



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA EMBEBIDO
PARA EL CONTROL DE UN MÓDULO DE DISTRIBUCIÓN DE
MATERIALES. CASO PRÁCTICO ESPOCH**

Tesis de Grado previa a la obtención del Título de:

INGENIERO EN SISTEMAS INFORMÁTICOS

AUTOR: EDGAR VINICIO AVALOS YUQUE

TUTOR: ING. RAÚL ROSERO MIRANDA

Riobamba – Ecuador

2015

©2015, Edgar Vinicio Avalos Yuque

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMATICA Y ELECTRONICA
ESCUELA DE INGENIERIA EN SISTEMAS

El Tribunal de Tesis certifica que: El trabajo de investigación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA EMBEBIDO PARA EL CONTROL DE UN MÓDULO DE DISTRIBUCIÓN DE MATERIALES. CASO PRÁCTICO ESPOCH, de responsabilidad del señor: Edgar Vinicio Avalos Yuque, ha sido minuciosamente revisado por los Miembros del Tribunal de Tesis, quedando autorizada su presentación.

Ing. Gonzalo Samaniego Erazo

DECANO

Dr. Julio Santillán Castillo

DIRECTOR DE ESCUELA

Ing. Raúl Rosero Miranda

DIRECTOR DE TESIS

Ing. Marco Viteri Barrera

MIEMBRO DEL TRIBUNAL

DOCUMENTALISTA

SISBIB-ESPOCH

Yo, Edgar Vinicio Avalos Yuque, soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis y el patrimonio intelectual de la Tesis de Grado pertenece a **LA ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO.**

Edgar Vinicio Avalos Yuque

DEDICATORIA

A mi padre Jorge Avalos y mi madre Laura Yuque, a Luz María Zambrano mi abuela gracias por su incondicional apoyo, a mis hermanos, a mis profesores; a mis familiares y amigos.

Edgar

AGRADECIMIENTO

Agradezco a la Escuela Superior Politécnica de Chimborazo. A Dios, a mis padres, familiares, a mis profesores y amigos.

Edgar

CONTENIDO

RESUMEN	xiv
SUMMARY	xv
INTRODUCCIÓN	1
CAPÍTULO I.....	7
1. MARCO TEÓRICO REFERENCIAL.....	7
1.1 Generalidades de los Sistemas Embebidos	7
1.1.1 <i>Introducción a los Sistemas Embebidos</i>	7
1.1.2 <i>Estructura de un Sistema Embebido</i>	10
1.1.3 <i>Elementos de un Sistema Embebido</i>	10
1.1.4 <i>Arquitectura básica empleada</i>	12
1.2 Gnu/Linux.....	13
1.2.1 <i>Scientific Linux</i>	14
1.3 Herramienta de diseño de Sistemas Embebidos de control Software ...	15
1.3.1 <i>Labview</i>	16
CAPÍTULO II	19
2. MARCO METODOLÓGICO	19
2.1 Desarrollo del módulo de distribución de materiales.....	19
2.1.1 <i>Instalación Sistema Operativo Scientific Linux</i>	19
2.1.2. <i>Configuración Oracle VM VirtualBox con Scientific Linux 6.6</i>	21
2.1.3. <i>Instalación Labview sobre Scientific Linux</i>	40
2.1.4. <i>Labview sobre Scientific Linux</i>	47
2.2. Construcción del módulo de distribución de materiales	48
2.2.1 <i>Drivers de Comunicación.</i>	57
2.2.2 <i>Sistema embebido.</i>	62
CAPÍTULO III.....	68
3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS	68
3.1 Resultados y Analisis	68
3.1.1 <i>Definición de parámetros</i>	68
3.1.2 <i>Resultados individuales de