio y vídeo.
- **Potencia:** El LR es muy complejo y necesita de una potente herramienta para ser implementado.

1.3.4. Pasos para la implementación de un LRR

Aunque no hay una metodología definida, ni estándar, a la hora de implementar LRR, en documentos como [10] se presenta una propuesta para el diseño de un ambiente remoto, de la cual se han extraído y planteado, básicamente los siguientes pasos:

- a) Diseñar y desarrollar un sistema simulado, con sus capacidades de procesado, sensores y controles contenidos en uno o más ordenadores, que permite acceso local a los recursos reales y/o simulados, lo que se denomina un Instrumento Virtual.
- b) Realizar el control en el VI para obtener un Instrumento Remoto, a través del cual se envíen parámetros de control y se obtenga como respuesta el

estado de las variables del sistema. La conexión entre los usuarios y dicho recurso remoto puede ser considerado como una arquitectura cliente servidor.

- c) Implementar un sistema de acceso y reserva para permitir la solicitud y planificación de la planta remota por un periodo determinado.
- d) Diseñar el entorno de aprendizaje virtual mediante el uso de recursos de aprendizaje, como por ejemplo, contenidos didácticos, actividades programadas o tareas en línea.
- e) Hacer uso de un sistema Gestor de Contenido, el cual consiste en una aplicación de servidor desarrollada en un lenguaje de programación que se utiliza para la gestión y distribución de actividades formativas.

1.3.5. Retos actuales de los LRR

Siendo el LRR una herramienta de carácter educativo, continúa siendo un gran reto lograr integrar de manera efectiva estos recursos tecnológicos con los modelos que actualmente son utilizados para la realización de prácticas de laboratorio de tal manera que sea posible impartir conocimiento al tiempo que se desarrollan habilidades y destrezas de los estudiantes, una de las tendencias en la que se debe trabajar es el desarrollo de entornos virtuales de aprendizaje inmersivo o plataformas tecnológicas en 3D para lograr más participación e interacción del estudiante y/o investigador con los experimentos a desarrollar.

2. ANÁLISIS DE RECURSOS Y REQUISITOS

En este capítulo se describen los recursos tanto hardware como software, con los cuales cuenta el LCP y que serán usados en la implementación del LRR, más adelante se determinan los requisitos tanto del cliente como del servidor y son clasificados en dos categorías: funcionales y no funcionales, dentro de éste último se enfoca un análisis para el cumplimiento de uno de los objetivos propuestos que se refiere a garantizar la integridad física de la planta

2.1. Recursos del LCP

Para la recopilación de la información relevante a todos aquellos recursos disponibles y necesarios para la implementación de un LRR sobre la planta de SED se tuvieron en cuenta las recomendaciones de las experiencias plasmadas en algunos documentos como [16,3].

Como se mencionó anteriormente, éste proyecto tiene como objetivo general la implementación de un sistema de acceso remoto a la planta SED, el análisis de recursos disponibles y necesarios estará basado en esta planta específicamente.

2.1.1. Recursos Hardware

El LCP de la Universidad del Cauca cuenta con recursos importantes a nivel de software y hardware, entre los más importantes están las plantas físicas sobre las cuales los estudiantes del PIAI realizan las prácticas de laboratorio, conformando de esta manera un soporte fundamental para diferentes asignaturas de este programa.

De los recursos mencionados anteriormente solo cinco plantas se encuentran activas, estas realizan diferentes procesos: se pueden llevar a cabo procesos de control de presión (Planta de Presión), control de nivel (Planta de Nivel), control de temperatura (Planta de Temperatura), control PID desacoplado para un sistema de tanques interactuantes (Planta Multivariable de Tanques Interactuantes) y procesos de realización de bebidas carbonatadas (Planta SED).

2.1.1.1. Descripción física de la planta caso de estudio

La Planta SED es un sistema compuesto por una motobomba, un circuito hidráulico, electroválvulas, agitadores, detectores de nivel y varios tanques

de almacenamiento conectados a un panel de control y está diseñada para emular un proceso de elaboración de bebidas carbonatadas o jugos cítricos, pero también es posible implementar otros procesos afines según sea necesario.

En el computador de la planta están instaladas herramientas software que permiten el modelado y análisis utilizando diversos formalismos. La generación del código ladder para el PLC se puede realizar manualmente o de forma automática con algunas herramientas suministradas (CRP [17]).

2.1.1.2. Distribución de la planta SED en el LCP

La distribución de ésta planta en el LCP está dividida en dos paneles, un panel de control y un panel de campo. A continuación se describe, de manera general, cada uno de ellos, el lector podrá encontrar una ampliación de esta información en [18]:

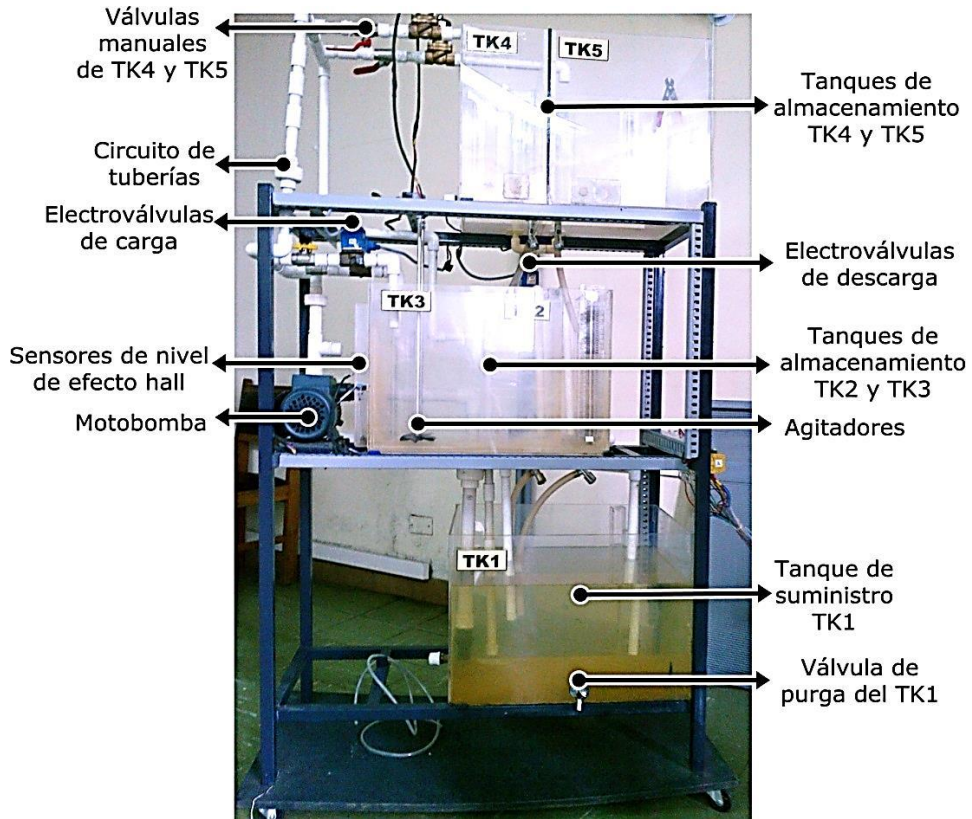
2.1.1.2.1. Panel de Campo

La planta SED está compuesta por diversos tanques para el almacenamiento y mezclado de sustancias líquidas mediante agitadores, el flujo de materia prima se distribuye por un sistema de tuberías y su control se hace por medio de electroválvulas y sensores de nivel de efecto Hall. Una motobomba impulsa el fluido que se distribuye entre los tanques, los flujos se pueden regular manualmente por medio de válvulas de bola. Ver Ilustración 1

2.1.1.2.2. Panel de Control

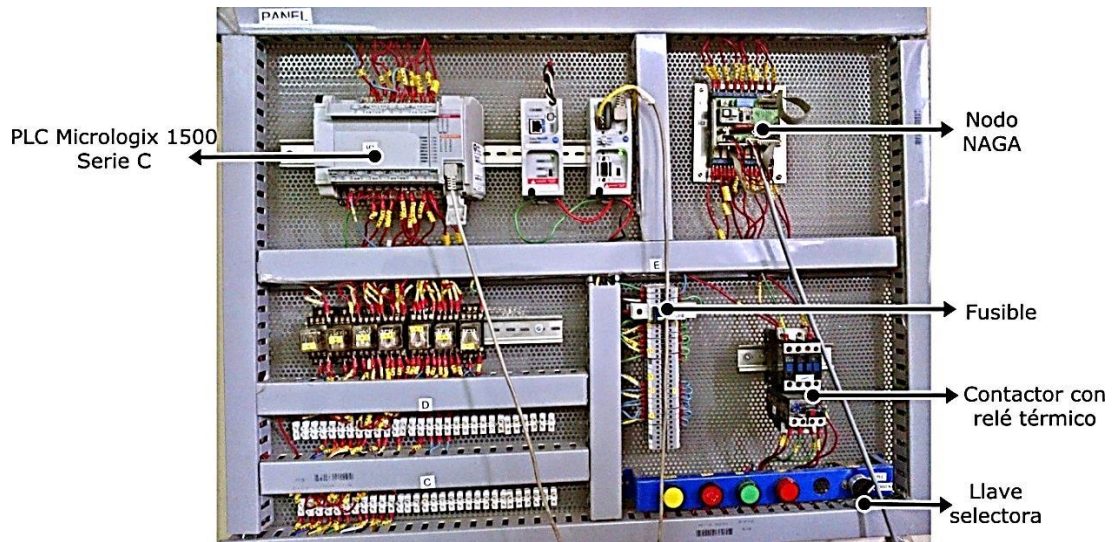
El control y monitoreo del comportamiento de la planta SED se hace por medio de dos escenarios de automatización, el primero basado en PLC y el segundo basado en una red de bus de campo, en éste panel se ubican, entre otros, los relés electromecánicos encargados de la conmutación de la instrumentación en campo entre los dos escenarios según se determine por una llave selectora, un contactor con su relé térmico para el circuito de mando y potencia de la motobomba, un PLC Micrologix 1500 y un nodo DeviceNet NAGA. Ver Ilustración 2.

Ilustración 1. Panel de Campo de la Panta SED del LCP



Fuente: Propia, Enero de 2013

Ilustración 2. Panel de control de la planta SED del LCP



Fuente: Propia, Enero de 2013

2.1.2. Recursos Software

En cuanto al software disponible en el LCP, el PIAI cuenta con algunos paquetes tecnológicos de MatLab y Rockwell Automation, los cuales deben ser explorados para evaluar su funcionalidad dentro del desarrollo e implementación de un LRR en ambientes educativos.

2.2. Análisis de los Requisitos

Después de realizar la descripción general de los recursos Hardware y Software disponibles en el LCP del PIAI, es necesario identificar, analizar, documentar y validar los requerimientos del sistema que son indispensables para la implementación de un LRR y con los cuales no se cuenta actualmente. Básicamente estos requisitos se dividen en requisitos hardware y software, funcionales y no funcionales

Los requisitos funcionales corresponden a los que ha de asumir el sistema para que opere de una forma deseada y correcta. Por otra parte los requisitos no funcionales son aquellos que definen cualidades del sistema tales como la calidad, el rendimiento, la escalabilidad o la seguridad. Para definir estos requisitos se realizó una extracción de información de sistemas anteriores existentes, con lo cual se obtuvo un conjunto de características atractivas para los usuarios de un LRR.

2.2.1. Requisitos Hardware y Software

En cuanto a los requisitos Hardware y Software más adecuados para la implementación de un LRR, la bibliografía es bastante amplia en relación a este tipo de proyectos. Teniendo en cuenta que un LRR funciona básicamente bajo una arquitectura de tipo Cliente-Servidor, es posible identificar los requerimientos generales de cada parte.

2.2.1.1. Requisitos Hardware

La disponibilidad de los requisitos que se mencionan a continuación solo será necesaria tenerla en cuenta en el lado del servidor, pues, el cliente solo deberá contar con un computador con acceso a internet para poder acceder al LRR.

- Un computador para alojar Servidor Web, Servidor de datos.
- Cámaras usb
- Microfono

2.2.1.2. Requisitos Software

Gran parte de los recursos software necesarios para la implementación del LRR deberán ser instalados sobre el equipo de cómputo ubicado en la planta física, pues, este equipo tendrá que cumplir con las funciones de un servidor y será el responsable de realizar tareas de control de acceso al recurso, comunicación entre la planta y el usuario final, interacción entre la interfaz de usuario y la planta de SED, entre otras.

- **Software para el sistema de autenticación y reserva:** Aplicación encargada de la gestión del acceso al sistema de laboratorio y la reserva de horarios para el uso del mismo.
- **Software para comunicación entre el recurso físico y el usuario:** Software encargado de poner a disposición de terceros programas o aplicaciones, las variables adquiridas y guardadas en el servidor de datos.
- **Ip Pública:** Necesaria para poder acceder a la planta SED desde una red externa a la red de la Universidad del Cauca. El proceso para la solicitud de este requisito debe tramitarse directamente en la división de las TICs de la Universidad del Cauca.

2.2.2. Requisitos Funcionales

Los requisitos funcionales definen el comportamiento interno de sistema, por tanto se hace necesario primero conocer los usuarios y el funcionamiento general que se espera tener.

2.2.2.1. Usuarios del sistema

Se establecieron dos roles de usuario para el uso del LRR: el **Administrador**, que además de las funciones del **Usuario**, también está encargado de administrar los usuarios y permitir/denegar el acceso al panel remoto, gestionar la reserva de horarios y hacer mantenimiento y gestión del sitio web y el **Usuario** quien está autorizado para acceder al panel de control remoto de la planta SED.

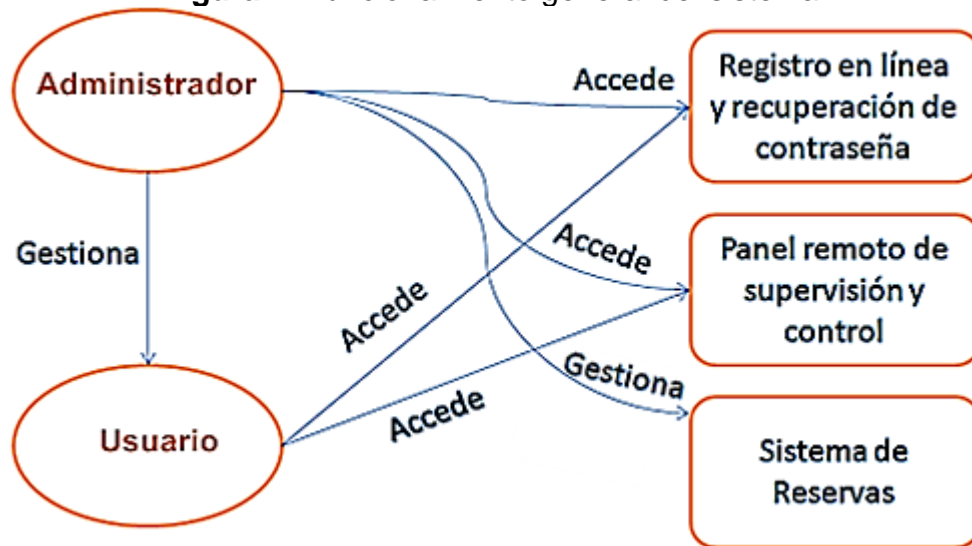
2.2.2.2. Funcionamiento general deseado

Inicialmente los estudiantes se registran en el sitio web pero es el administrador quien gestiona a los alumnos que utilizarán el acceso remoto a la planta SED, es decir, el administrador podrá ver todos los usuarios

registrados y administrar solo los que deban ser autorizados, además podrá crear, editar o borrar reservas de horarios para bloquear las horas en que la Planta SED estará en uso.

Los alumnos pueden ver información general del LCP, ver horarios no disponibles y cuando sean autorizados, hacer uso del Panel Remoto para operar la Planta SED. En la Figura 1 se presentan las tres acciones que relacionan los dos tipos de usuarios del sistema.

Figura 1. Funcionamiento general del sistema



Fuente: Propia, Marzo de 2013

2.2.2.3. Listado de funcionalidades del sistema

A continuación se listan, de manera general, las funcionalidades de los usuarios definidos previamente:

Funcionalidades del usuario

- Acceso al Panel remoto
- Registro en el sitio web
- Recuperar contraseña

Funcionalidades del administrador

Además de las funcionalidades del usuario, el administrador podrá:

- Agregar, editar o eliminar reservas de horarios para ejecución de practica remota

- Ver disponibilidad de horarios y los usuarios asignados a cada reserva
- Asignar roles de usuarios

2.2.3. Requisitos no funcionales

Se recomienda haber realizado la práctica de familiarización con la planta de SED [19] y tener al menos un conocimiento básico de los siguientes temas: Operación del PLC Micrologix 1500 de Allan Bradley y Operación del software RSLogix 500 de Rockwell Automation.

2.2.3.1. Integridad Física

Es necesario proteger la planta SED utilizándola de la forma más adecuada, garantizando así el buen estado de la misma para su continuo uso. Dado que la planta será accedida mediante internet, en este proyecto se realiza un análisis para proteger la integridad física (o también llamada mecánica) hasta donde sea posible dada la instrumentación y los recursos disponibles nombrados en la sección 2.1.1.

Es válido aclarar que la integridad mecánica hace referencia a aquella que tiene por objeto garantizar que todo equipo de proceso sea diseñado, instalado, operado, inspeccionado, mantenido y/o remplazado oportunamente para prevenir fallas, accidentes o potenciales riesgos a personas, instalaciones y al ambiente, y es diferente a la integridad de los datos, la cual es una propiedad que busca mantener con exactitud la información tal cual fue generada, sin ser manipulada o alterada por personas o procesos no autorizados [20].

A continuación se analizan los posibles fallos y las formas de corregirlos o prevenirlos.

2.2.3.1.1. Análisis de posibles fallos en la planta caso de estudio

Éste análisis se realizará bajo el método AMEF, con el fin de considerar prácticamente todas las posibilidades de fallos que se puedan evitar de forma remota o local, y con el resultado determinar las especificaciones necesarias que permitan realizar acciones preventivas para la planta.

2.2.3.1.2. Definición AMEF

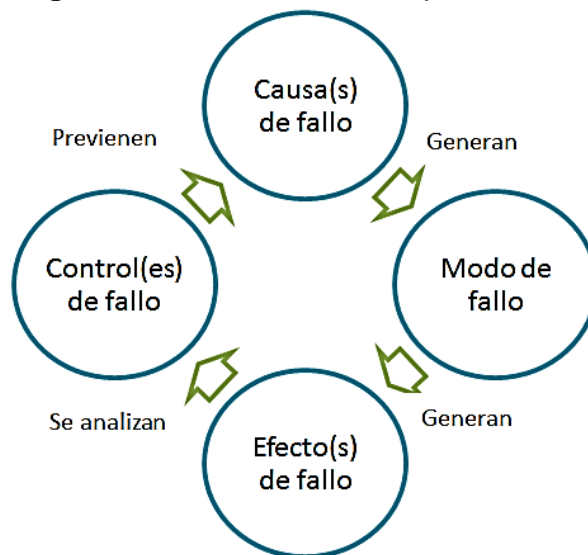
El AMEF es un proceso sistemático para la identificación y evaluación de posibles fallas del diseño de un producto o de un proceso antes de que estas

ocurran, con el propósito de eliminarlas o de minimizar el riesgo asociado a las mismas [21].

Es una de las herramientas más usadas en la planificación de la calidad, existen dos tipos: El AMEF de un producto para evaluar su diseño y el AMEF de un proceso para evaluar las deficiencias que puede ocasionar un mal funcionamiento. Su uso es recomendado como herramienta de trabajo dentro de las actividades de planificación incluidas en las tareas de diseño, ingeniería y gestión. Los conceptos más importantes que se deben tener en cuenta son explicados a continuación, un resumen está contemplado en la Figura 2:

- **Fallo:** Se dice que un producto/servicio/proceso falla, cuando no lleva a cabo, de forma satisfactoria, la prestación que de él se espera (su función).
- **Modo de fallo:** Es la forma en la que es posible que un producto/ servicio/ proceso falle.
- **Efecto de fallo:** Es la consecuencia que pueda traer consigo la ocurrencia de un Modo de fallo [22].
- **Causa de fallo:** Son todas las razones directas o indirectas por las que podría estar ocurriendo el fallo.
- **Controles actuales de fallo:** Son los controles que buscan prevenir las posibles causas de fallo.

Figura 2. Relación de conceptos AMEF



Fuente: Propia, Marzo de 2013.

2.2.3.1.3. Aplicación de AMEF al caso de estudio

En el Anexo 3 se expone el análisis realizado, conforme a este, se han clasificado los fallos de acuerdo a su control actual, esto con el fin de definir qué modos de fallo pueden ser corregidos o prevenidos remotamente (los que se pueden controlar desde líneas de código ladder) y los que solo se pueden corregir o prevenir manualmente, que finalmente hacen parte de las condiciones iniciales que debe tener la planta antes de ser accedida de forma remota y que también estarán establecidas en el Anexo 1.

2.2.3.1.3.1. Fallos que se pueden prevenir mediante programación

De acuerdo al Anexo 3 se han encontrado dos errores que son posibles de prevenir mediante el código ladder que será descargado en el PLC Micrologix 1500, éstos deben ser tenidos en cuenta si se desea implementar una práctica diferente a la propuesta en este documento.

- Sobrepaso del nivel máximo de los tanques de almacenamiento.
- Encendido de la motobomba sin salida de líquido.

Como el lector podrá ver más adelante, para este caso, el código será realizado a través de Redes de Petri y la herramienta CRP [17].

2.2.3.1.3.2. Fallos a corregir manualmente

El lector podrá encontrar un listado de los fallos que se solo se pueden prevenir o corregir manualmente en el Anexo 1.

2.2.3.1.4. Práctica a ejecutar de forma remota en la planta caso de estudio

A continuación es presentada la práctica que será tenida en cuenta en el desarrollo del código de programación, ya que dicho código deberá garantizar la integridad física de la planta.

Hoy en día, ya se encuentran realizadas tres guías sobre esta planta, por lo tanto se estudiaron cuatro posibilidades (las tres prácticas y una nueva práctica) para ser desarrolladas remotamente como se indica en la Tabla 2, bajo tres criterios:

- Involucra manipulación de elementos: Dado que el estudiante estará frente a un computador, se hace necesario darle la posibilidad de interactuar con los instrumentos y elementos de la planta.

- El objetivo de la práctica no requiere elaboración de código en línea: Implica que se puede ejecutar un código ya preestablecido y descargado sobre el PLC, esto con el fin de limitar la manipulación inadecuada de la planta.
- Reutilizar una práctica existente: Éste criterio minimiza el tiempo, para los autores de este proyecto, de realización de una nueva práctica, dado que no es uno de los objetivos de este documento.

Tabla 2. Procedimiento a ejecutar remotamente

Opción/Criterio	Involucra manipulación de elementos	No requiere elaboración de código en línea	Reutiliza una práctica existente
Practica 1		<input type="checkbox"/>	<input type="checkbox"/>
Practica 2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Practica 3	<input type="checkbox"/>		<input type="checkbox"/>
Nueva Práctica	<input type="checkbox"/>	<input type="checkbox"/>	

Fuente: Propia, Marzo de 2013

Finalmente se ha escogido utilizar la Práctica 2. Caracterización del flujo en los tanques [18], el procedimiento y la guía completa, modificados de acuerdo a este proyecto, están consignados en el Anexo 2.

3. HERRAMIENTAS

En este capítulo se realizara el proceso de evaluación y selección de las herramientas software que serán utilizadas durante la implementación del LRR, teniendo en cuenta el análisis de recursos realizado anteriormente.

3.1. Herramientas para elaboración del supervisor

Para el caso de los fallos que se pueden prevenir mediante programación se ha utilizado el modelado con RdP (mediante CRP V2.2) y Grafcet, que finalmente se incorporan en RS Logix 500. En esta sección se explican los conceptos generales necesarios para que el lector pueda tener mayor claridad sobre el uso de estas herramientas.

3.1.1. CRP

A continuación se hace una breve introducción a las RdP y se explica el funcionamiento de la herramienta.

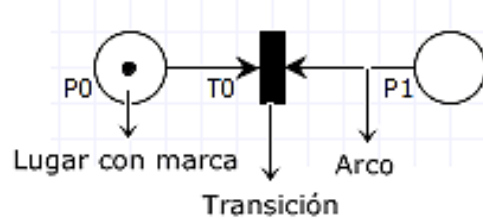
3.1.1.1. RdP

RdP es una alternativa para modelar sistemas, ya que permite una representación matemática o gráfica de un SED, con lo cual es posible visualizar el comportamiento dinámico, controlar, evaluar y optimizar procesos [23].

Está compuesta por elementos *dinámicos* y *estáticos*, ver Figura 3. Los elementos *estáticos* corresponden a los **lugares**, que representan las acciones, salidas o estados del sistema y se dibujan con círculos, las **transiciones** son equivalentes a las entradas y se simbolizan con barras, de su activación o no dependerá la evolución de la red y los **arcos** son los encargados de indicar la dirección de la RdP mediante la unión de transiciones con lugares y viceversa, no está permitida la unión entre lugares o entre transiciones.

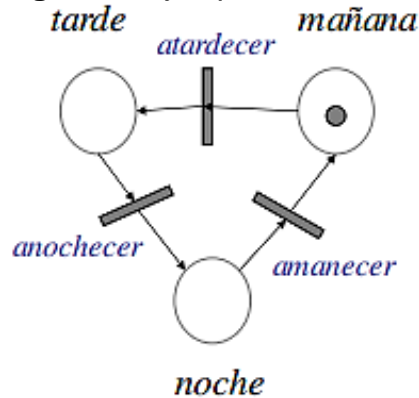
Por último los elementos dinámicos son las marcas o fichas, ubicadas dentro de los lugares e indican el estado del sistema en cada momento [24]. Un ejemplo de una RdP es mostrado en la Figura 4.

Figura 3. Componentes de una RdP



Fuente: Propia, Abril de 2013

Figura 4. Ejemplo de una RdP



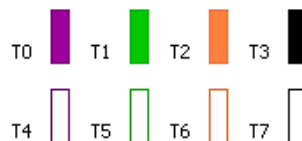
Fuente: Tomado de [25]

- **Modelado de transiciones temporizadas [17]**

Se modela el tiempo por medio de un retraso en el disparo de las transiciones, una vez habilitada la transición y está lista para disparar, empezara a contar un temporizador el cual una vez terminado activara el disparo, siempre y cuando la transición continúe habilitada y el evento siga presente, en caso de tener asociado alguno.

Cada vez que la transición se dispara se reinicia el temporizador, de esta manera se vuelve a iniciar el ciclo, las transiciones temporizadas y otros tipos de transiciones se indican en la Figura 5, en la cual (de izquierda a derecha) el color *morado* significa no controlable, el color *verde*: no observable: el color *naranja*: no controlable y no observable, el color *negro*: normal. Arriba inmediatas, abajo temporizadas.

Figura 5. Tipos de transiciones editadas en PIPE extendido



Fuente: Tomado de [17]

- **Evolución de una RdP**

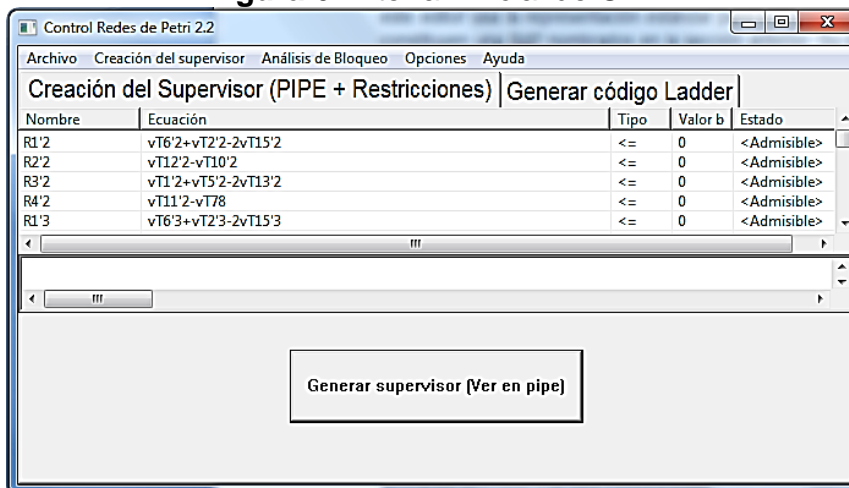
El marcado de los lugares cambia cuando se sensibilizan y disparan las transiciones, es decir se retira(n) la(s) marca(s) de un lugar de origen y se deposita(n) en cada lugar de destino. Una transición se sensibiliza cuando todos los lugares de entrada u origen están marcados [26]. Ver Figura 43.

3.1.1.2. Funcionamiento CRP

CRP es una herramienta que permite generar código ladder directamente desde RdP. La interfaz inicial de este software es presentada en la Figura 6. Tiene incorporada la herramienta PIPE, el cual es un software libre, que cuenta con una interfaz intuitiva y fácil de usar para crear y analizar RdP, este editor usa la representación estándar para los diferentes elementos que constituyen una RdP nombrados en la sección anterior. Su interfaz gráfica es mostrada en la Figura 7.

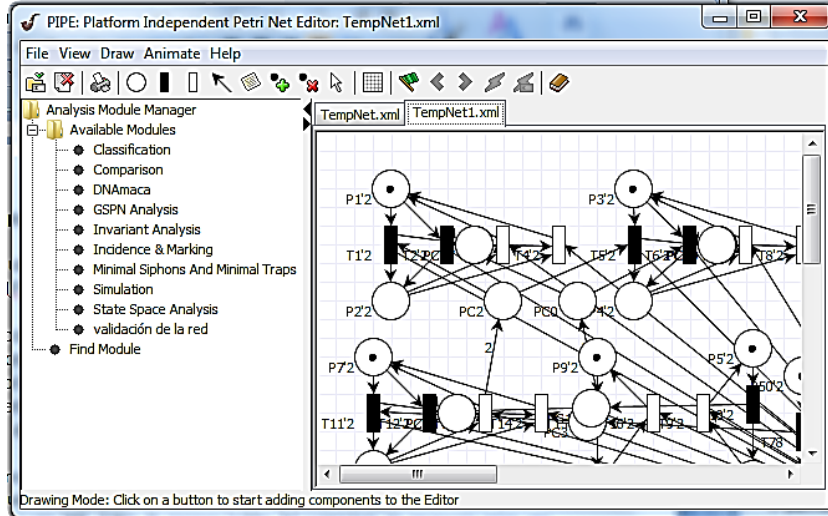
CRP calcula automáticamente el ladder del supervisor mediante una teoría expuesta en [17], el resultado de esta operación es un archivo “.SLC” para el software RsLogix 500. Antes de generar el ladder a partir del supervisor, es necesario crear una especificación para el PLC que se utilizará. Para realizar este procedimiento se hace uso de la aplicación PLC Detector, desarrollada en [17], la cual, adecua la configuración del RsLogix 500, después de esto se guarda la configuración en un archivo .SLC. PLC Detector toma este archivo y lo convierte a uno con extensión .PLC, el cual, se puede utilizar para la generación del ladder [27].

Figura 6. Interfaz inicial de CRP



Fuente: Propia, Marzo de 2013

Figura 7. Interfaz PIPE integrada en CRP V2.2



Fuente: Propia, Marzo de 2013.

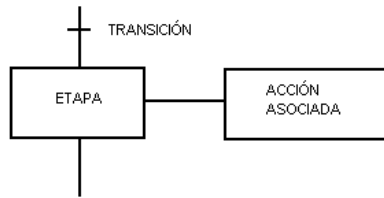
El archivo *.PLC* se debe cargar en el CRP y se deben asignar las entradas y salidas del controlador como se especificará en la sección 4.3.1.1. Una vez realizada la asignación de las E/S a los lugares y transiciones se puede generar el ladder y guardarlo en la ubicación que se desee para posteriormente ejecutarlo en Rslogix 500.

3.1.2. Grafcet

Grafcet es un método para resolver problemas de automatización de forma sistematizada, su realización es un diagrama que describe los procesos a automatizar, basándose en las acciones y condiciones para llevar a cabo la solución del proceso [28]. A continuación se describen los pasos para su elaboración:

- a) Descomponer la tarea en diferentes **etapas** (representadas por un cuadro) para luego ejecutarlas de forma secuencial. Las etapas de inicio se marcan con doble cuadro. Mediante una **acción asociada** se puede definir la acción que va a realizar la etapa. Ver Figura 8.
- b) El paso de una etapa a otra debe representarse con una línea de **transición** que las una. Perpendicularmente a estas líneas, serán descritas mediante símbolos, la condición o condiciones que se deben cumplir.
- c) La estructura final ha de ser desarrollada de arriba (Inicio de la secuencia) hacia abajo (fin de la secuencia) y siempre debe haber una condición entre dos etapas.

Figura 8. Componentes de un Grafcet



Fuente: Tomado de [28]

- **Traducción a ladder**

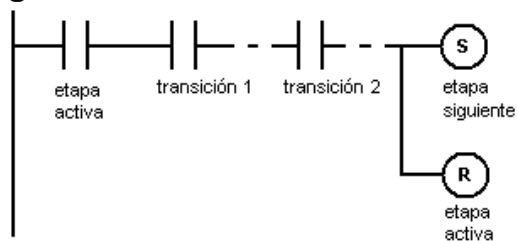
El código ladder se divide en dos partes, en la primera parte se activan etapas con transiciones y en la segunda parte se describen las acciones asociadas a cada etapa.

Parte 1: Se debe tener en cuenta solo el camino por el cual debería ir el diagrama si se saltara de etapa a etapa cuando se cumpla(n) la(s) condición(es) necesaria(s). En el caso de la etapa inicial se deberá hacer una línea diferente que responda, por ejemplo, a un botón de inicio.

Para pasar de una etapa a otra será necesario esperar a que la(s) transición(es) se cumpla(n), cuando esto ocurra se activa la etapa siguiente y se desactiva la actual, siempre debe estar UNA SOLA etapa activa [29]. Una manera de hacerlo es presentada en la Figura 9.

Parte 2: Se deben conectar las acciones asociadas a cada etapa, es decir, enganchar las acciones asociadas solo cuando cada etapa este activa.

Figura 9. Traducción de Grafcet a ladder



Fuente: Tomado de [29]

3.1.3. RSLogix 500

Este software utiliza el código ladder basado en la teoría de grafcet y generado por CRP y permite programar PLCs de la marca Allen Bradley de la familia de los SLC 500 y Micrologix (Entre los que se encuentra el PLC Micrologix 1500 instalado en el LCP). La guía de configuración y uso se puede encontrar en el Anexo 1.

3.2. Herramientas para la supervisión y comunicación

Para lograr la comunicación e interacción entre el dispositivo encargado de ejercer control sobre la planta y el usuario final de este, se deben evaluar las diferentes herramientas disponibles teniendo presente el análisis de recursos realizado anteriormente.

3.2.1. Alternativas

Las alternativas que a continuación se mencionan, fueron tenidas en cuenta dada la importancia que tiene para este proyecto lograr la comunicación de forma eficiente entre la planta de SED y el usuario final.

3.2.1.1. Alternativa Rockwell Automation

Teniendo en cuenta que gran parte de la infraestructura del LCP del PIAI es de Rockwell Automation, y además, el PIAI cuenta con un paquete de herramientas educativas del mismo proveedor, se optó por realizar una búsqueda de información al respecto con el objetivo de determinar la utilidad de dicho software y su posible uso en la implementación del LRR. A continuación se realiza una descripción general de las herramientas software del proveedor ya mencionado.

3.2.1.1.1. Herramientas Básicas

Son las herramientas típicas para la configuración de los drivers, la programación del PLC y la visualización de las interfaces de usuario. Estas son:

- **Rslinx:** Herramienta para la gestión y configuración de los drivers.
- **Rslogix 500:** Herramienta de programación para el PLC Micrologix 1500.
- **RsView32:** Herramienta basada en componentes HMI para el monitoreo y control de máquinas y procesos de automatización.

3.2.1.1.2. Herramientas Adicionales

Son las herramientas que ofrecen la posibilidad de desarrollar una solución para el tipo de implementación planteada en este documento, pero que se han utilizado muy poco o nada, pues, el PIAI no cuenta con licencias para su uso.

- **FactoryTalk ViewPoint:** Esta herramienta está basada en tecnología Web y permite publicar los displays del proyecto de Factory Talk Site

Edition en un explorador Web, todo esto mediante el manejo de roles. Ya que se basa en tecnología web, soporta todos los exploradores y no necesita instalar ningún cliente.

- **RSView32 Active Display System:** Es una aplicación cliente/servidor que extiende el alcance del software RSView32. Con esta herramienta se puede ver y controlar proyectos RSView32 desde ubicaciones remotas.
- **RSView32 WebServer:** Permite que cualquier persona con una cuenta de usuario de RSView32 pueda acceder a vistas de instantáneas de las pantallas gráficas, etiquetas y alarmas a través de cualquier navegador web estándar.

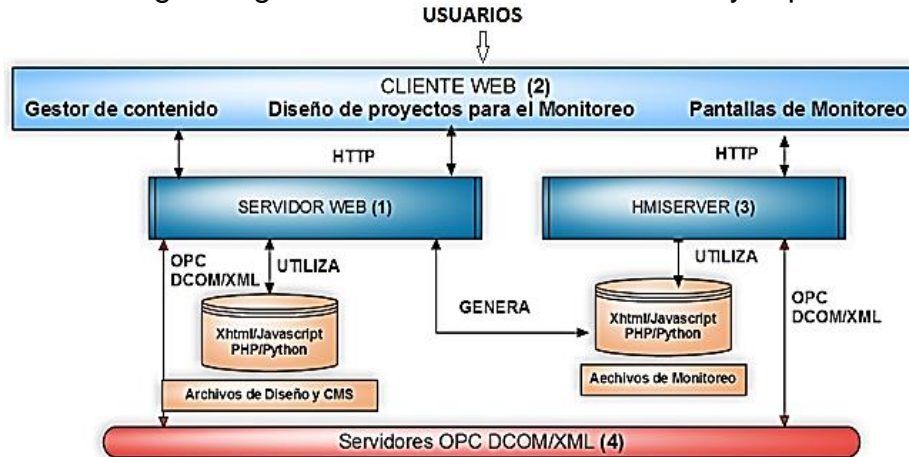
3.2.1.2. Alternativa MM&SW

En el proceso de búsqueda y selección de la información relevante a los LRR, fue posible encontrar el desarrollo de un proyecto al interior del PIAI el cual tenía entre sus objetivos, el desarrollo de un MM&SW [30], como lo denomino su autora. El propósito de este módulo permite a los usuarios diseñar y monitorear los procesos desde cualquier punto que disponga de una conexión a internet, para tal fin una aplicación web obtiene información de un servidor OPC DCOM.

El diagrama de funcionamiento y comunicaciones del módulo de MM&SW se describe en la Figura 10 y consiste en realizar de manera cíclica las siguientes etapas:

1. El servidor web (1), espera peticiones de los clientes web (2), los usuarios realizan las peticiones (ver enlaces, identificarse, registrarse, diseñar pantallas, configurar alarmas etc.), el cliente web envía la petición al servidor web, el servidor web recibe la petición, busca el recurso, y lo retorna utilizando la misma conexión por la que recibió petición.
 2. El usuario ingresa a las pantallas de monitoreo por medio de un cliente web, el servidor HMIServer (3) inicia el intercambio de datos con los servidores OPC (4), y almacena la información en memoria para que el cliente web pueda hacer uso de ella, y la visualice en la pantalla del navegador.
- **Servidor web (1):** Programa que sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten y usando el protocolo HTTP, se utiliza el sub-módulo gestor de contenido y diseño de proyectos.

Figura 10. Diagrama general del módulo de monitoreo y supervisión web.



Fuente: Tomado de [30]

- **Cliente web (2):** Programa mediante el cual el usuario solicita al servidor web el envío de información. Permite la visualización de pantallas al usuario mediante un navegador.
- **HMIserver (3):** Conjunto programas encargados del intercambio de información entre los servidores OPC y los clientes web, se utiliza para el sub-módulo de monitoreo.
- **Servidores OPC DCOM (4):** Se encargan de la captura de datos de las plataformas/dispositivos de control.
- **Archivos HTML y XHTML:** Aquí se establece el código fuente de las páginas web (HTML, Java script, PHP).

3.2.1.2.1. Herramientas Utilizadas para el Desarrollo del MM& SW

A continuación se describen las tecnologías y herramientas utilizadas por el autor para el desarrollo e implementación del MM& SW.

- **HTML (Lenguaje de Marcado de Hipertexto):** es la base de casi todas las páginas web, y se utiliza como base para las pantallas del MM&SW.
- **SVG (Scalable Vector Graphics):** permite gráficos interactivos de alta calidad que se incrustan directamente en las páginas web, los gráficos SVG son soportados por casi todos los navegadores web modernos.
- **JSON (Javascript Object Notation):** este protocolo fue desarrollado para proporcionar un medio sencillo y fiable de las comunicaciones, con

funcionalidades de alto nivel. Es utilizado en el MM&SW para el transporte de datos desde el servidor HMIServer y el cliente web.

- **MYSQL 3.23.X o superior:** es un sistema de gestión de base de datos relacional, multiusuario. MYSQL es utilizado en este módulo para almacenar información acerca de los usuarios y el sistema de monitoreo y supervisión.
- **PHP 4.2.X o superior:** es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas.
- **Apache 1.13.19 o superior:** el servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix, Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP y la noción de sitio virtual.
- **Javascript:** es el lenguaje común de programación para páginas web, y es utilizado por el MM&SW para controlar la visualización de los datos y el intercambio de datos con el servidor.
- **Wampserver:** software que permite instalar y configurar fácilmente en el sistema las últimas actualizaciones tanto del servidor web Apache, como del lenguaje de programación PHP y del servidor de base de datos MySQL.
- **Python 2.4 o superior:** lenguaje de programación en donde se implementa el servidor HMI y los clientes OPC DCOM.
- **OpenOPC:** librería que permite la implementación del cliente OPC DCOM.
- **Svg-Edit:** es un editor en línea de imágenes SVG, utilizado en el MM&SW para la construcción de los esquemas HMI.
- **RTE (Cross-Browser Rich Text Editor):** es un editor de texto diseñado en Javascript [31], utilizado en el sub-módulo de diseño de pantallas para que el usuario pueda editar la ayuda de cada proyecto.
- **Hmiserver:** servidor diseñado en Python, utilizado en el MM&SW para la implementación del sub-módulo monitoreo.
- **Flot:** es una biblioteca de Javascript, que produce gráficas de conjuntos de datos.

3.2.1.3. Alternativa LabView 2010

Tanto en implementaciones académicas como industriales LabView es un paquete software ampliamente utilizado, pues, proporciona un gran número de módulos o toolbox que facilitan el trabajo al momento de realizar proyectos técnicos o científicos en el área de la ingeniería.

LabView es una plataforma de programación gráfica o lenguaje G, lo que quiere decir que los programas son dibujados mas no escritos, esto con el

fin de minimizar el tiempo a los programadores para usarlo en el diseño de la interfaz gráfica que es la que posteriormente tendrá la interacción con el usuario final [32], ésta herramienta ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas [33].

Los programas desarrollados mediante LabView se denominan VIs, porque su apariencia y funcionamiento imitan los de un instrumento real. Cada VI consta de dos partes diferenciadas:

- **Panel Frontal:** Es la parte interactiva con el usuario. Los usuarios podrán observar los datos del programa actualizados en tiempo real. En esta interfaz se encuentran los controles (entradas) e indicadores (salidas).
- **Diagrama de Bloques:** Es el código fuente que controla el programa, donde se define su funcionalidad, aquí se colocan íconos que realizan una determinada función y se interconectan.

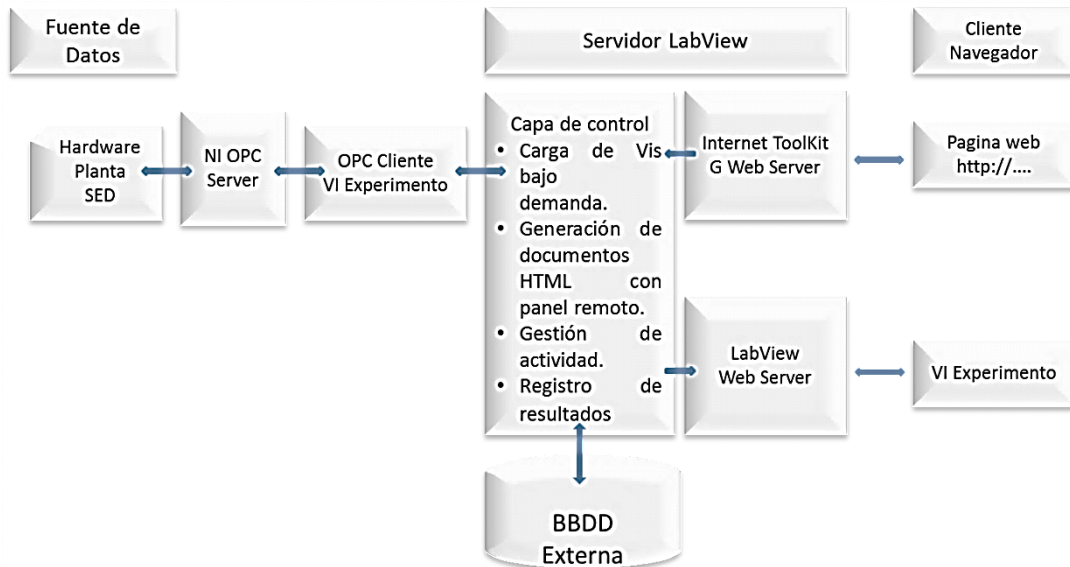
Al igual que en el MM&SW, LabView obtiene la información proveniente del dispositivo de campo por medio de un servidor OPC, en este caso el NI OPC server. En la Figura 11 puede observar un esquema de como un cliente con un navegador y mediante una URL, genera una petición al servidor LabVIEW, este lo pasa a la capa de control y devuelve el VI demandado (embebido con código HTML).

3.2.1.3.1. Herramientas utilizadas por LabView2010 para la implementación del LRR

Las herramientas o toolbox utilizadas para la implementación de LR pueden variar de un proyecto a otro, esto debido a que cada proyecto demanda del programa el uso de diferentes utilidades. Para llevar a cabo el desarrollo de proyectos como el expuesto a lo largo de este documento, se analizaron tentativamente algunas de las herramientas que ofrece LabView, estas son:

- LabView 2010
- Professional Development System
- Application Builder
- Datalogging and Supervisory Control Module
- Remote Panels
- Ni OPC Server
- Vision Development Module 2010

Figura 11. Esquema de funcionamiento de LabView para la interacción usuario – recurso remoto.



Fuente: Propia, Marzo de 2013. Basada en [34]

3.2.2. Evaluación de las Alternativas Software

Con base en los requerimientos del sistema que se desea implementar, se procedió a realizar una evaluación de las herramientas descritas anteriormente. Debido al carácter práctico del LRR se optó por realizar la instalación y prueba de los paquetes software de cada alternativa propuesta, pues, es de vital importancia para el proyecto garantizar la comunicación entre el recurso físico caso de estudio y el usuario final del mismo. Es válido aclarar que el objetivo de realizar este proceso de evaluación es mostrar de forma general las herramientas que fueron exploradas antes de escoger la que finalmente fue usada para la implementación de LRR, por esta razón el lector encontrará en detalle los pasos seguidos solo en la herramienta que, como resultado de esta evaluación, sea escogida para el cumplimiento de los objetivos propuestos.

3.2.2.1. Alternativa Rockwell

Para evaluar las posibilidades ofrecidas por esta herramienta se procedió a realizar la instalación de los paquetes software disponibles en el PIAI, con base en los procedimientos sugeridos en el manual de usuario [35]. De forma general los pasos seguidos se resumen de la siguiente manera:

Paso 1: Instalación y configuración de las herramientas Rslinx Classic, RsLadder, RsLogix500, RsEmulate500, RsView32, Active Display Server, Active Display Station, Active Display Browser.

Este proceso no representa mayores inconvenientes si se tienen en cuenta las recomendaciones del manual de usuario de cada paquete, los cuales están disponibles en [35]. Cabe destacar que todo el conjunto de programas mencionados se encuentran en la versión de 32 bits.

Paso 2: Integración del RSview32 con Active Display System.

Posterior al proceso de instalación y configuración de los paquetes software, se procedió a crear un proyecto para realizar las pruebas necesarias. Haciendo uso del código LADDER_SED y el supervisorio correspondiente a la práctica [36], se realizó la configuración y puesta en marcha del Active Display Server el cual posibilita el acceso a las pantallas creadas en RSView32 de forma remota. El acceso a la interfaz de RsView32 es posible hacerlo de dos formas, la primera usando el Active Display Browser y la segunda haciendo uso de Active Display Station.

Paso 3: Pruebas de comunicación remota.

Para realizar las pruebas remotas utilizando el Active Display System se tenían dos opciones como se describe a continuación:

- Utilizando el Active Display Browser: de esta manera lo que se hace es que el usuario o cliente del sistema remoto accede al proyecto creado en RsView32 utilizando únicamente un navegador web, esto siempre y cuando este usuario cuente con los permisos para acceder a dicho proyecto, estos permisos de acceso pueden ser configurados directamente en el proyecto de RsView32.
- Utilizando el Active Display Station: de esta manera el usuario del sistema debe instalar en su equipo el paquete software llamado Active Display Station, posterior a este proceso se podrá tener acceso al proyecto creado en RsView32 de forma remota.

Al realizar las pruebas de acceso remoto utilizando las dos alternativas mencionados anteriormente, solo fue posible acceder al proyecto de RsView32 mediante el Active Display Station, pues, mediante el Active Display Browser no se pudo lograr dicha conexión debido a un error recurrente en relación con el servidor que no pudo ser solucionado debido a la poca información disponible al respecto.

3.2.2.2. Alternativa Modulo MM&SW

Para hacer uso del módulo MM&SW desarrollado e implementado en el trabajo de grado [30], se procedió de la siguiente manera.

Paso 1: Instalación y configuración del software que contiene el módulo MM&SW

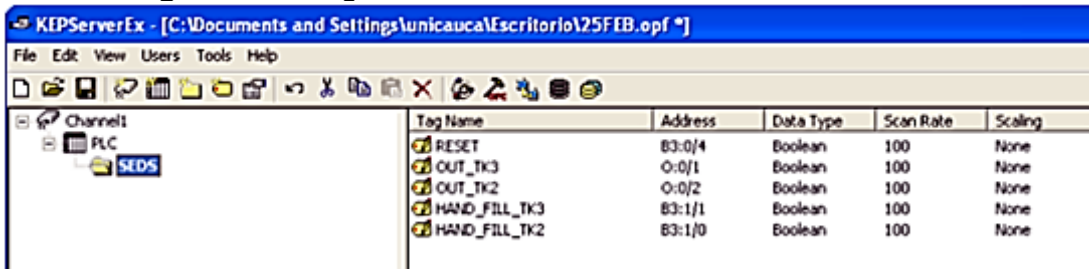
Este procedimiento requiere de la instalación del software VMware Player versión 3.1.4 o posterior para iniciar la máquina virtual que contiene los programas necesarios para ejecutar el modulo en mención, este recurso se encuentra disponible en el PC de la planta de Presión y Flujo de Aire del LCP bajo el nombre XPOPC.

Paso 2: Pruebas de comunicación entre el módulo MM&SW y la planta de SED

Para llevar a cabo este paso, se procedió de la siguiente manera:

- Haciendo uso del software Rlinx Classic y RsLogix500, se descargó el código ladder llamado **LADDER_SED**, sobre el PLC de la planta de SED, utilizado actualmente para realizar la practica dos (2) [36].
- Se identificaron las variables del código ladder con las cuales se deseaba interactuar desde el modulo MM&SW. Para efecto de pruebas se eligieron únicamente algunas variables relevantes como se indica en la Tabla 3
- Se configuró el servidor OPCKeepserverEx 4.0 en la máquina virtual siguiendo los pasos expuestos en el Anexo B de [30]. Realizada esta configuración se debe tener una ventana como la mostrada en la Figura 12

Figura 12. Carga de TAGS al servidor OPC KEEPSErverEx



Tag Name	Address	Data Type	Scan Rate	Scaling
RESET	B3:0/4	Boolean	100	None
OUT_TK3	O:0/1	Boolean	100	None
OUT_TK2	O:0/2	Boolean	100	None
HAND_FILL_TK3	B3:1/1	Boolean	100	None
HAND_FILL_TK2	B3:1/0	Boolean	100	None

Fuente: Propia, Enero de 2013.

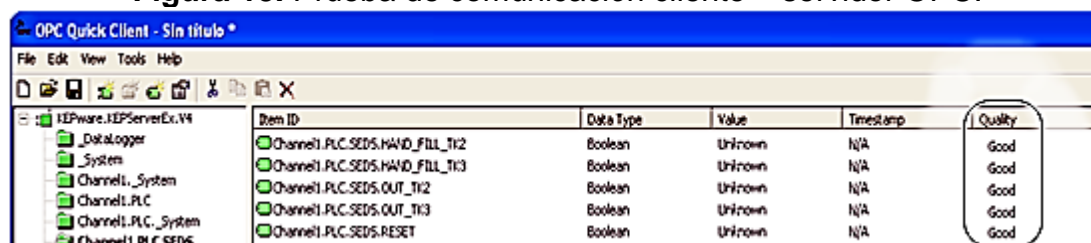
Tabla 3. Tags de prueba para el módulo MM&SW.

Nombre de la TAG	Dirección	Descripción de funcionamiento
HAND_FILL_TK2	B3:1/0	Carga manual del tanque TK2
HAND_FILL_TK3	B3:1/1	Carga manual del tanque TK3
OUT_TK2	O:0/2	Descarga manual del tanque TK2
OUT_TK3	O:0/1	Descarga manual del tanque TK3
RESET	B3:0/4	Botón de reset el cual detiene cualquier proceso en ejecución sobre la planta.

Fuente: Propia, Marzo de 2013.

- Una vez cargadas las TAGS en el servidor OPC, se realizó una prueba de comunicación local entre un cliente OPC que trae incluido el KeepServerEx y el PLC Micrologix 1500, el objetivo de este procedimiento fue verificar la comunicación entre la fuente de datos y el servidor OPC. Si la comunicación era exitosa se debía observar en el campo Quality la palabra Good, como se muestra en la Figura 13.

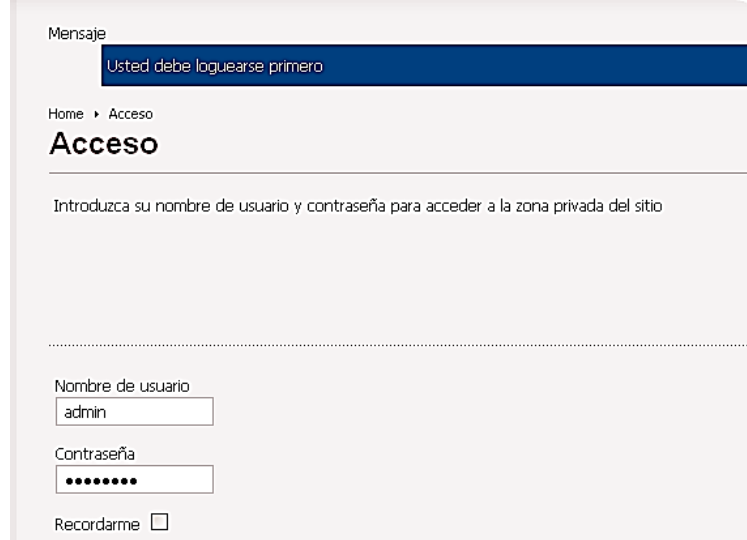
Figura 13. Prueba de comunicación cliente – servidor OPC.



Fuente: Propia, Enero de 2013.

- En este punto, el servidor OPC está listo para ser usado. De forma general el proceso consistió en ingresar a un sitio web haciendo uso de un Login y un password (ver Figura 14). Al ingresar se escogió la opción de diseño y monitoreo (ver Figura 15), y se creó un nuevo proyecto (ver Figura 16). Una vez creado el nuevo proyecto fue necesario configurar la fuente de datos desde donde se importan las variables a manipular (ver Figura 17)

Figura 14. Interfaz de acceso para el diseño de HMI



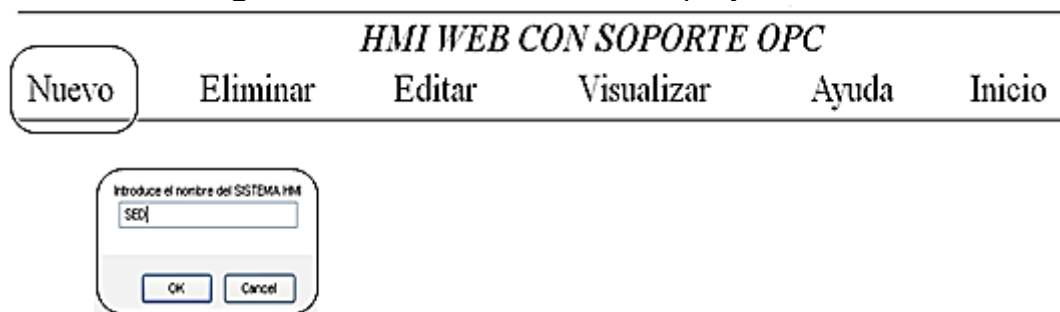
Fuente: Propia, Enero de 2013

Figura 15. Interfaz de usuario para el administrador.



Fuente: Propia, Enero de 2013.

Figura 16. Creación de un nuevo proyecto HMI.



Fuente: Propia, Enero de 2013.

Figura 17. Configuración de la fuente de datos para la interfaz HMI.



Fuente: Propia, Enero de 2013.

- Configurada la fuente de datos, se realizó el diseño de la interfaz de usuario al dirigirse a la pestaña **Diseño gráfico** (ver Figura 18)

Figura 18. Diseño de la interfaz de usuario



Fuente: Propia, Enero de 2013.

- Como resultado de los pasos anteriores se obtuvo una página web con extensión XHTML (ver Figura 19), desde la cual se pudo manipular los botones asociados a las TAGS cargadas con anterioridad al servidor de datos OPC.

Figura 19. Interfaz de la página web diseñada en el módulo MM& SW.



Fuente: Propia, Enero de 2013.

Paso 3: Prueba de comunicación remota

Utilizando la página web que se obtiene del paso dos se realizaron pruebas de comunicación dentro de la red LAN de la Universidad del Cauca, de la siguiente manera:

- Utilizando algunos equipos ubicados dentro de la red LAN de la Universidad del Cauca y por medio de los navegadores Internet Explorer, Google Chrome, y Mozilla Fire Fox, se intentó realizar el acceso a la planta por medio del link <http://169.254.55.125.8080/adminSED.xhtml> que se obtuvo del paso 2.
- El acceso al sitio web fue posible después de realizar algunas configuraciones relativas a permisos en el equipo que cumplía la función de servidor.

3.2.2.3. Alternativa LabView 2010

Para evaluar las posibilidades ofrecidas por esta herramienta se procedió a realizar la instalación de los paquetes software necesarios. De forma general los pasos que se debieron seguir se resumen de la siguiente manera:

Paso 1: Instalación y configuración de LabView y las herramientas usualmente utilizadas en la implementación de este tipo de proyectos como se describe en el numeral 3.2.1.3.1.

Paso 2: Pruebas de comunicación entre LabView y la planta SED

La comunicación entre LabView y la planta Sed se dividió en dos partes, la primera corresponde la configuración y comunicación entre la planta y el servidor de datos NI OPC Servers, y la segunda parte está relacionada con la creación de un VI que hace uso de las variables cargadas al servidor de datos.

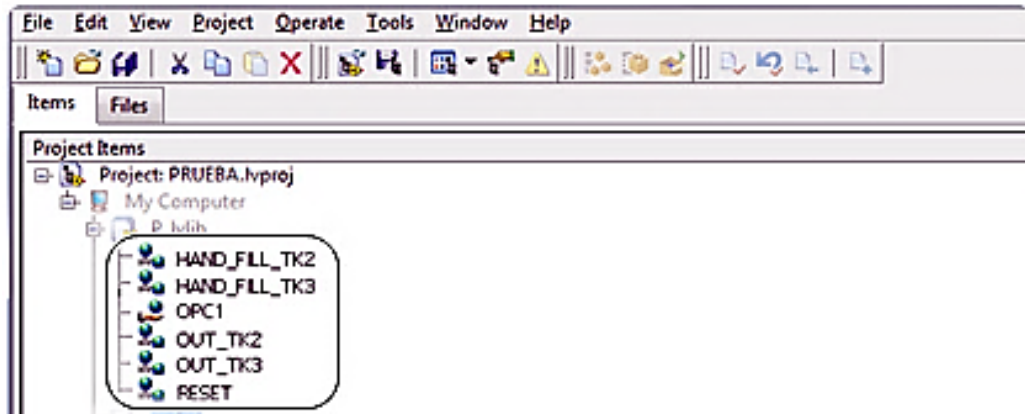
- **Configuración del Servidor NI OPC Servers.**

La configuración y comunicación del NI OPC Servers es idéntica a la descrita en 3.2.2.2 con la diferencia que el programa que hace uso de las variables de prueba de la Tabla 3 no es el módulo MM&SW sino LabView.

- **Creación del VI de pruebas**

En este punto las variables de prueba están disponibles en el servidor de datos NI OPC Servers (ver Figura 20), entonces se procede a crear un VI que permita realizar pruebas de comunicación entre LabView y la planta de SED, (ver Figura 21).

Figura 20. Variables de prueba en el NI OPC Servers.



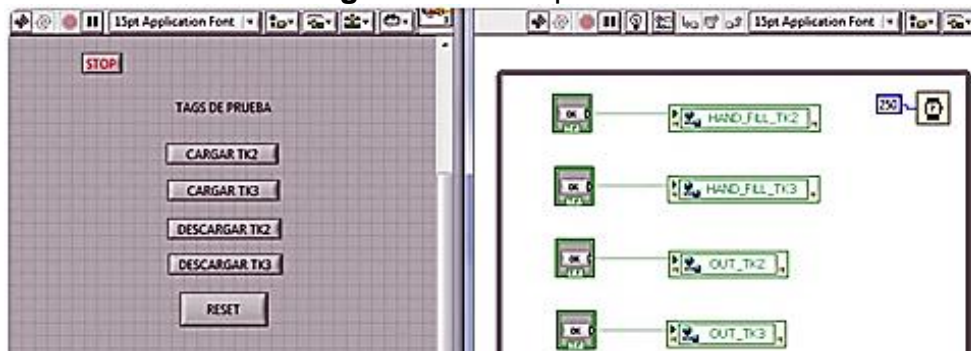
Fuente: Propia, Febrero de 2013.

Paso 3: Pruebas locales y remotas

Las pruebas locales se realizaron ejecutando el VI desde el PC ubicado en planta, los resultados fueron satisfactorios puesto que se consiguió interactuar con la misma por medio de las variables que se configuraron previamente en el servidor OPC.

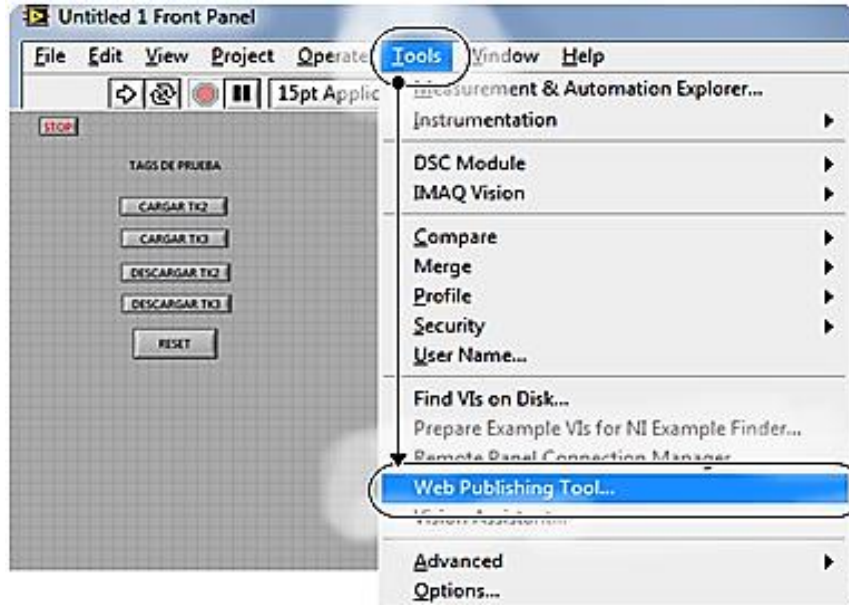
Para realizar las pruebas remotas se utilizó la herramienta Web Publishing Tool (ver Figura 22) provista por LabView 2010, esta herramienta permite que un VI sea publicado como una página web con extensión HTML a la cual es posible acceder desde un navegador como se observa en la Figura 23.

Figura 21. VI de pruebas.



Fuente: Propia, Febrero de 2013.

Figura 22. Herramienta de publicación de LabView.



Fuente: Propia, Febrero de 2013.

Figura 23. Página Web Generada por la Herramienta Web Publishing Tool



Fuente: Propia, Febrero de 2013.

3.2.3. Selección de la herramienta de supervisión y comunicación

Dado que las tres herramientas utilizadas para tratar de comunicar al usuario con el recurso físico, fueron diseñadas para cumplir objetivos diferentes, pero para el caso de estudio planteado en este documento son capaces de brindar una solución a un problema específico, se procedió a realizar la selección de la herramienta que permite la supervisión y comunicación entre

el usuario y el recurso físico teniendo en cuenta tres criterios aplicables a cada alternativa. La elección de dichos criterios se deriva de las experiencias obtenidas por autores de LR en ambientes educativos como [10,2,8], así como de los resultados obtenidos durante el proceso de evaluación realizado anteriormente.

El proceso de evaluación cuantitativa de los criterios de selección se realizó con base en la experimentación directa con cada una de las alternativas planteadas y en la asesoría de las personas que tienen un contacto directo con las plantas del LCP específicamente con la planta SED.

A continuación se describen las características que identifican los criterios ya mencionados y las cuales fueron tenidas en cuenta para realizar la selección de la herramienta de supervisión y comunicación:

a) Funcionalidad: Es la capacidad del software de cumplir y proveer las funciones para satisfacer las necesidades explícitas e implícitas cuando es utilizado en condiciones específicas. Las principales características que describen el criterio de funcionalidad son las siguientes:

- Disponibilidad funcional: Capacidad para proveer un adecuado conjunto de funciones que cumplan los objetivos especificados por el usuario.
- Exactitud: Capacidad para realizar procesos y entregar los resultados solicitados con precisión o de forma esperada.
- Interoperabilidad: Capacidad de interactuar con uno o más sistemas específicos.
- Seguridad: Capacidad para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones.

b) Usabilidad: En el ámbito de la informática y la tecnología, la usabilidad hace referencia al estudio de los principios que hay tras la eficacia percibida de un producto que ha sido desarrollado con el fin de alcanzar un objetivo en concreto. El grado de usabilidad puede definirse como una medida empírica y relativa, pues, no se deriva de opiniones o sensaciones sino de pruebas y observaciones obtenidas mediante trabajo de campo que logran cumplir con las metas planteadas. Las características que describen la usabilidad son:

- Entendimiento: Capacidad que tiene el software para permitir al usuario entender como debe ser utilizado para las tareas y las condiciones particulares de la aplicación.

- Aprendizaje: Consiste en la forma como el software permite al usuario aprender su uso. Es necesario tener en cuenta la documentación y las ayudas que el software entrega.
- Operatividad: Manera como el software permite que el usuario lo opere y/o lo controle.
- Diseño: Referente a la presentación del software, esta debe ser atractiva al usuario, es decir las cualidades del software deben ser agradables al usuario, como por ejemplo, el entorno gráfico.

c) **Confiabilidad:** La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizado en condiciones específicas. En este caso la confiabilidad se amplía a sostener un nivel específico de funcionamiento y no una función requerida. Las características que describen de una mejor manera la confiabilidad son:

- Madurez: Capacidad del software para evitar fallas cuando encuentra errores.
- Tolerancia a errores: Capacidad del software para mantener un nivel de funcionamiento en caso de errores.
- Capacidad de recuperación: Capacidad del software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.

Con base en los criterios descritos anteriormente, y teniendo presente las características que los identifican, se realizó la evaluación cuantitativa de las herramientas con el fin de obtener una valoración de las mismas a partir de las experiencias obtenidas con cada una de estas.

En el Anexo 5 se define la métrica para cada una de las características que identifican a los criterios de selección.

En la Tabla 4 se presentan los resultados obtenidos al realizar el proceso de evaluación a las alternativas planteadas en el numeral 3.2.1, teniendo en cuenta la métrica propuesta en el Anexo 5. El total ponderado corresponde a la sumatoria de las calificaciones de cada característica que describe al criterio de evaluación dividido entre el total de características evaluadas.

En la Tabla 4 se puede observar que de acuerdo con los criterios de selección utilizados, las alternativas que mejor se ajustan a los mismos son la de Rockwell Automation y LabView. Sin embargo teniendo en cuenta los resultados obtenidos de forma práctica, la herramienta LabView2010 tuvo un mejor desempeño al momento de realizar las pruebas de interacción entre el

usuario y el recurso físico de forma tanto local como remota. Por lo tanto la herramienta seleccionada para realizar la comunicación entre la planta de SED y el usuario remoto fue LabView2010.

Tabla 4. Evaluación cuantitativa de las alternativas para el proceso de supervisión y comunicación.

	Alternativa Rockwell Automation	Alternativa MM&SW	Alternativa LabView
Funcionalidad			
Disponibilidad funcional	4	3	4
Exactitud	4	3	4
Interoperabilidad	2	4	5
Seguridad	4	2	4
Usabilidad			
Entendimiento	4	2	4
Aprendizaje	4	3	4
Operatividad	4	3	4
Atracción	3	2	5
Confiabilidad			
Madurez	4	2	4
Tolerancia a errores	5	3	4
Capacidad de recuperación	5	2	4
Total Ponderado	4.1	2.46	4.18

Fuente: Propia, Marzo de 2013

3.3. Herramientas para el diseño del sistema de acceso y reserva.

Para el diseño y desarrollo del sistema de acceso y reserva de horarios para hacer uso del LRR, se consideraron tres opciones, PHP, JSP y ASP.NET. Las tres herramientas son orientadas a objetos y muy poderosas, pero finalmente se optó por el uso de la herramienta ASP.NET, ya que es una herramienta que tiene un buen rendimiento, minimiza la cantidad de código para el desarrollador y tiene uno de los mejores entornos de desarrollo IDE.

3.3.1. Tecnología ASP.NET

ASP.NET es un framework para el desarrollo de aplicaciones web, creado desarrollado y comercializado por Microsoft. Es usado comúnmente para la

construcción de sitios web dinámicos, aplicaciones web y servicios web XML. Las páginas de ASP.NET son más conocidas como formularios web (web forms) y son el principal medio para la construcción de aplicaciones web.

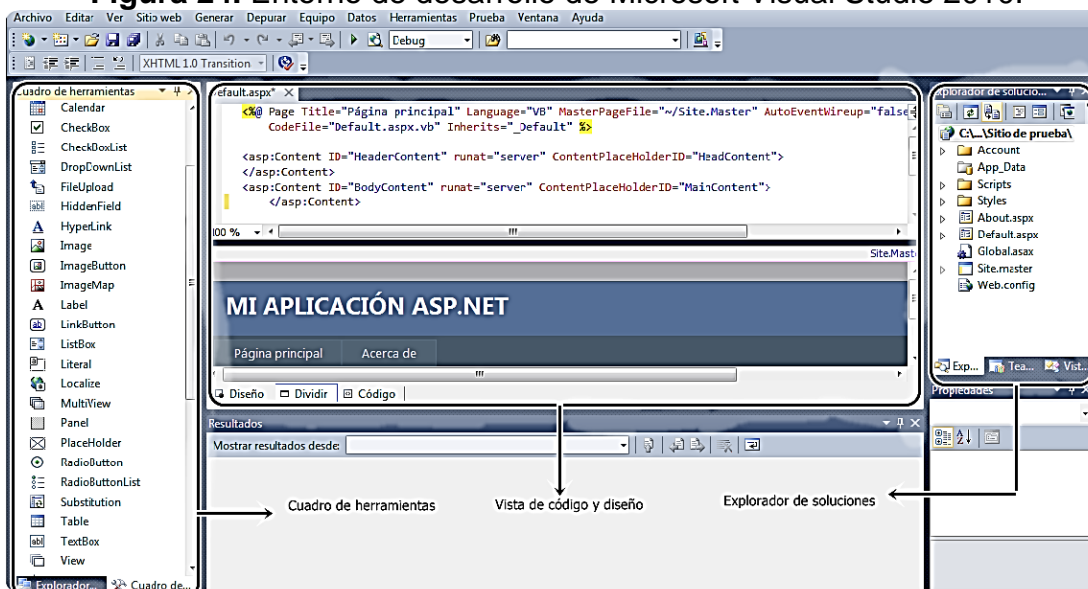
3.3.2. Entorno de desarrollo

Una buena opción para el entorno de desarrollo es Microsoft Visual Studio 2010. Esta herramienta proporciona múltiples características de programación que facilitan y agilizan el proceso de desarrollo, algunas de estas características son: plantillas de página, Intellisense (aplicación de autocompletar), llenado automático de código, y compilación en tiempo de diseño. En la Figura 24 se puede observar el entorno de desarrollo que brinda esta herramienta.

3.3.3. Servidores Web

Para trabajar con una aplicación web ASP.NET normalmente se utiliza IIS como servidor web. De esta manera se pueden probar las aplicaciones ejecutando IIS de forma local en el equipo o implementar las páginas en un servidor compartido.

Figura 24. Entorno de desarrollo de Microsoft Visual Studio 2010.



Fuente: Propia, marzo de 2013.

3.3.4. Bases de datos

Dado que la aplicación debe permitir el acceso y la reserva de horarios para ingresar al LRR, se debe considerar el almacenamiento de datos, para esto

se hace necesario el uso de una aplicación de base de datos, que para el caso es Microsoft SQL Server.

4. DESARROLLO DEL SISTEMA DE ACCESO REMOTO A TRAVÉS DE INTERNET

En éste capítulo se explica el desarrollo del sistema de acceso remoto, que a partir de este momento será llamado SEDLAB. En la primera parte se definirán las especificaciones, después el diseño basado en los requerimientos del funcionamiento y por último la implementación que tiene que ver con la comunicación necesaria para el funcionamiento del proyecto, y puesta en marcha de los servidores.

4.1. Especificaciones

Aunque en este punto ya se ha realizado la selección de las herramientas a utilizar para el desarrollo del SEDLAB, de forma independiente a las tecnologías a usar en la implementación, en esta sección se pretende describir cual debería ser el comportamiento del sistema.

4.1.1. Modelos de casos de uso

Estos modelos describen la funcionalidad propuesta del sistema, en la cual se indica la interacción entre los tipos de usuario y sistema, es decir que asocia las funcionalidades de quienes intervienen en la operación del SEDLAB. El modelo de casos de uso de SEDLAB se presenta en el Anexo 6.

4.2. Diseño

En aquellas implementaciones que requieren la interacción entre un usuario y un sistema por medio de una interfaz gráfica, es importante llevar a cabo un proceso de diseño de dicha interfaz, pues, será esta la encargada de brindar al usuario las herramientas necesarias que le permitan hacer un uso productivo del recurso, aplicación, información, etc. que se encuentre disponible en el sitio web.

4.2.1. Diseño de Entorno Web

Haciendo uso del entorno de desarrollo de Visual Studio se realizó el diseño de la interfaz con la cual va a interactuar el usuario cuando ingrese al sitio web. Este sitio consta de diferentes pantallas que estarán disponibles y

cambiaran su funcionalidad de acuerdo a la acción que esté realizando el usuario y los permisos con que este cuente. De manera explicativa a continuación se describen de forma general las pantallas que encontrara el usuario al ingresar a la aplicación de .NET acompañadas del código más relevante asociado a la misma y que describe su funcionamiento general.

La base de datos del SEDLAB se encuentra alojada en un servidor SQL Server utilizado principalmente por su sintaxis sencilla, capacidad de almacenamiento, documentación y compatibilidad con varios sistemas operativos. Esta base de datos es usada para la gestión del recurso físico y los usuarios.

4.2.1.1. Inicio

Es la página que aparece por defecto cuando cualquier usuario ingresa al sitio web, y está encargada de dar una presentación general del laboratorio y su servicio, mediante una breve introducción y video que muestra parte del funcionamiento de las otras páginas (que solo son vistas por usuarios autorizados) como Controlar Planta, lo anterior con el fin de generar expectativa y posiblemente incrementar su uso.

4.2.1.2. Acerca de

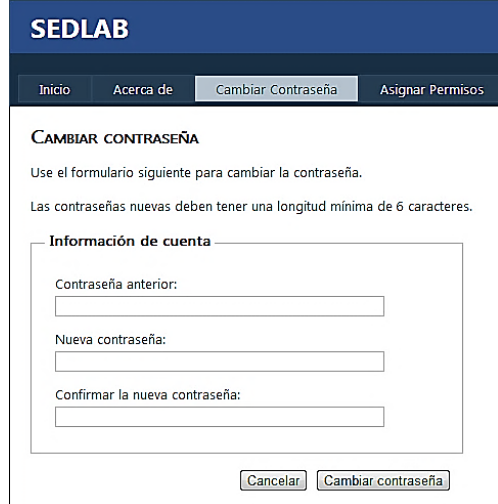
Esta página se encarga de dar una descripción general del LCP, describir el propósito de la elaboración de este proyecto y comunicar al usuario los requerimientos básicos, con los cuales se garantiza el correcto funcionamiento del sitio web en el computador que sirva como cliente, dentro de dichos requerimientos se provee de un enlace a los complementos que sean necesarios para la ejecución del Panel Remoto y la guía de usuario.

4.2.1.3. Cambiar Contraseña

En la Figura 25 se muestra la página que permite cambiar la contraseña, puede ser accedida por cualquier usuario, consiste en un formulario que solicita la clave anterior, la nueva y la confirmación. La herramienta usada para el diseño de esta página, trae incorporados algunos formularios por defecto, este es uno de ellos, por lo tanto no se realizó programación, los pasos para su creación son detallados a continuación.

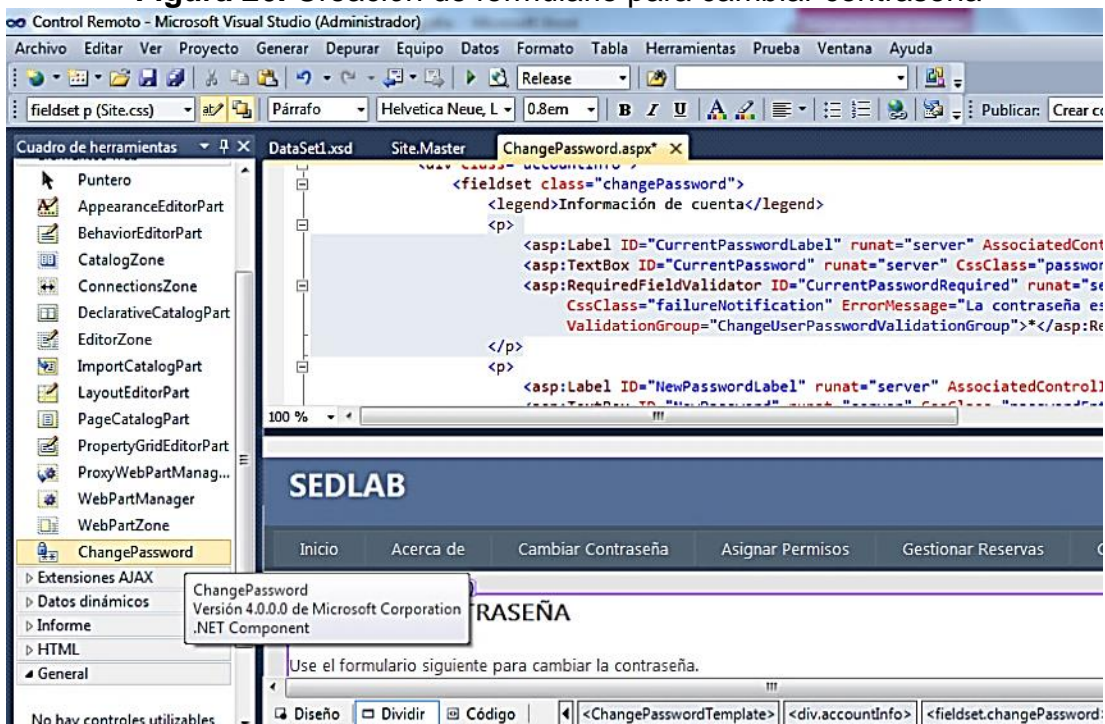
En el cuadro de herramientas que aparece a la derecha, dentro del entorno de desarrollo Visual studio, en la opción **Elementos Web**, buscar Change password y arrastrarlo hasta el cuadro de diseño, de esta manera es obtenida la página que permite el cambio de contraseña ver Figura 26.

Figura 25. Pantalla Cambiar Contraseña



Fuente: Propia, Marzo de 2013.

Figura 26. Creación de formulario para cambiar contraseña



Fuente: Propia, Marzo de 2013.

4.2.1.4. Asignar Permisos

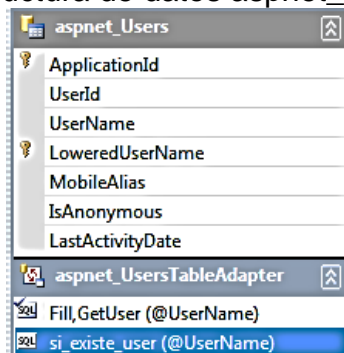
A esta pantalla solo tendrá acceso el administrador, y será usada para autorizar a los estudiantes; para realizar esta acción, primero, es necesario que el usuario se registre, luego notifique al administrador del nombre de

usuario que ha escogido (se sugiere el mismo nombre de usuario del correo institucional).

Esta pantalla utiliza la tabla de datos *aspnet_Users*, generada por defecto al crear el sitio web, a la cual se le ha agregado el método seleccionado en la Figura 27; *si_existe_user*, esta usa como entrada el texto que se escribe en el campo **Usuario** de la Figura 28 y realiza una consulta para conocer cuantos usuarios hay, cabe anotar que el resultado solo podrá ser uno o cero dado que cada usuario es único, en caso de ser uno se asocia a un estado True y en caso de ser cero al estado False, como se indica en el código que se muestra a continuación:

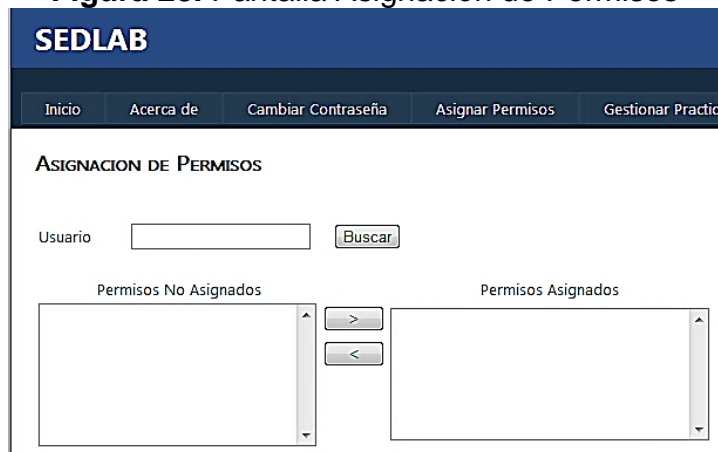
```
public static bool existe_usuario(string UserName)
{
    if (adapter_usuarios.si_existe_user(UserName).Equals(1))
        return true;
    else return false;
}
```

Figura 27. Estructura de datos *aspnet_Users*



Fuente: Propia, Marzo de 2013.

Figura 28. Pantalla Asignación de Permisos



Fuente: Propia, Marzo de 2013.

En la Figura 28 FIGURA 28 se pueden evidenciar 3 botones con los que el administrador podrá interactuar:

a) Botón Buscar

```
protected void Button7_Click(object sender, EventArgs e)
{
    if (TextBox1.Text == "" || Class1.existe_usuario(TextBox1.Text)==false)
    {
        Label1.Text = "Digite el usuario";
        TextBox1.Focus();
        ListBox1.Items.Clear();
        ListBox2.Items.Clear();
    }
    else
    {
        Label1.Text = "Bien !!";
        Obtener_Roles();
    }
}
```

```
Inicio
Si Campo de Usuario está vacío OR El usuario no se ha registrado entonces
    Escribir Digite usuario
Sino
    Escribir Bien !!
    Obtener roles del usuario
Fin
```

b) Boton >

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (ListBox1.SelectedItem != null)
    {
        Roles.AddUserToRole(TextBox1.Text,ListBox1.SelectedItem.ToString());
        ListBox2.Items.Add(ListBox1.SelectedItem);
        ListBox1.Items.Remove(ListBox1.SelectedItem);
        ListBox1.ClearSelection();
        ListBox2.ClearSelection();
    }
}
```

```
Inicio
Si Usuario da clic en el botón > entonces
    Agregar rol a usuario
    Agregar rol seleccionado a Permisos asignados
    Remover rol seleccionado de Permisos no asignados
Fin Si
Fin
```


c) Boton <

```
protected void Button5_Click(object sender, EventArgs e)
{
    if (ListBox2.SelectedItem != null)
    {
        Roles.RemoveUserFromRole(textBox1.Text, ListBox2.SelectedItem.ToString());
        ListBox1.Items.Add(ListBox2.SelectedItem);
        ListBox2.Items.Remove(ListBox2.SelectedItem);
        ListBox1.ClearSelection();
        ListBox2.ClearSelection();
    }
}
```

```
Inicio
    Si Usuario da clic en botón < entonces
        Agregar rol a usuario
        Remover rol seleccionado de Permisos asignados
        Agregar rol seleccionado a Permisos no asignados
    Fin Si
Fin
```

4.2.1.5. Gestionar Reserva

La Figura 30 muestra la pantalla con la que, solo el Administrador podrá gestionar las reservas, mediante ésta podrá no solo crear una nueva, sino también Editar o Eliminar las existentes, cada vez que esta pantalla se despliegue el Administrador tendrá acceso a un listado de las Reservas ya asignadas, el usuario relacionado, las horas reservadas y un campo de observaciones para colocar alguna nota importante.

Ésta pantalla también utiliza la estructura de datos *aspnet_Users*, los métodos *si_existe_user*, *GetUser*. Otra tabla de datos usada es *práctica*, ver Figura 29, de la que se usan los métodos: *agregar*, *disponibilidad*, *verificacion*, *obtener practicas por fecha 1*, *obtener practicas por fecha 2*. Cada uno se expone a continuación:

- **GetUser(@userName):** Obtiene el nombre de usuario, se genera automáticamente con la estructura de datos *aspnet_Users*
- **agregar(USER_ID, @FECHA, @INICIO, @FIN, @OBSERVACIONES):** Evalúa si hay una nueva reserva de usuario y en caso afirmativo devuelve True, los parámetros de entrada son el ID del Usuario, la fecha escogida, la hora de inicio y fin y las observaciones escritas.

Figura 29. Tabla

de datos PRACTICA

The image shows a screenshot of a database table named 'PRACTICA' and its corresponding adapter class 'PRACTICATableAdapter'. The table has the following columns: ID, USERS_ID, FECHA, INICIO, FIN, and OBSERVACIONES. The adapter class has the following methods: agregar (@USERS_ID, @FECHA, @INICIO, @FI..., disponibilidad (@FECHA, @H_I, @H_F), FillBy_get_practicas_for_username (@UserNa..., FillBy1_obtener_practicas_por_fecha_1 (@FEC..., FillBy2_obtener_practicas_por_fecha_2 (@FEC..., si_existe_por_username (@UserName), and verificacion (@username, @fecha, @hora).

Fuente: Propia, Marzo de 2013

```
public static bool agregar_practica(Guid User_Id, DateTime fecha, TimeSpan inicio,
TimeSpan fin, string observacioes)
{
    if (adapter_practica.agregar(User_Id, fecha, inicio, fin,observacioes)==1)
        return true;
    else return false;
}
```

- **DISPONIBILIDAD (@FECHA, @H_I, @H_F):** Consulta si hay reservas ya creadas para esa fecha y hora, en caso de que este libre ese tiempo entonces retorna el estado True.

```
public static bool disponibilidad(DateTime fecha, TimeSpan inicio, TimeSpan fin)
{
    if (adapter_practica.disponibilidad(fecha,inicio,fin).Equals( 0))
        return true;
    else return false;
}
```

- **verificacion(@username, @fecha, @hora):** Obtiene datos de los usuarios que tengan una reserva, como nombre, fecha de la reserva y hora.

```
public static bool verificar(string username, DateTime fecha, TimeSpan hora)
{
    if (adapter_practica.verificacion(username,fecha,hora) == 1)
        return true;
    else return false;
}
```

- **obtener _practicass_por_fecha_1(@FECHA):** Dado que para crear una reserva se muestran las horas reservadas en dos secciones; la mañana y la tarde, como se muestra en la Figura 32, este método retorna las consultas entre las 00:00 y 11:00 horas, es decir las reservas de la mañana.

```
public static Control_Remoto.DataSet1.PRACTICADDataTable obtener_paracticass_
por_fecha_1 (DateTime fecha)
{
    return adapter_practica.obtener_paracticass_por_fecha_1(fecha);
}
```

- **obtener _practicass_por_fecha_2(@FECHA):** Este método retorna las consultas entre las 12:00 y 23:00 horas, es decir las reservas de la tarde.

```
public static Control_Remoto.DataSet1.PRACTICADDataTable obtener_paracticass_
por_fecha_2 (DateTime fecha)
{
    return adapter_practica.obtener_paracticass_por_fecha_2(fecha);
}
```

Figura 30. Pantalla Gestionar Reserva



Fuente: Propia, Marzo de 2013.

La Figura 31 es una de las ventanas que se despliega a partir de la Pantalla Gestionar Reservas después de hacer clic en Editar, aquí se deben colocar los nuevos datos de usuario, hora de inicio y de fin y si lo desea, una observación, al hacer clic en Actualizar se agrega la nueva información a la lista de reservas creadas.

La Figura 32 es la interfaz que se usa para Crear Nueva Reserva, en la cual se presenta un formulario que el Administrador debe llenar correctamente; a los campos les fue diseñado el siguiente código:

Figura 31. Pantalla Editar Reserva (De Gestionar Reserva)

SEDLAB Sesion Iniciada como **Administrador** [[Cerrar sesión](#)]

Inicio Acerca de Cambiar Contraseña Asignar Permisos **Gestionar Reservas** Controlar Planta

GESTIONAR RESERVAS DE HORARIOS EN LA PLANTA SED

[Nueva Reserva](#)

Usuario	Fecha	Hora Inicio	Hora Fin	Observaciones	
Administrador	07/04/2013 12:00:00 a.m.	10:00:00	11:59:00	sin obs	Actualizar Cancelar
yadorado	06/05/2013 12:00:00 a.m.	22:00:00	23:59:00	prueba	Editar Eliminar

Fuente: Propia, Marzo de 2013.

Figura 32. Pantalla Crear Nueva Reserva (De Gestionar Reserva)

SEDLAB Sesion Iniciada como **Administrador** [[Cerrar sesión](#)]

Inicio Acerca de Cambiar Contraseña Asignar Permisos **Gestionar Reservas** Controlar Planta

CREAR NUEVA RESERVA

Todos los campos con * son obligatorios. Horarios **NO** disponibles para la fecha seleccionada

Usuario *

Fecha * Mañana Tarde

< mayo de 2013 >

dom	lun	mar	mié	jue	vie	sáb
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Hora de inicio *

Hora de fin *

Observaciones*

Fuente: Propia, Marzo de 2013.

a) Usuario

```

if (TextBox1.Text.Equals(""))
{
    Label1.Text = "Campo Obligatorio";
    TextBox1.Focus();
    return;}
if (!Control1.Remoto.Class1.existe_usuario(TextBox1.Text))
{
    Label1.Text = "El usuario no existe";
    TextBox1.Focus();
    return;}
else Label1.Text = "Bien!!";
    
```

```

Inicio
Si El campo Usuario se encuentra vacío entonces
    Escribir Campo Obligatorio
Si El usuario no existe entonces
    Escribir El usuario no existe
Fin Si
Sino Escribir Bien!!
Fin Si
Fin

```

b) Fecha

```

protected void Calendar1_SelectionChanged(object sender, EventArgs e)
{
    if (Calendar1.SelectedDate.ToString() == "01/01/0001 12:00:00 a.m.")
    {
        Label2.Text = "Campo Obligatorio";
        return;
    }
    else
    {
        if (Calendar1.SelectedDate >= DateTime.Now)
        {
            Label2.Text = "Bien!!";
            DateTime fecha = DateTime.Parse(Calendar1.SelectedDate.ToString());
            GridView1.DataSource = Control_Remoto.Class1.obtener_
                paracticas_por_fecha_1(fecha);
            GridView1.DataBind();
            GridView2.DataSource = Control_Remoto.Class1.obtener_
                paracticas_por_fecha_2(fecha);
            GridView2.DataBind();
        }
        else Label2.Text = "No puedes Separar el lab para fechas pasadas a la
            actual";
    }
}
}

```

```

Inicio
Si Calendario tiene la fecha por defecto 01/01/0001 12:00:00 A.M. entonces
    Escribir Campo Obligatorio
Sino
Si La fecha seleccionada es mayor o igual a la actual entonces
    Escribir Bien!!
    Agregar a la lista de reservas
Sino Escribir No puedes Separar el lab para fechas pasadas a la actual
Fin Si
Fin

```

c) Hora de Inicio

```

if (DropDownList1.SelectedItem.ToString() == "")
{
    Label3.Text = "Campo Obligatorio";
    DropDownList1.Focus();
    return;} else {
    Label3.Text = "Bien!! ";}

```

```

Inicio
  Si El campo de Hora de Inicio está vacío entonces
    Escribir Campo Obligatorio
  Sino
    Escribir Bien!!
  Fin Si
Fin

```

d) Hora de Fin

```

if (DropDownList2.SelectedItem.ToString() == "")
{
    Label4.Text = "Campo Obligatorio";
    DropDownList2.Focus();
    return;
}
else
{
    if (DropDownList1.SelectedItem.ToString() != "")
    {
        hora_in = DropDownList1.SelectedValue + ":00";
        inicio = TimeSpan.Parse(hora_in);

        hora_fn = DropDownList2.SelectedValue + ":00";
        fin = TimeSpan.Parse(hora_fn);

        if (inicio < fin)
        {
            Label4.Text = "Bien!!";
        }
        else Label4.Text = "La hora de fin debe ser mayor que la de inicio";
    }
}

```

```

Inicio
  Si El campo de Hora de Fin está vacío entonces
    Escribir Campo Obligatorio
    Enfocar Campo
  Sino
    Si el campo de hora de fin no está vacío entonces
      Hora_in = Hora inicio seleccionada
      Hora_fn = Hora fin seleccionada

      Si Hora Inicio < Hora Fin entonces
        Escribir Bien!!
      Sino Escribir La hora de fin debe ser mayor que la de inicio
    Fin Si
  Fin Si
Fin

```

e) Botón Crear

```
protected void Button_Crear_Click(object sender, EventArgs e)
{
    string hora_in ;
    TimeSpan inicio ;

    string hora_fn ;
    TimeSpan fin ;

    Validar Usuario
    Validar Fecha
    Validar Hora Inicio
    Validar Hora de Fin

    hora_in = DropDownList1.SelectedValue + ":00";
    inicio = TimeSpan.Parse(hora_in);

    hora_fn = DropDownList2.SelectedValue + ":00";
    fin = TimeSpan.Parse(hora_fn);

    DateTime fecha = DateTime.Parse(Calendar1.SelectedDate.
        ToString());

    if (Control_Remoto.Class1.disponibilidad(fecha, inicio, fin))
    {
        Guid idusuario = new Guid();
        string observaciones = TextBox2.Text;

        DataSet1.aspnet_UsersDataTable tabla = Class1.obtener_datos_
            usuario(TextBox1.Text);

        foreach (DataRow row in tabla.Rows)
        {
            idusuario = (Guid)row["UserId"];
        }

        if (Class1.agregar_practica(idusuario, fecha, inicio,fin,
            observaciones))
        {
            Label6.Text = "Registro Exitoso";
        }
        else Label6.Text = "No se pudo ingresar la nueva practica";
    }
    else Label6.Text = "La fecha y hora están ocupadas!!. Selecciona otro espacio
        de tiempo"; }
```

```
Inicio
Validar Usuario
Validar Fecha
Validar Hora de Inicio
Validad Hora de fin
Si hay disponibilidad (fecha, hora inicio, hora fin) entonces
    Obtener los datos del usuario
    Si Agregar reserva (id_usuario, fecha, hora inicio, hora fin, observacion) entonces
        Escribir Registro Exitoso
    Sino Escribir No se pudo ingresar la nueva reserva

    Sino Escribir La fecha y hora están ocupadas!! Selecciona otro espacio de tiempo.
Fin Si
Fin
```

4.2.1.6. Controlar Planta

Mediante esta pantalla el usuario puede acceder al Panel Remoto, siempre y cuando el sistema haya verificado el usuario, y la fecha y hora reservada como se muestra a continuación:

```
protected void Page_Load(object sender, EventArgs e)
{
    Panel1.Enabled = false;
    string username =(string) (Membership.GetUser().UserName);
    DateTime fecha_actual = DateTime.Now;
    int hora = fecha_actual.Hour;
    int minuto = fecha_actual.Minute;
    int segundos = fecha_actual.Second;
    TimeSpan hora_actual = new TimeSpan(hora,minuto,segundos);

    if (Control_Remoto.Class1.verificar(username, fecha_actual,
                                       hora_actual))
    {
        Label1.Text = "Éxitos en la práctica!!";
        Panel1.Enabled = true;
    }
    else Label1.Text = "No es su turno !!";
}
```

Inicio

```
PanelRemoto = Deshabilitado
Username = Adquirir usuario()
Fecha Actual = Adquirir fecha actual
Hora Actual= Adquirir Hora actual (horas, minutos, segundos)

Si Verifica que la fecha, hora, usuario correspondan a una reserva(username,
                                                                    Fecha_actual, Hora_actual) entonces
    Escribir Éxitos en la práctica!!
    PanelRemoto = Habilitado
Sino Escribir No es su turno !!
Fin
```

4.2.1.7. Iniciar sesión

El formulario para Iniciar sesión, Recuperar contraseña o Registrarse son presentados en las pantallas de la Figura 33, Figura 34, Figura 35 las cuales son opciones que tiene la herramienta, la manera de crearlas es las opciones de Inicio de sesión que están en el cuadro de herramientas, y arrastrando la que sea necesaria hasta el cuadro de diseño como se muestra en la Figura 36.

Figura 33. Inicio de Sesión

SEDLAB

Inicio Acerca de Cambiar Contraseña Asignar Permisos Gestionar Reservas Controlar Plantas

INICIAR SESIÓN

Especifique su nombre de usuario y contraseña. [Registrarse](#) si no tiene una cuenta.

Información de cuenta

Nombre de usuario: *

Contraseña:

Mantenerme conectado

[Olvidaste la contraseña?](#)

Fuente: Propia, Marzo de 2013

Figura 34. Crear una Nueva Cuenta

SEDLAB

Inicio Acerca de Cambiar Contraseña Asignar Permisos Gestionar Reservas Controlar Plantas

CREAR UNA NUEVA CUENTA

Use el formulario siguiente para crear una cuenta nueva.

Las contraseñas deben tener una longitud mínima de 6 caracteres.

Información de cuenta

Nombre de usuario:

Correo electrónico: *

Contraseña:

Confirmar contraseña:

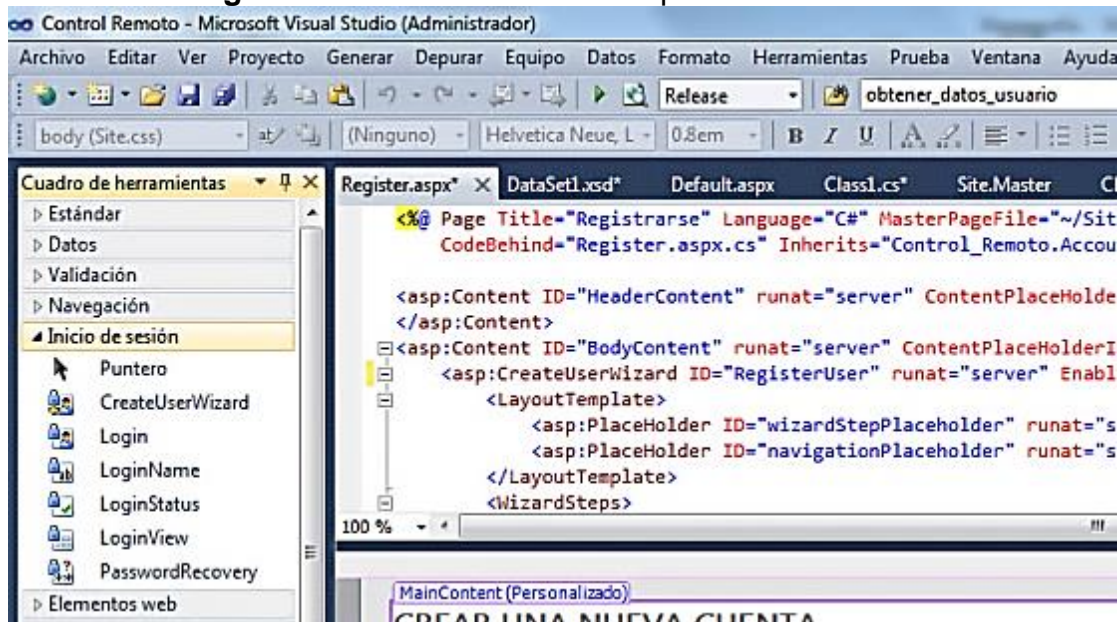
Fuente: Propia, Marzo de 2013

Figura 35. Recuperar Contraseña



Fuente: Propia, Marzo de 2013

Figura 36. Creación de Pantalla para Iniciar Sesión



Fuente: Propia, Marzo de 2013

4.3. Implementación

En este apartado se explicarán los procesos de implementación para llevar a cabo el desarrollo del SEDLAB mediante las herramientas tecnológicas que han sido escogidas previamente. Entre las principales tecnologías que se utilizarán están Windows como sistema operativo, ASP como herramienta de desarrollo web, HTMLx como lenguaje Web y LabView 2010 como lenguaje de programación gráfico.

4.3.1. Elaboración del supervisor

En este apartado se lleva a cabo el diseño de las restricciones necesarias para lograr que la RdP funcione adecuadamente, con base en los procedimientos que los estudiantes realizaran sobre la planta SED.

4.3.1.1. Aplicación de RdP al caso de estudio

En RdP se realizaron seis modelos por cada tanque de almacenamiento (TK2, TK3, TK4 y TK5), los cuales, junto a las **Restricciones físicas lógicas o de control** deben asegurar que los tanques no sobrepasen su nivel máximo y la motobomba no se encienda sin garantizar el paso del líquido. A continuación se explican detalladamente estos modelos para un solo tanque (TK2), dado que los demás están realizados del mismo modo.

La nomenclatura que aparece en la imágenes cambiará después del apóstrofe según sea el tanque, por ejemplo, el lugar P1'2 que aparece en la Figura 37, cambiará por P1'3 para el tanque 3 o P1'4 para el tanque 4.

En algunos casos se hará referencia a las salidas del PLC Micrologix 1500, las cuales han sido cableadas físicamente en el LCP, dicha información esta especificada en la Tabla 5.

Tabla 5. Descripción de puertos conectados al PLC Micrologix 1500

TAG	DESCRIPCIÓN	PUERTO IN/OUT
LLE TK1	Límite inferior tk1	I:0/1
LHE TK1	Límite superior tk1	I:0/2
LLE TK2	Límite inferior tk2	I:0/3
LHE TK2	Límite superior tk2	I:0/4
LLE TK3	Límite inferior tk3	I:0/5
LHE TK3	Límite superior tk3	I:0/6
LLE TK4	Límite inferior tk4	I:0/7
LHE TK4	Límite superior tk4	I:0/8
LHE TK5	Límite superior tk5	I:0/9
LLE TK5	Límite inferior tk5	I:0/10
FY1	Electroválvula de descarga tk 3	O:0/1

FY2	Electroválvula de descarga tk 2	O:0/2
FY3	Electroválvula de carga tk2	O:0/3
FY4	Electroválvula de carga tk3	O:0/4
FY5	Electroválvula descarga tk 4 y 5	O:0/5
FY6	Electroválvula carga tk4	O:0/6
FY7	Electroválvula carga tk5	O:0/7
MX1	Mezclador tk3	O:0/8
MX2	Mezclador tk2	O:0/9
-	Inicio remoto bomba (CM1)	O:0/10
-	Parada remota bomba (CM2)	O:0/11

Continuación Tabla 5

Fuente: Tomado de [17]

4.3.1.1.1. RdP de la Motobomba

El lugar P1'2 corresponde al modo OFF de la motobomba y contiene la marca inicial y su acción corresponde a la salida 0:0/10, por lo tanto, el lugar P2'2 corresponde al modo ON y su acción corresponde a la salida 0:0/11, en este modelo solo son posibles estos dos estados.

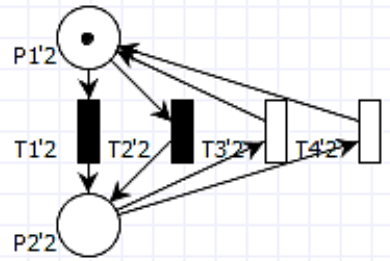
Adicionalmente tiene dos transiciones para cambiar al modo ON y dos transiciones temporizadas para el apagado. Ver Figura 37 y Tabla 6

Tabla 6. Estados y eventos asociados a la Motobomba

Motobomba			
Estados (Lugares)		Eventos (Transiciones)	
P1'2	Motobomba apagada	T1'2	Encender motobomba 2 (20'')
		T2'2	Encender motobomba 1 (30'')
P2'2	Motobomba encendida	T3'2	Apagar motobomba 1 después de 30''
		T4'2	Apagar motobomba 2 después de 20''

Fuente: Propia, Marzo del 2013

Figura 37. Modelo de Motobomba



Fuente: Propia, Marzo de 2013

4.3.1.1.2. RdP de la Electroválvula de carga del tanque TK2 (FY3)

El lugar P3'2 corresponde al modo OFF de la electroválvula FY3 y contiene la marca inicial, por lo tanto, el lugar P4'2 corresponde al modo ON y su acción corresponde a la salida 0:0/3, en este modelo solo son posibles estos dos estados, dado que es de tipo todo o nada.

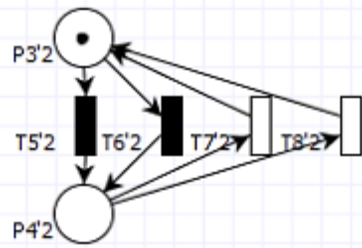
Adicionalmente tiene dos transiciones para cambiar al modo ON y dos transiciones temporizadas para el apagado. Ver Figura 38 y Tabla 7.

Tabla 7. Estados y eventos asociados a la Electroválvula de carga de TK2

Electroválvula de carga de TK2			
Estados (Lugares)		Eventos (Transiciones)	
P3'2	FY3 apagada	T5'2	Encender FY3 2 (20'')
		T6'2	Encender FY3 1 (30'')
P4'2	FY3 encendida	T7'2	Apagar FY3 1 después de 30''
		T8'2	Apagar FY3 2 después de 20''

Fuente: Propia, Marzo del 2013

Figura 38. Modelo de Electroválvula de carga de TK2



Fuente: Propia, Marzo de 2013

4.3.1.1.3. RdP para la Toma de Datos

El lugar P7'2 corresponde al modo OFF del modelo y contiene la marca inicial, por lo tanto, el lugar P4'2 corresponde al modo ON, en este modelo solo son posibles estos dos estados.

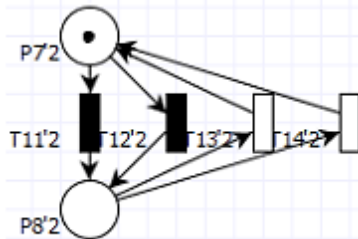
Adicionalmente tiene dos transiciones para cambiar al modo ON (la T12'2 permite la toma de datos cuando termina el primer llenado de TK2 por 30'' y la T11'2 corresponde a la segunda toma de datos, cuando termina el segundo llenado del tanque por 20'') y dos transiciones temporizadas para el apagado. Ver Tabla 8 y Figura 39

Tabla 8. Estados y eventos asociados a la Toma de datos de TK2

Toma de datos de TK2			
Estados (Lugares)		Eventos (Transiciones)	
P7'2	Toma de datos off	T11'2	Iniciar toma de datos 2 (15'')
		T12'2	Iniciar toma de datos 1 (15'')
P8'2	Toma de datos on	T13'2	Apagar toma de datos 1 después de 15''
		T14'2	Apagar toma de datos 2 después de 15''

Fuente: Propia, Marzo del 2013

Figura 39. Modelo para la Toma de datos de TK2



Fuente: Propia, Marzo del 2013

Los modelos que se describen a continuación, son necesarios para dar paso a los procedimientos del Anexo 2, básicamente se eligieron tres porque: el primero es necesario para dar inicio al procedimiento de cada tanque y los otros dos, a la toma de datos por parte de los estudiantes. Dado que esto último solo se puede realizar si, en el caso de los tanques 4 y 5, ya se han estabilizado debido a su conexión mediante una tubería. Es válido aclarar que éste tiempo de estabilización es diferente para cada uno de los procedimientos, precisamente por la naturaleza del experimento que consiste en caracterizar el flujo, lo cual implica que en unos procedimientos los tanques estén más llenos que en otros.

Las transiciones de los modelos nombrados previamente han sido temporizadas con un tiempo mínimo (0.01”), ya que solo se necesita que energicen algunos bits dependiendo de los estados y las restricciones de la RdP, por otra parte les fue asignada una entrada, pero manualmente en LADDER han sido cambiadas por bits, esto debido a que el programa de CRP no permite asignar directamente bits sino solo entradas (transiciones) o salidas (lugares).

4.3.1.1.4. Red de Petri para Inicio 1

El lugar P9'2 corresponde al modo OFF del modelo y contiene la marca inicial, por lo tanto, el lugar P10'2 corresponde al modo ON, en este modelo solo son posibles estos dos estados.

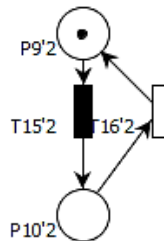
Adicionalmente tiene una transición para cambiar al modo ON es activada para iniciar el primer llenado de cada tanque, en este caso el TK2 y tiene asignada una entrada correspondiente a I:0/0 y una transición temporizada para el apagado. Ver Figura 40 y Tabla 9.

Tabla 9. Estados y eventos asociados a Inicio 1 para TK2

Inicio1 de TK2			
Estados (Lugares)		Eventos (Transiciones)	
P9'2	Inicio 1 TK2 off	T11'2	Encender Inicio 1
P10'2	Inicio 1 TK2 on	T12'2	Apagar Inicio 1 (0,01”)

Fuente: Propia, Marzo del 2013

Figura 40. Modelo para Inicio 1



Fuente: Propia, Marzo del 2013

4.3.1.1.5. Red de Petri para Inicio 2

El lugar P5'2 corresponde al modo OFF del modelo y contiene la marca inicial, por lo tanto, el lugar P6'2 corresponde al modo ON, en este modelo solo son posibles estos dos estados.

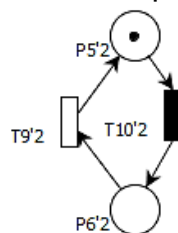
Adicionalmente tiene una transición para cambiar al modo ON es activada para iniciar la primer toma de datos y el segundo llenado de cada tanque, en este caso el TK2 y tiene asignada una entrada correspondiente I:0/1 y una transición temporizada para el apagado. Ver Tabla 10 y Figura 41.

Tabla 10. Estados y eventos asociados a Inicio 2 para TK2

Inicio 2 de TK2			
Estados (Lugares)		Eventos (Transiciones)	
P5'2	Inicio 2 TK2 off	T10'2	Encender Inicio 2
P6'2	Inicio 2 TK2 on	T9'2	Apagar Inicio 2 (0,01")

Fuente: Propia, Marzo del 2013

Figura 41. Modelo para Inicio 2



Fuente: Propia, Marzo del 2013

4.3.1.1.6. Red de Petri para Inicio 3

El lugar P11'2 corresponde al modo OFF del modelo y contiene la marca inicial, por lo tanto, el lugar P12'2 corresponde al modo ON, en este modelo solo son posibles estos dos estados.

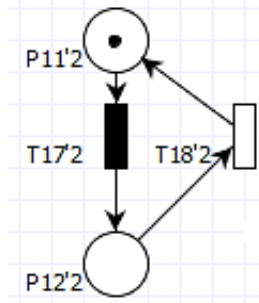
Adicionalmente tiene una transición para cambiar al modo ON (es activada para iniciar la primer toma de datos y el segundo llenado de cada tanque, en este caso el TK2 y tiene asignada una entrada correspondiente I:0/2 y una transición temporizada para el apagado. Ver Tabla 11 y Figura 42.

Tabla 11. Estados y eventos asociados a Inicio 3 para TK2

Inicio 3 de TK2			
Estados (Lugares)		Eventos (Transiciones)	
P11'2	Inicio 3 TK2 off	T17'2	Encender Inicio 3
P12'2	Inicio 3 TK2 on	T18'2	Apagar Inicio 3 (0,01")

Fuente: Propia, Marzo del 2013

Figura 42. Modelo para Inicio 3



Fuente: Propia, Marzo del 2013

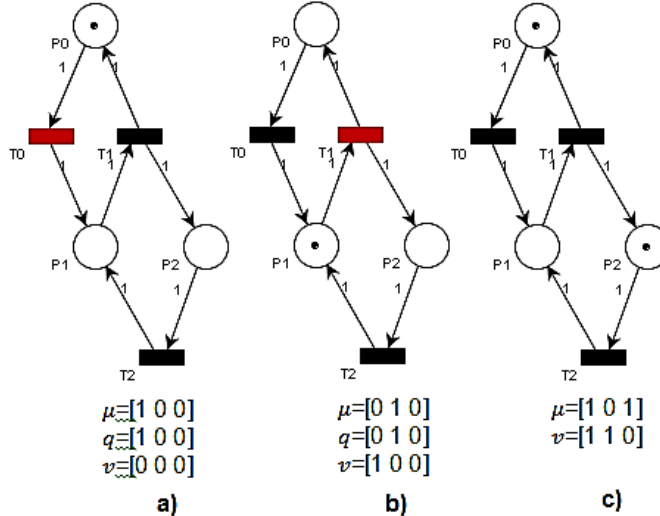
4.3.1.1.7. Restricciones físicas lógicas o de control

La edición de estas restricciones consiste en una ecuación (1), en la cual mediante los nombres de lugares y transiciones, se indican los elementos que intervienen. Se usan los caracteres u , q y v como auxiliares para indicar respectivamente, el vector de marcado, el vector de disparo y el vector de Parikh, el cual actúa como un contador de los disparos de las transiciones de la RdP. Las restricciones lineales vectoriales generales tienen la siguiente estructura:

$$L\mu + Hq + Cv \leq b \quad (1)$$

La Figura 43 a) muestra el disparo de T0 y el marcado previo, el vector de Parikh es cero, pues no se ha disparado ninguna transición antes, la Figura 43b) ilustra el disparo de T1, con el nuevo marcado, el vector de Parikh ya registra el disparo de T0 y la Figura 43 c) ilustra la red en su estado final. La inclusión del vector de Parikh permite llevar el historial de los disparos de las transiciones, lo que hace que el controlador a implementar sea dinámico [17].

Figura 43. Evolución de una RdP



Fuente: Tomado de [17]

Con el fin de diseñar un supervisor correcto para el proceso descrito en el Anexo 2, es necesario imponer previamente restricciones que eviten un comportamiento no deseado de la RdP. A continuación se describen dichas restricciones, la i corresponde al número del tanque:

1. $vT6'i + vT2'i - 2vT15'i \leq 0$: Si Inicio 1 es activado, entonces activar encendido 1 de Motobomba y encendido 1 de Electroválvula de carga
2. $vT3'i - vT2'i \leq 0$: Habilitar conteo 1 de Motobomba cuando el encendido 1 se active.
3. $vT7'i - vT6'i \leq 0$: Habilitar conteo 1 de Electroválvula cuando el encendido 1 se active.
4. $vT12'i - vT10'i \leq 0$: Si Inicio 2 es activado, entonces activar contador 1 para toma de datos
5. $vT13'i - vT12'i \leq 0$: Habilitar conteo 1 de Toma de datos cuando el encendido 1 se active.
6. $vT1'i + vT5'i - 2vT13'i \leq 0$: Si el conteo 1 para toma de datos ya finalizó, entonces activar encendido 2 de Motobomba y encendido 2 de Electroválvula de carga
7. $vT4'i - vT1'i \leq 0$: Habilitar conteo 2 de Motobomba cuando el encendido 2 se active.
8. $vT8'i - vT5'i \leq 0$: Habilitar conteo 2 de Electroválvula cuando el encendido 2 se active.
9. $vT11'i - vT78'i \leq 0$: Si Inicio 3 es activado, entonces activar contador 2 para toma de datos
10. $vT14'i - vT11'i \leq 0$: Habilitar conteo 2 de Toma de datos cuando el encendido 2 se active.

4.3.1.2. Aplicación Grafcet al caso de estudio

El Grafcet implementado está diseñado para la ejecución de los procedimientos descritos en el Anexo 2, este código utiliza y se enlaza con el código generado, a partir de CRP. El grafcet y su explicación son mostrados en el Anexo 4.

4.3.2. Implementación de LabView

Las siguientes acciones son necesarias para para la creación del panel de control y además poner en marcha el servidor de datos y el servidor web, cabe anotar que estos dos servidores fueron instalados en el mismo equipo debido a la falta de recursos, en caso de tener dos computadores, lo recomendable es instalarlos por separado con el fin de proteger la integridad de los datos según [20].

4.3.2.1. Configuración del NI OPC Servers

El servidor OPC de NI permite conectar LabView con cualquier PLC, para que cualquier cliente OPC pueda conectarse a éste servidor e intercambiar datos sin ningún tipo de restricción. Aunque no es necesario que el PLC tenga un programa ejecutándose, es conveniente hacerlo para verificar la lectura de las señales [37]. Los pasos a seguir son:

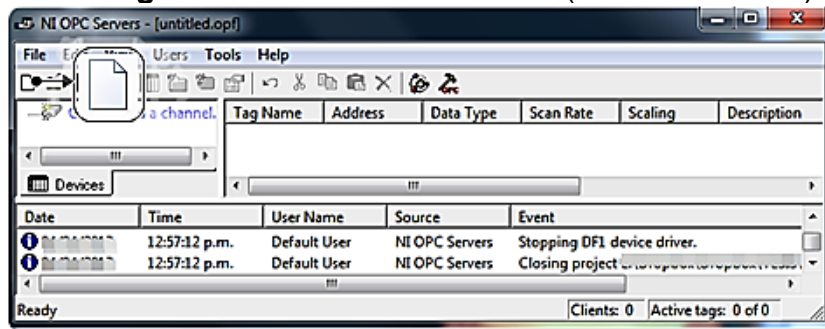
- a) Abrir NI OPC Servers seleccionando **Inicio>>Todos los programas>>National Instruments>>NI OPC Servers>> NI OPC Servers**, por defecto se abrirá la última aplicación en la que se trabajó. Ver Figura 44.
- b) Crear un nuevo archivo en **File>>New** ó haciendo clic en el ícono indicado en la Figura 45.
- c) Hacer clic en **Click to add a channel**, agregar un nombre al canal (por ejemplo **Ethernet** por indicar el modo de enlace), el cual es una forma de comunicación del computador con el hardware. Ver Figura 46.
- d) Clic en **Siguiente>**
- e) Seleccionar el driver **DF1** de la lista de drivers y clic en **Siguiente>**. Ver Figura 47.

Figura 44. Abrir NI OPC Servers



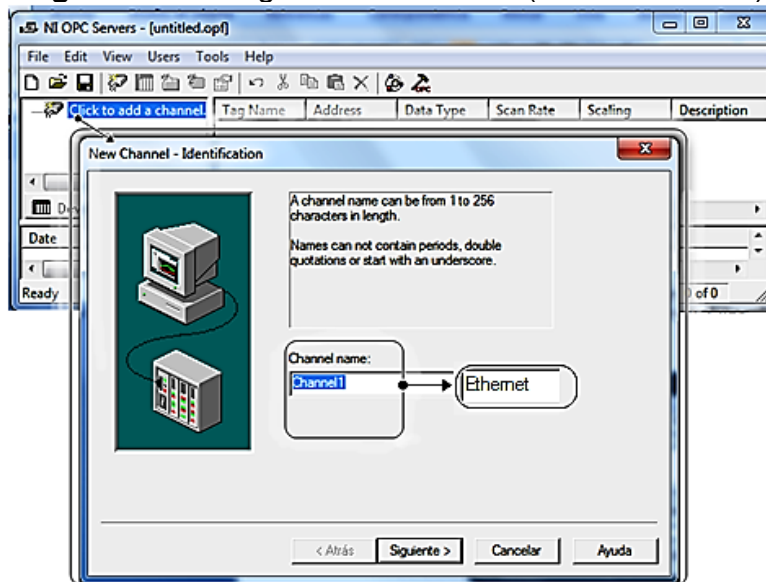
Fuente: Propia, Febrero de 2013

Figura 45. Crear nuevo archivo (NI OPC Servers)



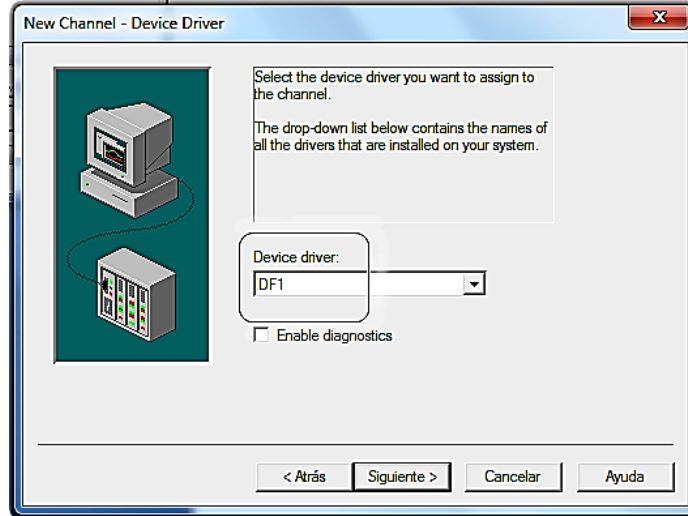
Fuente: Propia, Febrero de 2013

Figura 46. Configuración del canal (NI OPC Servers)



Fuente: Propia, Febrero de 2013

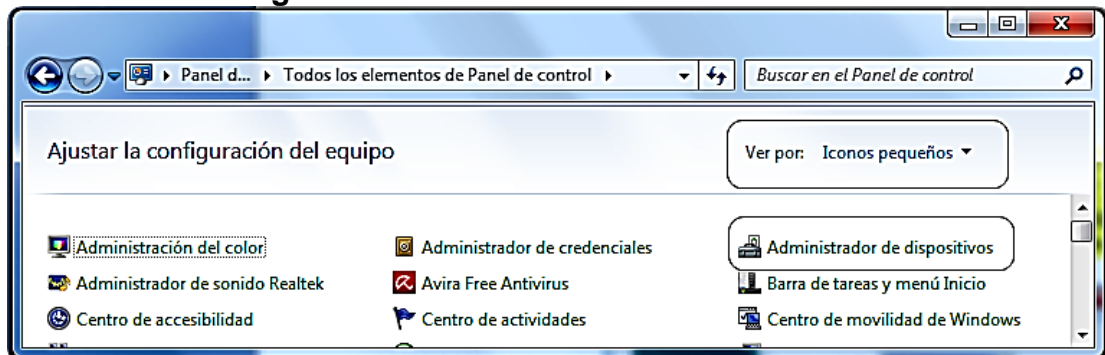
Figura 47. Configuración Driver (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- f) A continuación se configuran las **Comunicaciones**, para ello primero se debe verificar el puerto COM por el cual ha sido conectado el cable USB a serie RS232 (Ver Anexo 1), haciendo clic en **Panel de Control**, cambie la vista a Iconos pequeños y presione **Administrador de dispositivos**, en la ventana desplegada ubique el **Prolific USB-to-Serial Comm Port (COM __)**, y lea el número de COM que aparece entre paréntesis. Ver Figura 48.

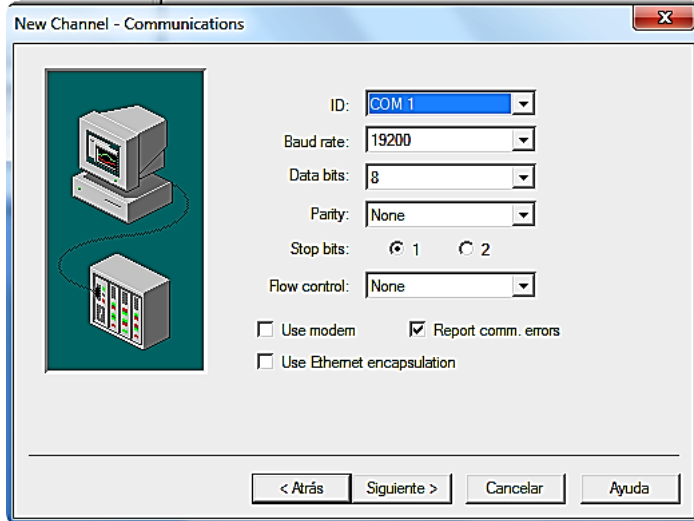
Figura 48. Verificación de Puerto COM



Fuente: Propia, Febrero de 2013

- g) El ID se debe configurar en el COM adecuado, la velocidad de transmisión en “19200 Baudios”, el tamaño de la palabra en “8 bits”, sin paridad, con un bit de stop y habilitada la opción de reporte de los errores de comunicación, clic en **Siguiente>**. Ver Figura 49.

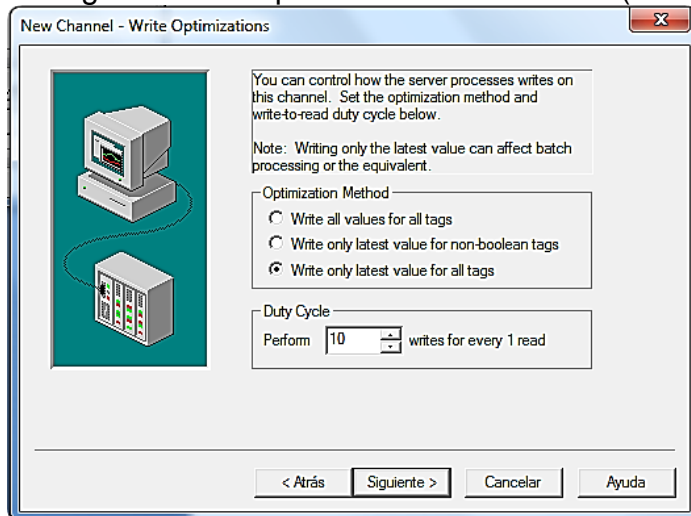
Figura 49. Configuración de comunicaciones (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- h) En la siguiente pantalla se configuran las opciones de optimización de escritura en donde se dejan los valores predeterminados, clic en **Siguiente>**.

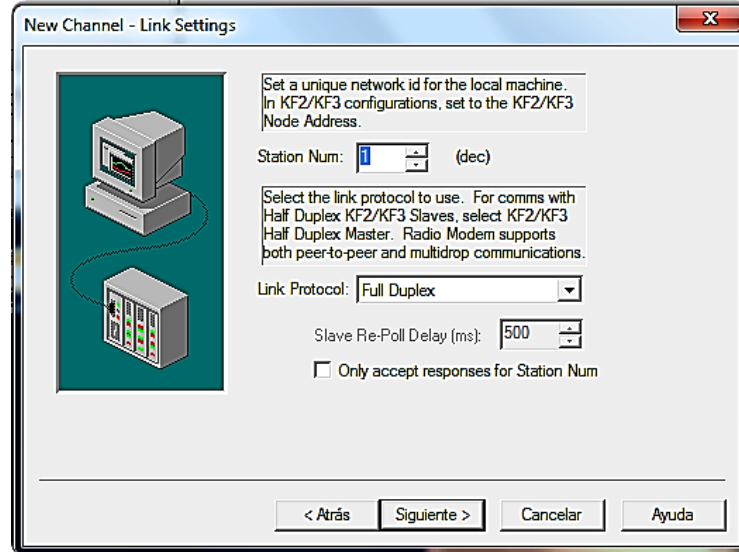
Figura 50. Configuración de optimización de escritura (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- i) Por último se configuran las opciones del enlace, se asigna un identificador de red para la máquina local o la dirección del nodo, en este caso "*Station Num: 1*" y el protocolo de enlace debe ser "*Full Dúplex*". Clic en el botón **Siguiente>**. Ver Figura 51.

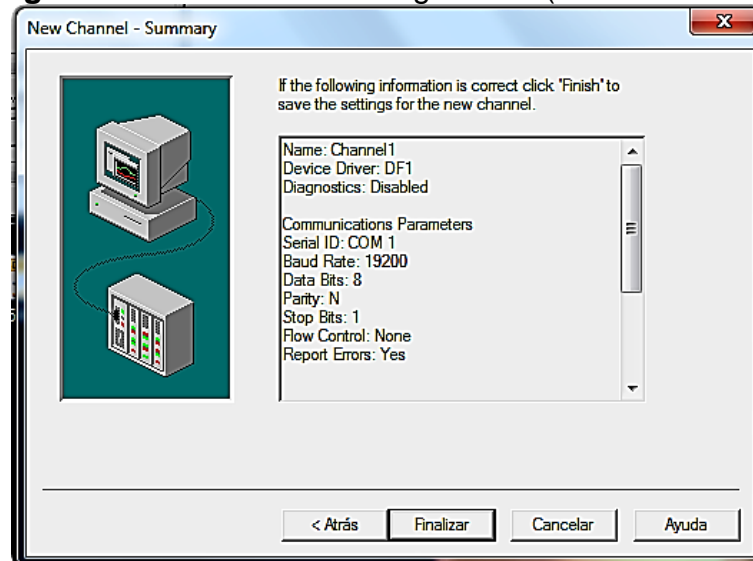
Figura 51. Configuración opciones de enlace (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- j) Por último, se presenta un resumen de la configuración y se finaliza el asistente, ver Figura 52

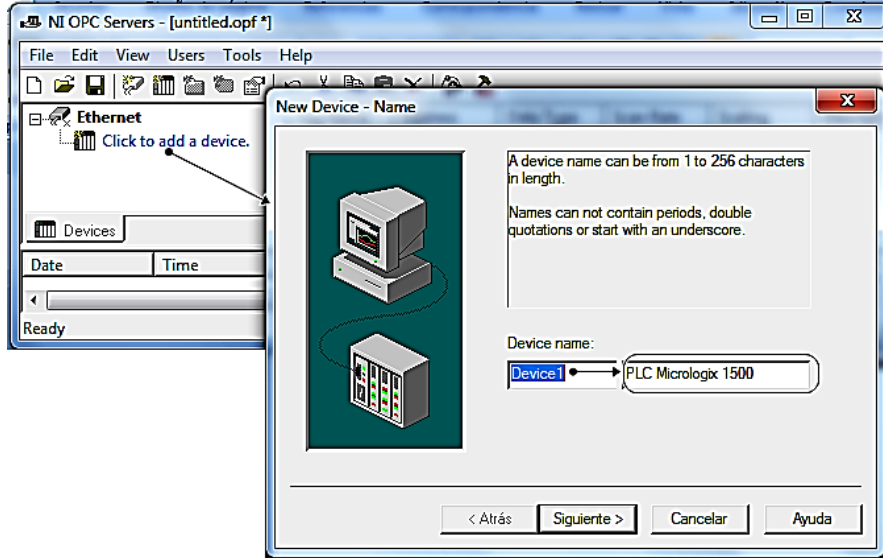
Figura 52. Resumen de configuración (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- k) Una vez se tenga el canal creado y configurado se debe crear un “device” que este caso es el PLC (*MicroLogix 1500*). Ver Figura 53.

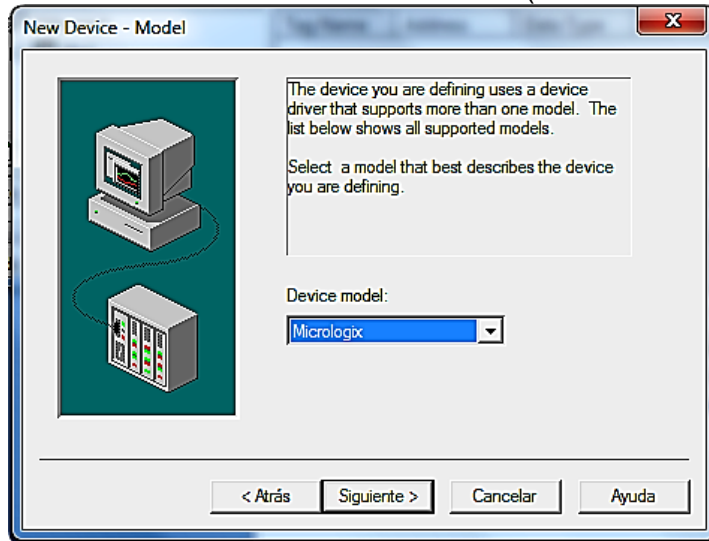
Figura 53. Creación del Device (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- I) Seleccionar el modelo del PLC como **Micrologix**. Clic en el botón **Siguiente>**. Ver Figura 54.

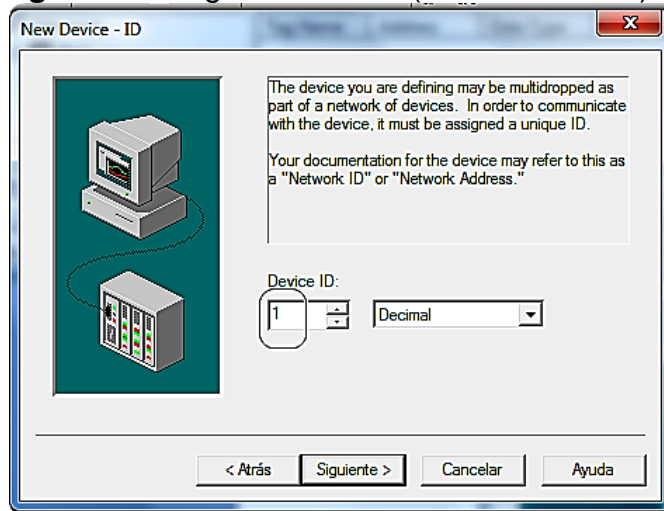
Figura 54. Selección del modelo del PLC (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- m) Asignar un ID al Device, en este caso 1. Ver Figura 55.

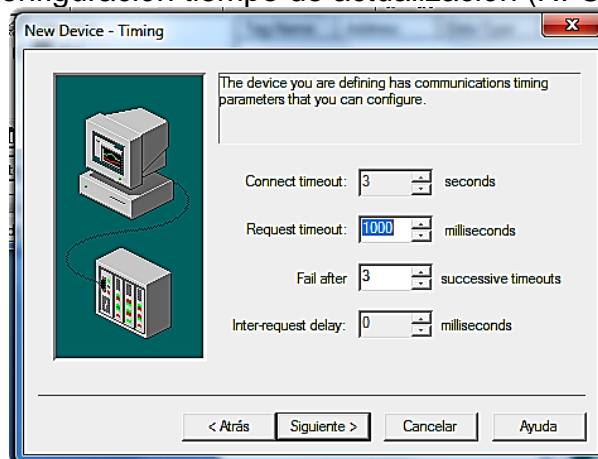
Figura 55. Asignación de ID (NI OPC Servers)



Fuente: Propia, Febrero de 2013

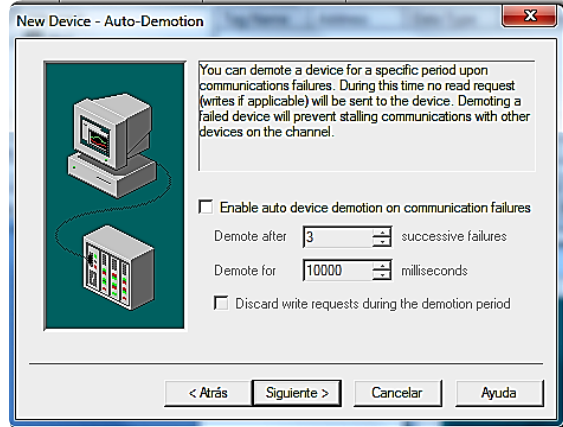
- n) En las siguientes pantallas conservar los valores por defecto, que hacen referencia al tiempo de refresco de los datos (Ver Figura 56) y a los reintentos de establecer la conexión en caso de pérdida de la misma antes de indicar error de comunicaciones, para esto se puede activar la opción **Auto Demotion** para que el driver pueda intentar reconectar el dispositivo en caso de una pérdida de comunicación (Ver Figura 57). Hacer clic en **Siguiente>** y por último en **Finalizar**, Ver Figura 58, Figura 59 y Figura 60.

Figura 56. Configuración tiempo de actualización (NI OPC Servers)



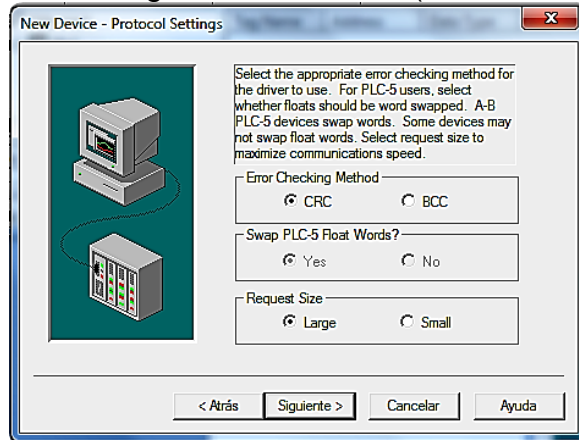
Fuente: Propia, Febrero de 2013

Figura 57. Opción de reconectar el dispositivo (NI OPC Servers)



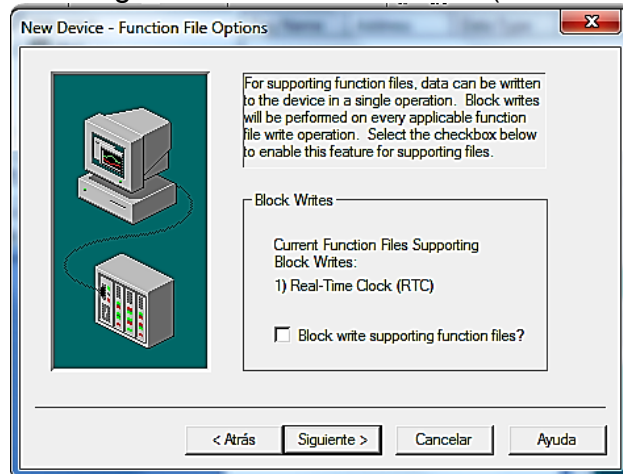
Fuente: Propia, Febrero de 2013

Figura 58. Configuración adicional (NI OPC Servers)



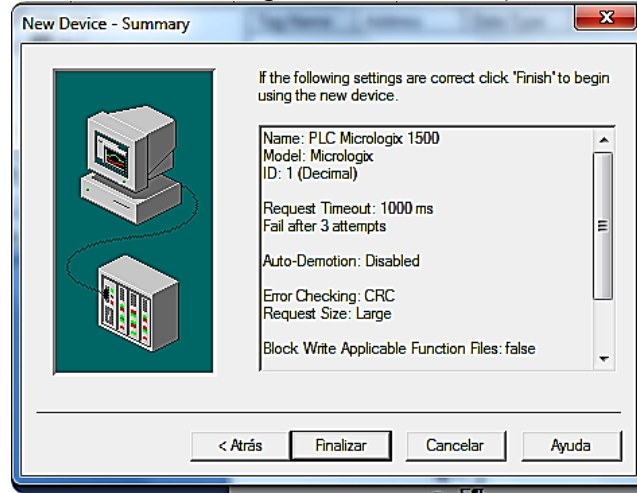
Fuente: Propia, Febrero de 2013

Figura 59. Configuración Real Time Clock (NI OPC Servers)



Fuente: Propia, Febrero de 2013

Figura 60. Resumen configuración Device (NI OPC Servers)

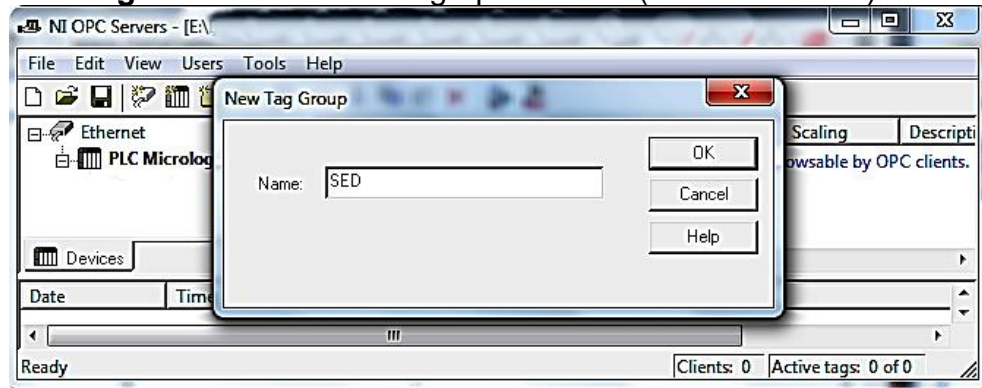


Fuente: Propia, Febrero de 2013

Hasta este punto ya se encuentra configurada la comunicación del PLC Micrologix 1500 con el computador mediante OPC; es decir que desde un cliente OPC (En este caso LabView) se pueden monitorear las entradas, salidas y parámetros del PLC.

- o) A continuación se crea un grupo de variables, dando clic derecho en el PLC creado, en este caso **PLC Micrologix 1500**, en la ventana colocar un nombre al grupo (**SED**). Ver Figura 61.

Figura 61. Creación de grupo e TAGs (NI OPC Servers)



Fuente: Propia, Febrero de 2013

- p) Sobre la carpeta creada anteriormente (SED), hacer clic derecho y en la ventana desplegada hacer clic en **New tag...** complete los campos según la Tabla 12, un ejemplo es mostrado en la Figura 62.

Tabla 12. Tags utilizadas en la implementación del LLR

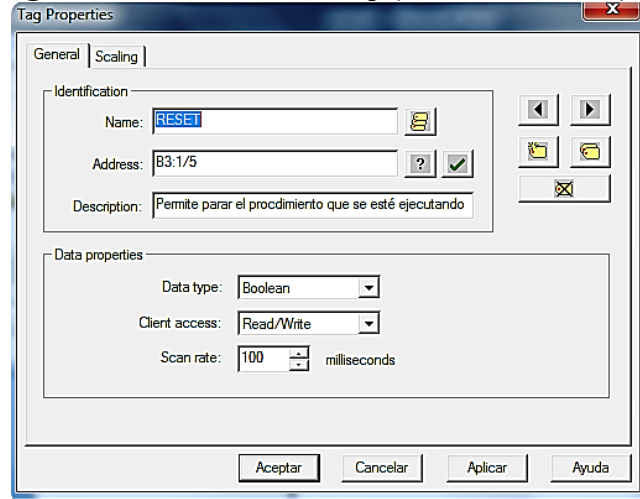
Name	Adress	Data Type	Client Access
Estab2	B3:5/6	Boolean	Read
Estab3	B3:5/7	Boolean	Read
Estab4	B3:5/8	Boolean	Read
Estabilización	B3:5/1	Boolean	Read
Intk2	O:0/3	Boolean	Read
Intk3	O:0/4	Boolean	Read
Intk4	O:0/6	Boolean	Read
Intk5	O:0/7	Boolean	Read
Luztomdat	B3:5/2	Boolean	Read
Outtk2	O:0/2	Boolean	Read/Write
Outtk3	O:0/1	Boolean	Read/Write
Outtk4tk5	O:0/5	Boolean	Read/Write
Proced1	B3:5/0	Boolean	Read/Write
Proced2	B3:5/3	Boolean	Read/Write
Proced3	B3:5/4	Boolean	Read/Write
Proced4	B3:5/5	Boolean	Read/Write
Pump_On	O:0/10	Boolean	Read/Write
Pumpoff	B3:3/11	Boolean	Read
Reset	B3:1/5	Boolean	Read/Write
Td1tk2	T4:0.Acc	Word	Read
Td1tk3	T4:1.Acc	Word	Read
Td1tk4	T4:2.Acc	Word	Read
Td1tk5	T4:3.Acc	Word	Read
Td2tk2	T4:4.Acc	Word	Read
Td2tk3	T4:5.Acc	Word	Read
Td2tk4	T4:6.Acc	Word	Read
Td2tk5	T4:7.Acc	Word	Read

Fuente: Propia, Febrero de 2013

- q) Para confirmar que el PLC se esté comunicando con el servidor OPC correctamente se puede lanzar el cliente OPC rápido. Presionar el botón **Quick OPC Client** como se indica en la Figura 62.
- r) En el árbol de la parte superior izquierda de la pantalla seleccionar el canal y el dispositivo que se ha configurado; en este caso, **Ethernet.PLC.Micrologix 1500.SED**. En la parte superior derecha de la pantalla aparecerá la variable con su valor y parámetros de configuración, allí se debe verificar que la **Quality** este **Good**, puede cerrar el Quick OPC

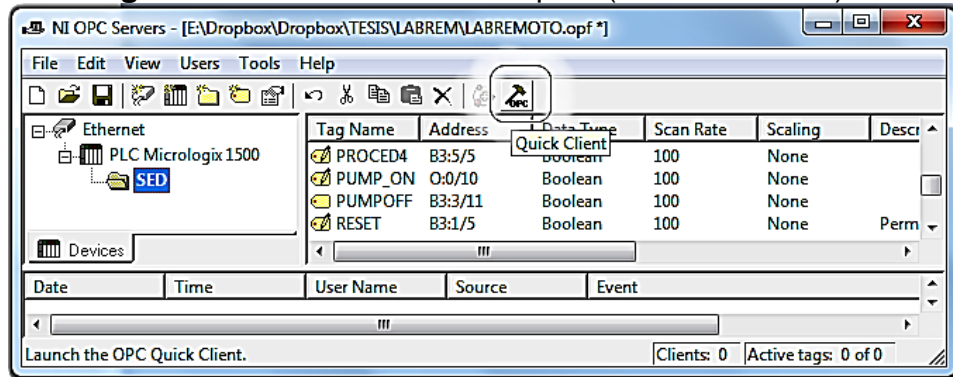
Client sin guardar la configuración. Y finalmente guardar el proyecto y cerrar el servidor. Ver Figura 64.

Figura 62. Creación de tag (NI OPC Servers)



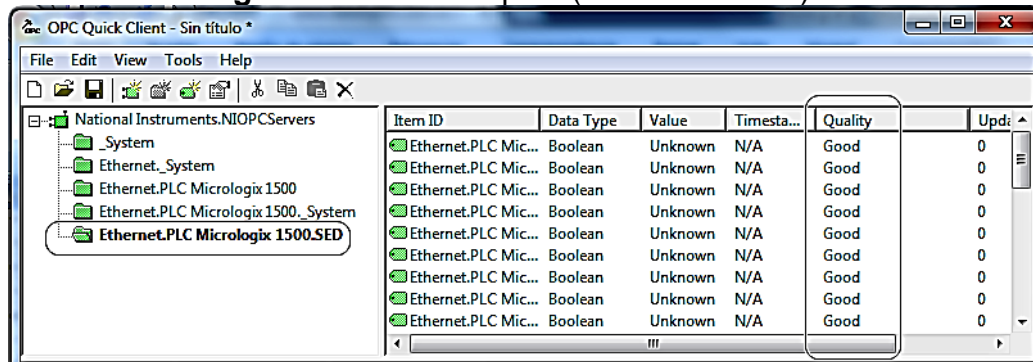
Fuente: Propia, Febrero de 2013

Figura 63. Abrir cliente OPC rápido (NI OPC Servers)



Fuente: Propia, Febrero de 2013

Figura 64. Cliente rápido (NI OPC Servers)



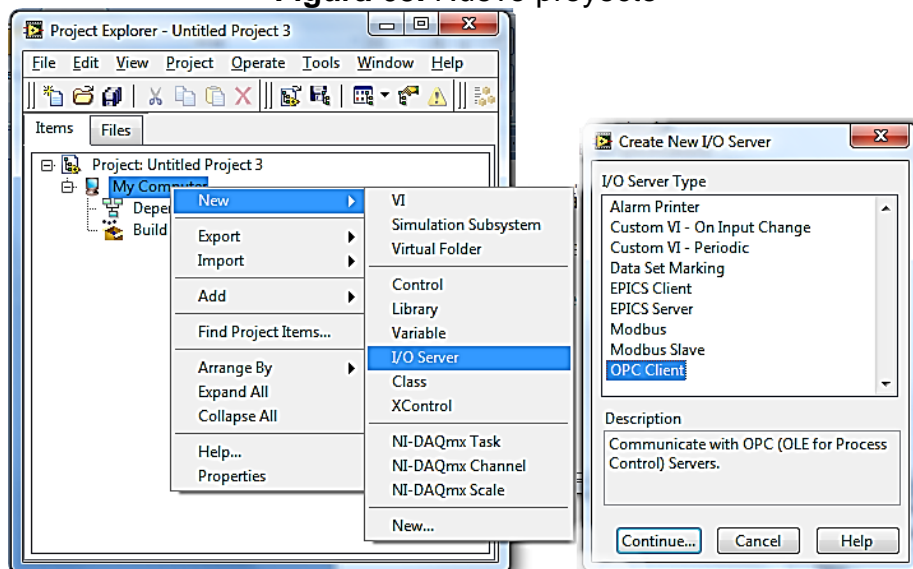
Fuente: Propia, Febrero de 2013

4.3.2.2. Creación del panel remoto

Una vez ha sido configurado el servidor OPC, ya se puede conectar cualquier cliente para monitorear el PLC MicroLogix 1500, en este caso se usará LabView como cliente OPC. Los pasos son:

- a. Ejecutar **National Instruments LabView 2010 (32-bits)**, que se encuentra en **Inicio>>Todos los programas**, hacer clic en **File >> New Project**. En la nueva pantalla hacer clic derecho sobre **My Computer**, en **New >> I/O Server**. En la ventana que aparece seleccionar el tipo de enlace que se usará entre el PC y el PLC; **OPC Client>> Continue...**Ver Figura 65.

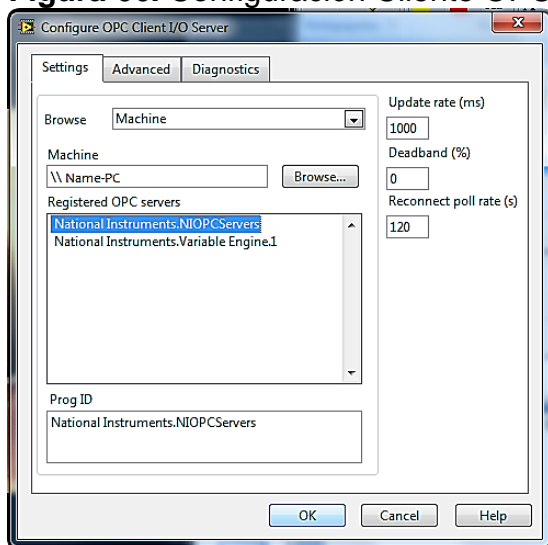
Figura 65. Nuevo proyecto



Fuente: Propia, Febrero de 2013

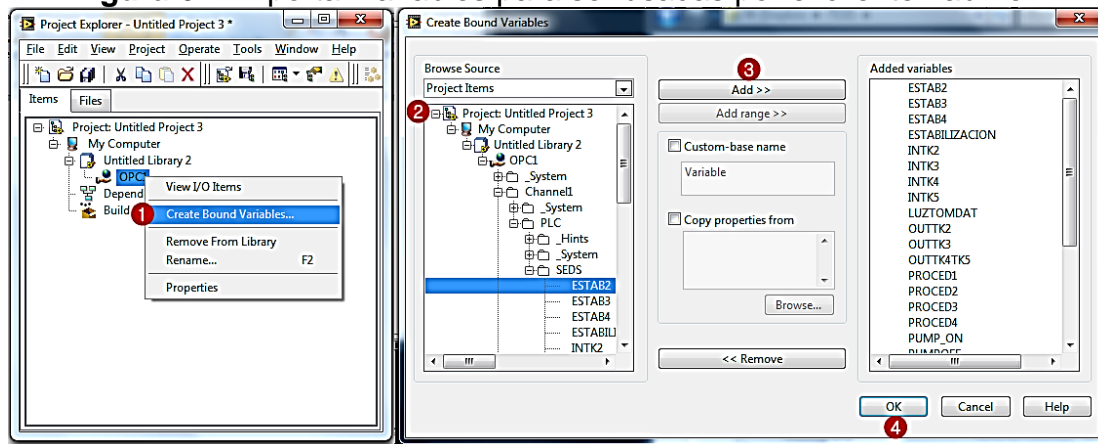
- b. Seleccionar la Máquina mediante **Browse...** escoja el nombre de su equipo, y **NationalInstruments.NIOPCServers**. Ver Figura 66.
- c. Se crea un nuevo ícono **OPC**, hacer clic en **Create Bound Variables...(1)**, en la nueva ventana (2) clic en el **(+)** del proyecto creado **Project Untitled Project 3 >> (+) My Computer >> (+) Untitled Library 2 >> (+) OPC1 >> (+) Channel 1>> (+) PLC >> (+) SEDS**, seleccionar todas las variables y hacer clic en **Add>> (3)**, verificar que todas fueron agregadas y dar clic en **OK (4)**, en la ventana que aparece clic en **Done**. Ver Figura 66.

Figura 66. Configuración Cliente OPC



Fuente: Propia, Febrero de 2013

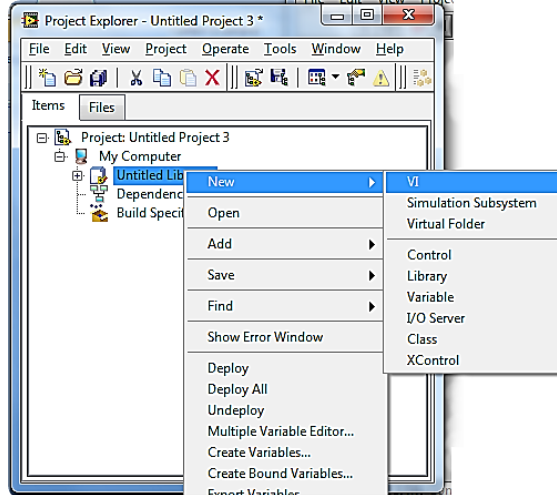
Figura 67. Importar variables para ser usadas por el cliente LabView



Fuente: Propia, Febrero de 2013

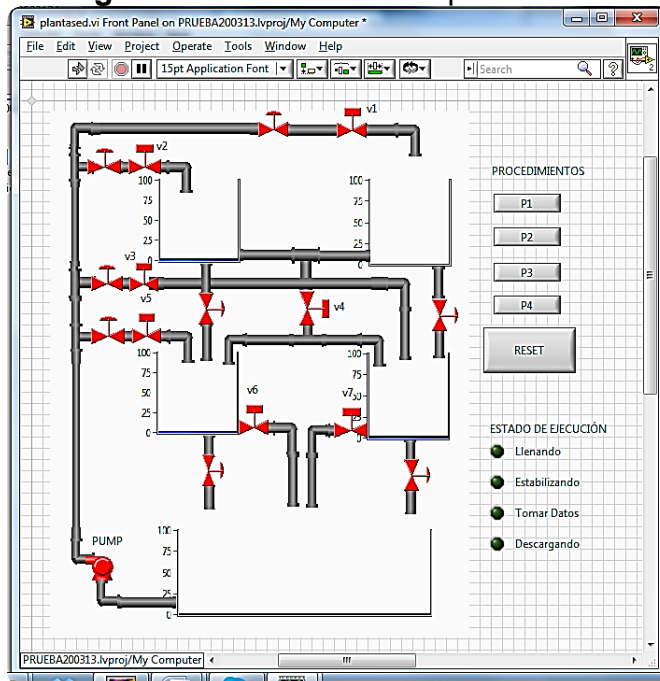
- d. Para crear el VI, que luego será publicado, hacer clic derecho en **Untitled Library >> New >> VI**. Ver Figura 68.
- e. En el panel frontal realizar el esquema indicado en la Figura 68, el cual consiste en el diagrama de la planta, cuatro botones para ejecutar los procedimientos, un botón de RESET y cuatro indicadores del estado de ejecución.

Figura 68. Crear un VI



Fuente: Propia, Febrero de 2013

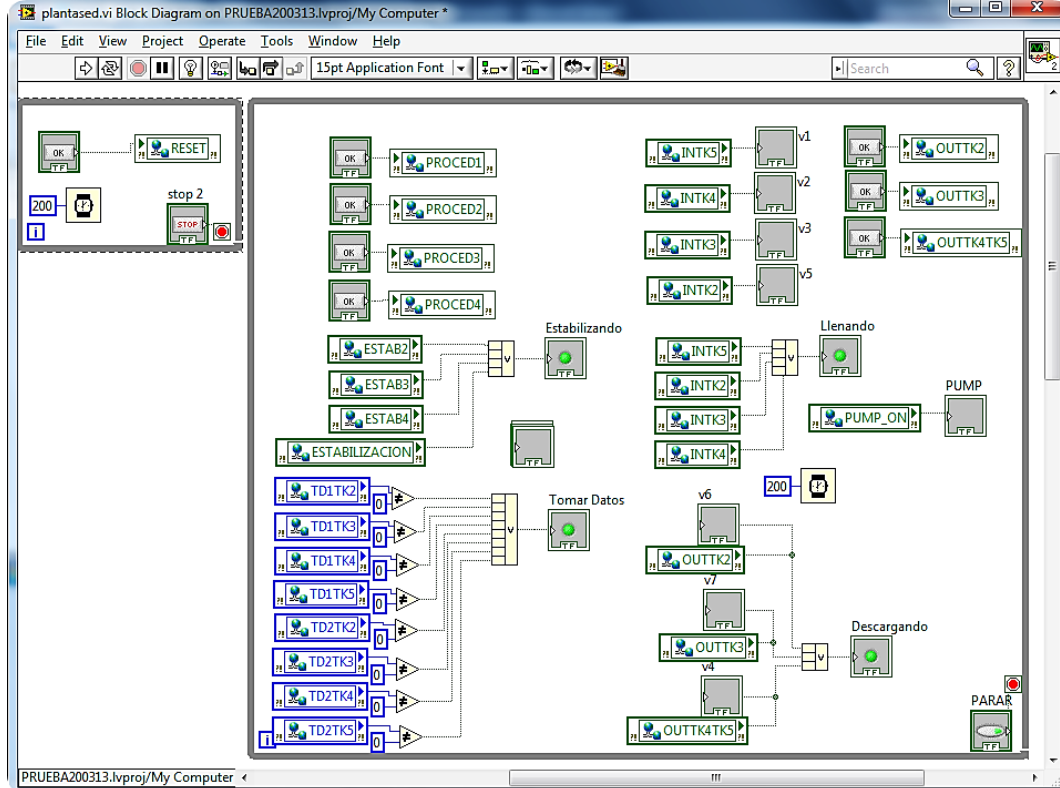
Figura 69. Panel frontal implementado



Fuente: Propia, Febrero de 2013

- f. La conexión de variables con cada uno de los elementos del panel frontal debe realizarse en el diagrama de bloques como es indicado en la Figura 70. Un manual que explica cómo realizar estos diagramas se puede encontrar en [38].

Figura 70. Diagrama de bloques implementado



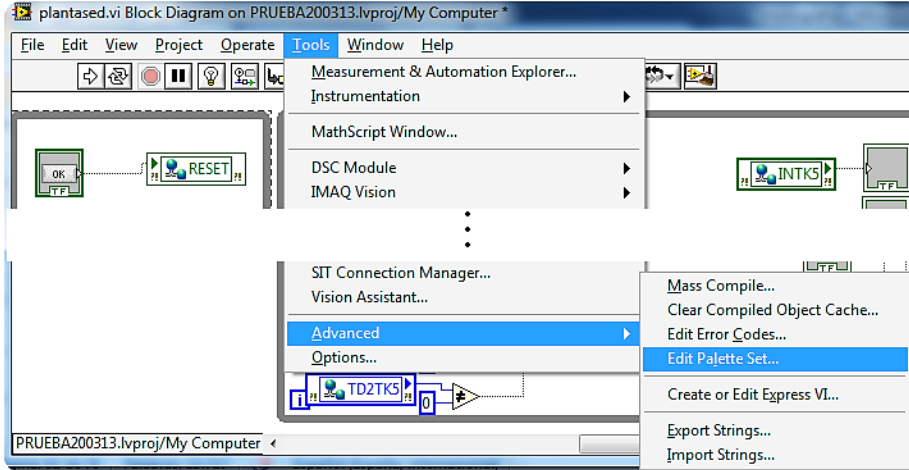
Fuente: Propia, Febrero de 2013

4.3.2.3. Cámaras

Para el funcionamiento de las cámaras es necesario un componente de Labview llamado IMAQ USB, el cual no viene incluido en el paquete del software, por lo tanto se hace necesario instalarlo y agregarlo a las opciones disponibles en el diagrama de bloques, a continuación se describe el proceso:

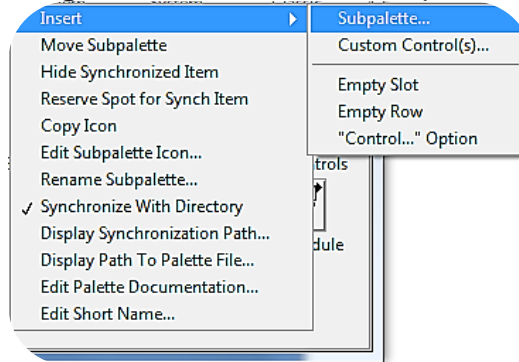
- a. Ejecutar el instalador de **ni_imaq_usb_installer_86**, disponible en la página de labview en [39], seguir los pasos que aparezcan en pantalla y descargar también el componente **IMAQ USB** disponible en [40], y guardar el archivo .llb en la ruta **National Instruments\LabVIEW 2010\vi.lib\addons**.
- b. En el diagrama de bloques hacer clic en **Tools >> Advanced >> Edit paletted set....** Ver Figura 71.
- c. Entre las ventanas que aparecen, ubicar la de **Funcions >> addons**, a continuación clic en **Insert >Subpalette...** Ver Figura 71.

Figura 71. Editar una paleta de opciones



Fuente: Propia, Febrero de 2013

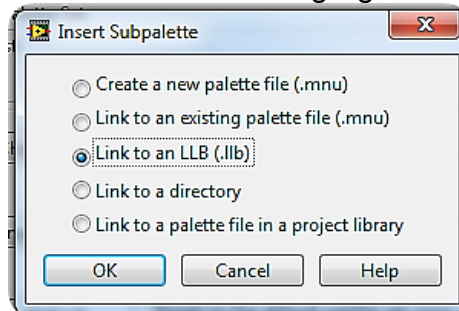
Figura 72. Insertar una subpaleta



Fuente: Propia, Febrero de 2013

d. Escoger **Link to an LLB (.llb)** ya que en este formato se encuentra Imaq USB. Ver Figura 73.

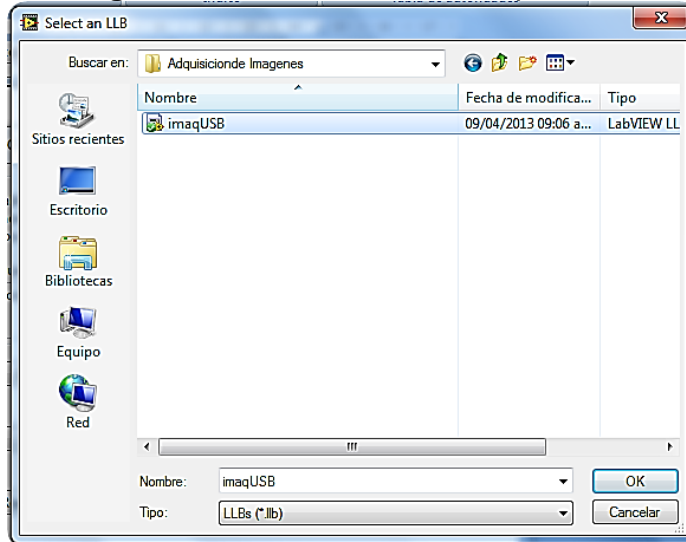
Figura 73. Extensión de VI a agregar en sub-paleta



Fuente: Propia, Febrero de 2013

- e. Dar la ubicación del archivo (**C:\National Instruments\LabVIEW 2010\vi.lib\addons**).

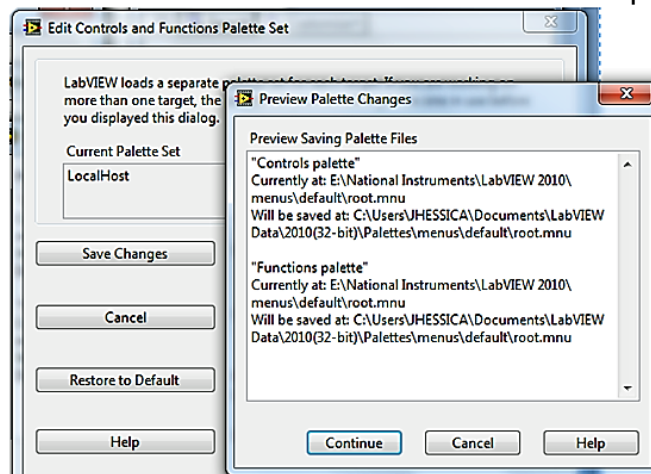
Figura 74. Ubicar archivo imaqUSB.llb



Fuente: Propia, Febrero de 2013

- f. Por último en la pantalla de **Edit Controls and Functions Palette Set>>Save Changes>> Continue**. Con este procedimiento el VI de ImaqUSB ya estará disponible si se hace clic derecho sobre el diagrama de bloques, en addons. Ver Figura 75.

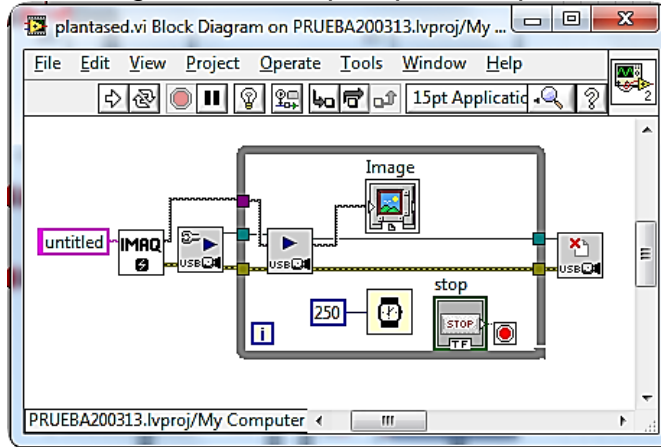
Figura 75. Finalización de modificaciones en la paleta.



Fuente: Propia, Febrero de 2013

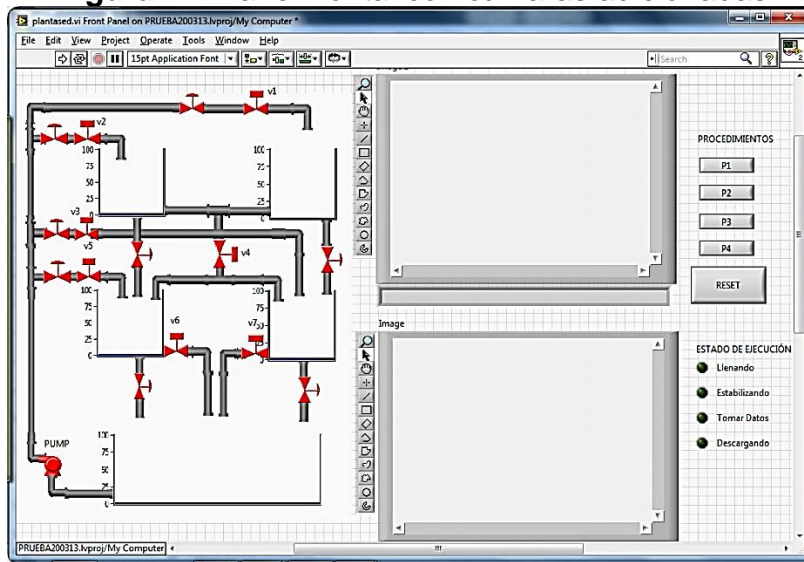
- g. Para crear la imagen del panel frontal y el diagrama de bloques el lector se podrá remitir a [41] o ver un video tutorial en [42]. Ver Figura 76 y Figura 77.

Figura 76. Diagrama de bloques para Adquisición de video



Fuente: Propia, Febrero de 2013

Figura 77. Panel frontal con cámaras adicionales



Fuente: Propia, Febrero de 2013

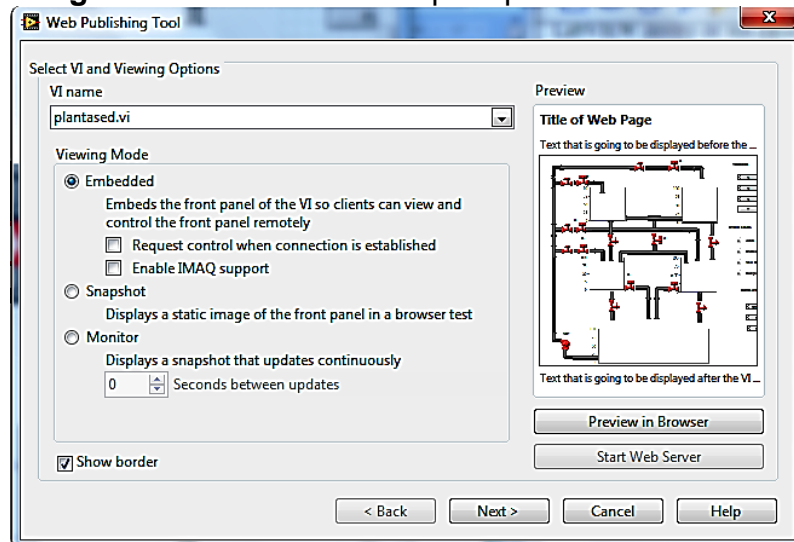
4.3.2.4. Puesta en marcha del Servidor Web del VI

Dentro de las características de Labview, se encuentra el Servidor Web a través del cual se publica el VI creado anteriormente, en internet, para hacer uso de este servicio se procede de la siguiente manera:

- a. En la barra de herramientas, hacer clic en **Tools >> Web Publishing Tool**, con esto se obtiene en pantalla la ventana indicada en la Figura 78, en la cual se debe nombrar el VI creado anteriormente, y se escoge la opción **Embedded**, con el fin de que el VI pueda ser visto y controlado de

forma remota por los estudiantes e investigadores, debido a que **Snapshot** solo muestra una imagen instantánea del VI y **Monitor** también tiene la función de Snapshot pero en este caso permite además, actualizar continuamente de acuerdo a la configuración previamente ingresada. Clic en **Next>**.

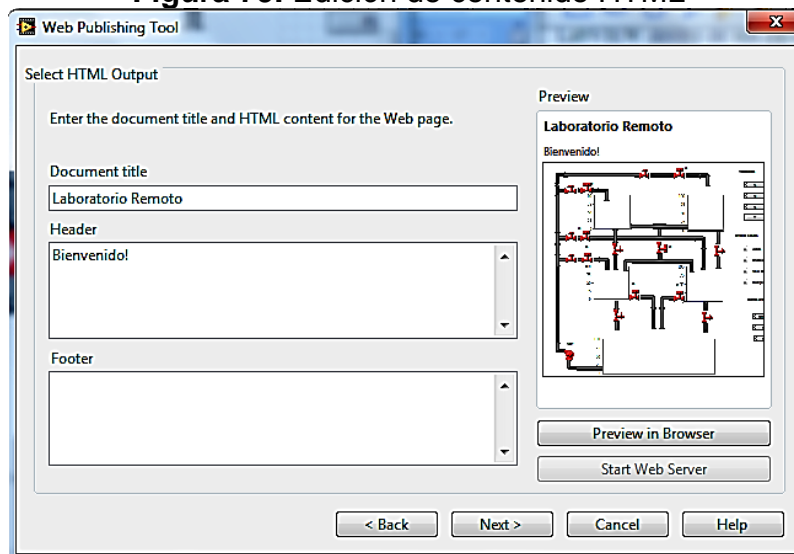
Figura 78. Selección de VI para publicar en internet



Fuente: Propia, Febrero de 2013

b. A continuación se puede escribir un título, encabezado y pie de página para la publicación, en la Figura 79 se muestra el sub-menú con estas configuraciones. Clic en **Next>**.

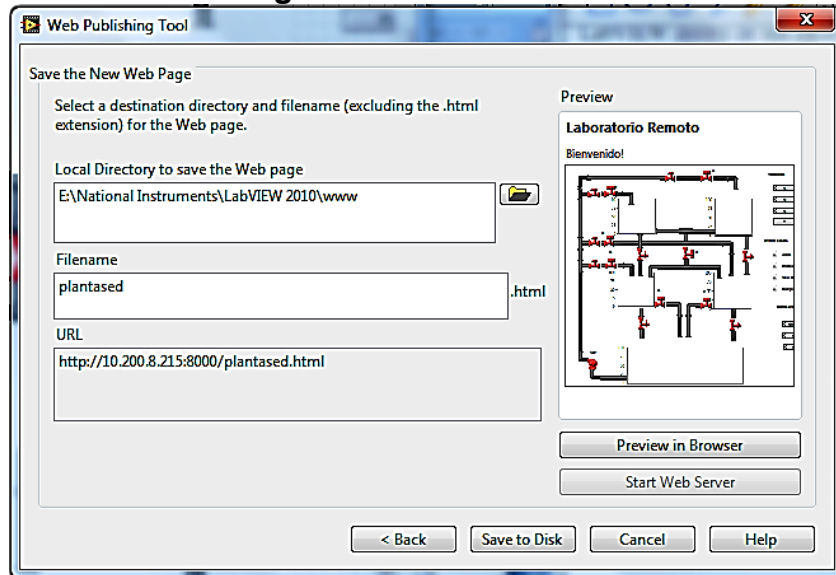
Figura 79. Edición de contenido HTML



Fuente: Propia, Febrero de 2013

- c. El nombre de la página web tendrá extensión html, y se puede editar en la ventana que aparece a continuación, ver Figura 80, en la cual se recomienda conservar la dirección donde es guardada, el **Filename** se puede modificar.

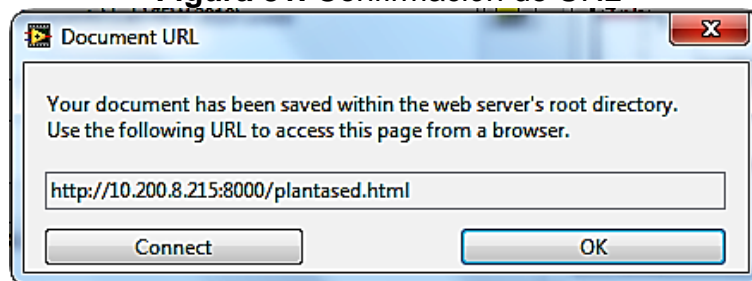
Figura 80. Dirección URL



Fuente: Propia, Febrero de 2013

- d. Finalmente aparece una ventana confirmando la dirección URL, si desea conectarse clic en **Connect**, de no ser así guarde este dato para ingresar después y de clic en **OK**. Ver Figura 81.

Figura 81. Confirmación de URL

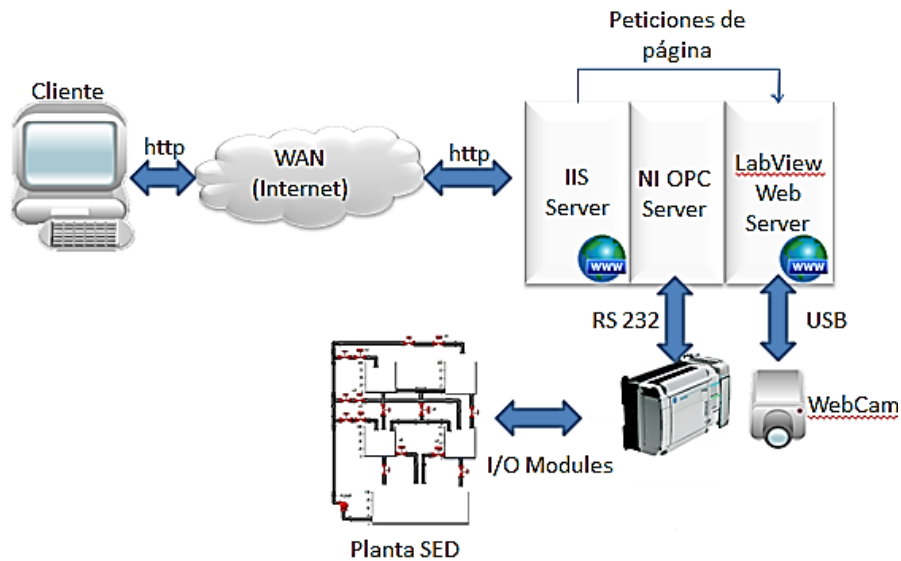


Fuente: Propia, Febrero de 2013

4.3.3. Arquitectura de la red

Para el funcionamiento de SEDLAB, se requieren un conjunto de elementos hardware y software que serán detallados a continuación, la arquitectura de la red se muestra en la Figura 82

Figura 82. Arquitectura de la Red



Fuente: Propia, Marzo de 2013

- **Cliente:** Son los usuarios que se conectan remotamente a SEDLAB y acceden dependiendo de los permisos al sistema de gestión de acceso y reserva de horarios y/o al panel remoto, entre otros.
- **Servidor Web Labview:** Se encarga de contener y publicar el panel remoto, que será incrustado en las páginas del sistema de acceso y reserva. Además permite la realimentación visual de la Planta física mediante cámara web. Este servidor también tiene la funcionalidad de comportarse como un cliente para el Servidor NI OPC.
- **Servidor Web IIS:** Se encarga de contener y publicar la aplicación que permite asignar permisos a los usuarios y otras funcionalidades asociadas al sistema de acceso y reserva de horarios, además realiza peticiones de página al Servidor Web Labview para permitir al cliente la interacción con el panel remoto.
- **Servidor NI OPC:** El servidor OPC de NI permite conectar LabView con el PLC Micrologix 1500, para que el cliente OPC (Servidor Web Labview) pueda conectarse a éste servidor e intercambiar datos sin ningún tipo de restricción, lo que quiere decir que es un intermediario entre el Servidor Web Labview y el PLC.

5. PRUEBAS E INTEGRACIÓN

Para garantizar el correcto funcionamiento del sistema, fueron necesarias un conjunto de pruebas para determinar si las soluciones propuestas estaban bien encaminadas. En este caso se destacarán las pruebas que se hicieron a los elementos que se consideran más importantes, y estos son: el código ladder descargado en el PLC, el panel de control remoto y el sistema de acceso y reserva de horarios.

5.1. Pruebas de validación del Supervisor

La simulación que se realiza en esta sección, permite validar si el sistema satisface las **Restricciones físicas lógicas o de control** que fueron creadas anteriormente, con el objetivo de evitar estados no deseados de la planta y lograr el procedimiento descrito en el Anexo 2. Las pruebas realizadas fueron consignadas en el Anexo 7.

5.2. Pruebas para el panel remoto

Esta prueba consistió en probar la comunicación entre el panel remoto creado utilizando la herramienta de LabView y la planta caso de estudio. Con esta prueba se logró la evaluación de dos aspectos importantes para el correcto funcionamiento del SEDLAB de forma remota. Estos son

- Comunicación entre el panel remoto y la planta

Este procedimiento permitió probar las acciones de lectura y escritura desde el panel remoto hacia el PLC Micrologix 1500, el cual controla las acciones que se realizan sobre la planta SED. El resultado de esta prueba fue satisfactorio, pues, fue posible llevar a cabo la ejecución de cada uno de los procedimientos descritos en el Anexo 2 de forma correcta.

- Observar el retardo entre una acción en el panel remoto y la respuesta del recurso físico.

Con la ejecución de los procedimientos desde panel remoto se buscaba observar el tiempo de retardo que existe entre una acción en el panel y su respuesta en la planta de SED.

Este proceso arrojó dos resultados importantes, el primero tiene que ver con que el retardo no es apreciable por el usuario cuando ejecuta acciones sobre los mandos del panel remoto, sin embargo el retorno del video desde las cámaras USB ubicadas en planta y el panel remoto dependía mucho de la calidad de la conexión a internet principalmente del lado del servidor, por lo tanto se optó por realizar estas pruebas en horarios donde la red de la Universidad del Cauca presentara menos congestión, en consecuencia el retorno del video mejoró considerablemente.

5.3. Pruebas para el sistema de acceso y reserva

Esta prueba consistió en realizar los procedimientos de registro de usuario, asignación de permisos de acuerdo al rol del mismo, crear, editar o eliminar una reserva para hacer uso del SEDLAB, cambio de contraseñas, y ejecución de una práctica teniendo en cuenta el horario de una reserva solicitada.

La prueba se realizó de la siguiente manera y para los dos tipos de usuario definidos para el sistema (Administrador / usuario):

- a) Registro de usuarios ingresando a la aplicación web para acceso y reserva.
- b) Gestión de horarios para la realización de las practicas sobre el SEDLAB.
- c) Asignación de permisos a los usuarios.
- d) Crear, editar o eliminar reservas para los usuarios autorizados.
- e) Cambiar y recuperar contraseñas de usuarios.

Los resultados obtenidos durante la prueba fueron positivos pues la aplicación diseñada para el sistema de acceso y reserva se desempeñó dentro de los parámetros esperados.

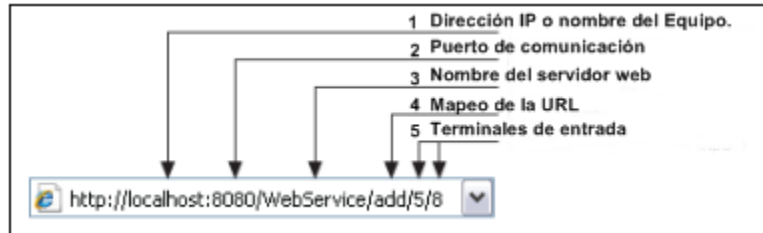
5.4. Pruebas de integración

La integración consiste en que la aplicación que presta los servicios de gestión relacionados con el acceso y la reserva de horarios, permita albergar la aplicación que despliega LabView a través de una página web HTML para que un usuario después de realizar el proceso de registro y asignación de reserva, sea capaz de ejecutar el panel remoto que interactúa con la planta de SED.

Para lograr esto se realizó la comunicación por URL. Los VI's pueden recibir parámetros por URL cuando éstos están actuando bajo el servidor

LabVIEW. Los resultados fueron satisfactorios pues se observó que al pasarle información mediante la URL, el servidor devolvía el panel remoto. En la Figura 83 se describe la información y los pasos necesarios para comunicarse con un VI que se ha desplegado como un servicio Web.

Figura 83. Petición al servidor para invocar un VI



Fuente: Propia. Abril de 2013. Basado en [34]

1. HTTP usa URLs para enviar una solicitud de un cliente a un servidor específico. La primera parte de la dirección HTTP especifica la ubicación física del servidor Web.
2. Puerto del servidor de aplicaciones Web, por defecto el puerto 8080 se asigna al servicio Web.
3. Como se pueden configurar múltiples servicios para una aplicación LabVIEW, es necesario enviar en la URL el parámetro del nombre del servicio.
4. Se pueden definir rutas personalizadas para cada servicio que se asignan a Vis.
5. Se pueden pasar los terminales de las entradas para invocar el VI.

5.5. Pruebas de funcionamiento general

Para probar el funcionamiento general del sistema, fue necesario realizar pruebas de tal manera que los subsistemas actúen como uno solo y de forma deseada. Esto se realizó de la siguiente manera:

- a) Registro de administrador ingresando a la aplicación para acceso y reserva.
- b) Gestión de horarios para la realización de las prácticas sobre el SEDLAB.
- c) Registro de usuario ingresando a la aplicación para acceso y reserva.
- d) Asignación de permisos a los usuarios por parte del administrador.
- e) Crear reserva para los usuarios autorizados.
- f) Puesta a punto del SEDLAB para su ejecución remota.
- g) Ingreso al sistema de acceso y reserva en hora y fecha establecida en la reserva.
- h) Ejecución de una práctica remota usando SEDLAB.

Esta prueba de funcionamiento se realizó en diferentes ocasiones y en determinado tipo de situaciones, los resultados fueron satisfactorios aunque en ocasiones se presentaron fallos en la ejecución del panel remoto, esto debido a que el navegador web del lado del cliente no recibió respuesta del servidor de LabView, sin embargo la causa de este fallo se debe posiblemente a la calidad de la conexión a internet del equipo servidor, pues, en las pruebas realizadas en horarios extra clase cuando el uso de la red al interior de la universidad es mucho menor, las pruebas no presentaron este fallo.

6. CONCLUSIONES, TRABAJOS FUTUROS Y LIMITACIONES

6.1. Conclusiones

- El objetivo general de este trabajo es “Implementar un sistema que permita acceder, controlar y manipular la planta de SED del LCP, a través de internet.”, durante el desarrollo del mismo se realizó el diseño e implementación de un sistema que permite acceder a la planta SED a través de internet por medio de una aplicación desarrollada en ASP.NET que cumple las funciones de registro y reserva de horarios antes de dar acceso al panel remoto de la planta en mención, la cual puede ser controlada y manipulada de forma remota por medio de una interfaz de usuario creada en LabView 2010.
- Las RdP son una herramienta valiosa para el modelado de Sistemas Dinámicos de Eventos Discretos en los cuales es de suma importancia garantizar que el sistema no llegue a estados no deseados, pero a medida que la complejidad de un proceso aumenta, las restricciones aumentan y se pueden generar nuevos estados no deseados que pueden ser difíciles de identificar.
- El software CRP utilizado para la generación del código ladder que se utilizó para realizar la práctica de forma remota demostró ser eficiente, sin embargo esta herramienta solo permite la generación automática de código ladder para PLC's que son programables mediante RSlogix.
- El uso de LabView y las herramientas que este paquete tecnológico utiliza facilitan el desarrollo de un LRR, pues, permiten adquirir los datos desde el experimento real hasta una interfaz de usuario que posteriormente puede ser publicada en la web de forma eficiente.
- La realimentación visual en el SEDLAB es el componente que mayor ancho de banda exige, por lo que la transmisión de video o imágenes debe realizarse empleando formatos de compresión y limitando la frecuencia con la que se envía imágenes.
- A lo largo del desarrollo del SEDLAB se tuvieron en cuenta aquellos aspectos más relevantes con el fin de lograr los objetivos planteados en el proyecto, sin embargo, en relación con los LRR queda mucho camino por recorrer. Pues estas herramientas tecnológicas pueden llegar a ser tan poderosas como sea necesario, pero su desarrollo es un trabajo que

demanda la inversión de capital humano y de recursos físicos por parte de las instituciones.

- ASP.NET como herramienta tecnológica para el desarrollo de aplicaciones web, brinda un gran número de funcionalidades propias relacionadas con formularios, registros, seguridad, entre otras. Esto permite al programador que los esfuerzos estén enfocados en dar solución a problemas puntuales de cada proyecto y evite esfuerzos considerables al hacer uso de funciones preestablecidas por la herramienta.
- La sistematización del proceso de reserva de un laboratorio y el hecho de poder llevar a cabo prácticas sobre el mismo de forma remota, facilitan el acceso a estos recursos al tiempo que se amplía la disponibilidad del mismo para su uso.
- Este proyecto plantea una serie de pautas necesarias para la implementación de un LRR sobre las plantas disponibles en LCP que cuenten con recursos hardware y software similares a los utilizados durante el desarrollo e implementación del SEDLAB. De forma general estas pautas describen un proceso de análisis de recursos disponibles y necesarios, selección de las herramientas software más adecuadas que permitan garantizar la integridad física de la planta, la comunicación entre una página web y el recurso físico y por último el desarrollo de una aplicación que realice la gestión de usuarios para la realización de una práctica de forma remota.
- La utilización de un LR implica cambios sustanciales en las metodologías empleadas en la realización de los trabajos de laboratorio y en la forma en que el alumno se apropia de los conocimientos. En consecuencia, es necesario valorar su impacto pedagógico y seguir las pautas y orientaciones de cómo utilizar esta herramienta de aprendizaje, para lo cual este trabajo aporta manuales de uso de administrador y estudiantes, en los cuales están consignadas las pautas necesarias para usar el SEDLAB desarrollado o para diseñar, construir y ejecutar un LRR sobre otras plantas.
- El PIAI como un programa de naturaleza innovadora debe incentivar entre sus estudiantes, la investigación y desarrollo de herramientas tecnológicas relacionadas con la TIC que aporten al proceso de formación profesional. Para tal fin es fundamental contar con un equipo de trabajo multidisciplinar, que permita la diversidad de conocimientos y experiencias, para hacer más detallado el desarrollo de este tipo de proyectos.

6.2. Trabajos futuros

- Implementación de un sistema de seguridad para el SEDLAB o cualquier otro proyecto relacionado con LRR en el LCP teniendo en cuenta el estándar ISA 99.
- Mejorar el entorno web con el fin de lograr que el estudiante disfrute de una mayor sensación de presencia física y participación en el laboratorio; para ello se debe investigar mucho mejor lo relativo a las técnicas de realidad aumentada o técnicas de realidad virtual.
- Virtualización de las plantas del LCP para la realización de las prácticas de laboratorio por medio de instrumentos virtuales.
- Implementación de LRR sobre los recursos disponibles en el LCP de la Universidad del Cauca, con el fin de crear convenios inter-institucionales que permitan a sus estudiantes compartir recursos de laboratorio para la realización de prácticas remotas.
- Desarrollo e implementación de un LRR que permita la programación online sobre los PLC Micrologix 1500.

6.3. Limitaciones

- El SEDLAB como gran parte de los proyectos relacionados con laboratorios remotos tiene la característica de ser accedidos de forma monousuario, es decir, solo un usuario a la vez puede realizar la práctica de laboratorio, lo que hace necesario implementar tecnologías y arquitecturas que permitan tener un acceso multiusuario.
- Si es necesario conectar varias plantas o dispositivos de control al mismo servidor, el problema radica en la necesidad de muchos puertos serie en el mismo.
- El SEDLAB solo cuenta con dos perfiles de usuario, sin embargo la creación de otro tipo de perfil debe realizarse cuidadosamente teniendo en cuenta que cada usuario debe contar con privilegios diferentes.
- El problema de la seguridad cibernética debe ser abordado en toda su complejidad.
- Falta realizar un estudio que permita el seguimiento de la calidad de esta implementación a través de encuestas a los usuarios.

- La aplicación web para la gestión de usuarios y reserva de horarios, solo funciona sobre el servidor IIS de Microsoft.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Manuel E. Macias. (2008) Laboratorios remotos de automatización y control. [Online]. http://www.cudi.edu.mx/primavera_2008/presentaciones/laboratorios_manuel_macias.pdf
- [2] Eduardo Caicedo Bravo, Eval Bladimir Bacca, Bayron Andrés Calvache, Jaiber Evelio Cardona, and Jaime Añilberto Buitrago, "Laboratorio Distribuido con Acceso Remoto para la Enseñanza de la Robótica," *Revista Educación En Ingeniería*, pp. 51-61, July 2009.
- [3] Juan Carlos Quiroga and David Linares Urrego. (2011) Desarrollo de un laboratorio remoto para el tunel del viento en el laboratorio de mecánica de la Universidad Distrital Francisco Jose de Caldas Facultad Tecnológica vía internet. [Online]. <http://tuneldelvientoud.wikispaces.com/file/view/TRABAJO+FINAL.pdf>
- [4] Bayron Calvache, Jaime Buitrago, Jaiber Cardona, Bladimir Bacca, and Eduardo Caicedo B. Laboratorio distribuido con acceso remoto a través de RENATA para la experimentación en robótica. [Online]. <http://www.renata.edu.co/index.php/ciencias-basicas/1282-laboratorio-distribuido-con-acceso-remoto-a-traves-de-renata-para-la-experimentacion-en-robotica.html>
- [5] Francisco A Candelas Herrias and José Sánchez Moreno, "Recursos didácticos basados en internet para el apoyo a la enseñanza de materias del área de ingeniería de sistemas y automática," *Revista iberoamericana de automática e informática industrial*, vol. 2, no. 2, pp. 93-101, 2005.
- [6] Edgar Rene Valverde Sanmiguel and Juan Pablo Rivera Palta, *DISEÑO E IMPLANTACION DE UN LABORATORIO REMOTO: CASO DE ESTUDIO PLANTA DE PÉRDIDAS DE CARGA EN CONDUCTOS A PRESION DEL LABORATORIO DE HIDRÁULICA.*, 2010.
- [7] A Rojko, D Hercog, and K Jezernik, "E-trainign in mechatronics using innovative remote laboratory," *ScienceDirect*, vol. 82, pp. 508-516, 2011.
- [8] Luis Payá, Oscar Reinoso, Arturo Gil, and M Jimenez, "Plataforma distribuida para la realización de prácticas de robótica móvil a través de internet," *Información Tecnológica*, vol. 18, no. 6, pp. 27-38, 2007.
- [9] H Vargas et al. (2009) Docencia en automática: Aplicación de las TIC a la realización de actividades prácticas a través de internet. [Online]. <http://rua.ua.es/dspace/bitstream/10045/17259/1/RIAI.pdf>
- [10] J. M Andújar Marquez and T. J Mateo Sanguino, "Diseño de laboratorios virtuales y/o remotos. Un caso práctico," *Revista iberoamericana de automática e informática industrial*, vol. 7, no. 1, pp. 64-72, 2010. [Online].

http://www.ehu.es/ikastorratza/3_alea/laboratorios.pdf

- [11] Nourdine Aliane, "Experiencia de uso de un laboratorio remoto de control," *Revista iberoamericana de automática e informática industrial*, vol. 7, no. 1, pp. 85-90, 2010.
- [12] Manuel Domínguez Gonzáles et al., "Estrategias docentes colaborativas basadas en la utilización de laboratorios remotos via Internet," *Cuadernos de Innovación Educativa en las Enseñanzas Técnicas Universitarias*, vol. 1, no. 1, pp. 75-81, 2007.
- [13] John Anderson Sandoval Moreno, José Miguel Ramírez Scarpetta, and Ana Lilian Valencia de Oro, "Desarrollo tecnológico de los laboratorios remotos de estructuras e ingeniería sísmica y dinámica estructural," *Ciencia e ingeniería neogranadina*, vol. 18, pp. 77-99, 2008.
- [14] Carlos Fernando Ariza Iadino and Darío Amaya Hurtado, "Laboratorio remoto aplicado a la educación a distancia," *Ciencia e ingeniería neogranadina*, vol. 18, no. 2, 2008.
- [15] J García Zubía, D López de Ipiña, G Alvez, and P Orduña Fernández. (2008) Análisis de requisitos software para un weblab. [Online]. <http://taee2008.unizar.es/papers/p30.pdf>
- [16] Luis M. Jiménez, Rafael Puerto, Óscar Reinoso, César Fernández, and Ramon Neco, "Recolab: Laboratorio remoto de control utilizando matlab y simulink," *Revista Iberoamerica de Automática e Informática Industrial*, vol. 2, no. 2, pp. 64-72, 2005.
- [17] Cristhian David Buchely and Fausto Ruiz Coque, "Herramienta basada en Redes de Petri para diseño de supervisores de sistemas de eventos discretos," 2012.
- [18] Juan Fernando Flórez M. Planta de sistemas a eventos discretos - Reconocimiento de la planta. [Online]. https://www.google.com.co/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDcQFjAB&url=http%3A%2F%2Fpis.unicauca.edu.co%2Fmoodle-2.1.2%2Fpluginfile.php%2F30121%2Fmod_folder%2Fcontent%2F1%2FP1_SED.pdf%3Fforcedownload%3D1&ei=GhJ4Ue3cK7TW0gHqyoFQ&usq=A
- [19] Laboratorios remotos y virtuales en.
- [20] Pablo Alejandro Pantoja Otero and Laura Marcela Segura Rosas, "Esquema de seguridad para la planta de control de presión del laboratorio de control de procesos basado en el estándar ISA - 99," , 2013.
- [21] Iván José Turmero Astros. (2012, Agosto) Detección y análisis de fallas. Análisis de modos y efectos de fallas. [Online]. <http://www.monografias.com/trabajos94/deteccion-modos-efectos-y-analisis-fallas/deteccion-modos-efectos-y-analisis-fallas.shtml>

- [22] Análisis modal de fallos y efectos (A.M.F.E). [Online]. <http://www.fundibeq.org/opencms/export/sites/default/PWF/downloads/gallery/methodology/tools/amfe.pdf>
- [23] Johana Stella Castellanos Arias and Leonardo Enrique Solaque Guzmán, "Modelado con redes de PETRI e implementación con GRAFCET de un sistema de manufactura flexible con procesos concurrentes y recursos compartidos," *Ciencia e ingeniería neogranadina*, vol. 20, no. 1, pp. 61-75, 2010.
- [24] Jose Refugio Martínez Reyes. (2003, Octubre) Diseño de un gestor de redes de PETRI.
- [25] Escuela de Ingenierías Industriales. Redes de Petri: Modelado, análisis y evaluación de prestaciones. [Online]. http://www.eis.uva.es/~jossan/doct/sed/fich/RdP_x6.pdf
- [26] Departamento de Ingeniería de Sistemas y Automática. Modelado con redes de Petri. [Online]. http://www.eueti.uvigo.es/files/material_docente/1705/tema6modeladoc_onredesdepetri.pdf
- [27] Yenny Cristina Zea Yanza and Cristian Alfonso Campo, "Sistema de supervisión para el proceso de medición dinámica de hidrocarburos," 2012.
- [28] Tech On My Web. [Online]. <http://www.slideshare.net/techonmyweb/grafcet-13496276>
- [29] José Manuel Gea. (2006, Mar.) Intriducción al Grafcet. [Online]. <http://www.automatas.org/redes/grafcet.htm>
- [30] Amparo Diaz Muñoz, "Arquitectura para la integracion de plataformas de control basadas en PC mediante OPC," 2011.
- [31] Whizzywig. Web based rich text editor for free. [Online]. <http://unverse.net/Whizzywig-web-based-rich-text-editor>
- [32] Comunidad Wikipedia. [Online]. es.wikipedia.org/wiki/LabVIEW
- [33] National Instruments. [Online]. <http://www.ni.com/labview/esa/>
- [34] Andres Felipe Caicedo Moreno. RWLab/Remote WaveLab. [Online]. <http://upcommons.upc.edu/pfc/bitstream/2099.1/11653/1/65479.pdf>
- [35] (2011, Julio) Rockwell Automation. [Online]. <http://www.rockwellautomation.com/>
- [36] Francisco Franco, "Caracterización del flujo en la Planta SED,".
- [37] Fernando Pascual and Moisés Pérez. OPC- LabView. [Online]. <http://www.etitudela.com/entrenadorcomunicaciones/downloads/12opclabview.pdf>
- [38] Instrumentación virtual con LabView. [Online]. <http://www.etitudela.com/entrenadorcomunicaciones/downloads/labviewcursodeinstrumentacion.pdf>

- [39] NI-IMAQ para cámaras USB. [Online]. <http://zone.ni.com/devzone/cda/epd/p/id/5030>
- [40] ImaqUSB. [Online]. <http://www.mediafire.com/?y9jh5p31w9z6942>
- [41] UserGuide NI-IMAQ for USB Cameras. [Online]. <http://www.ni.com/pdf/manuals/371492a.pdf>
- [42] (2011, Octubre) Adquisición de imágenes desde una webcam usando Labview. [Online]. http://www.youtube.com/watch?v=NGFW_7DNJT0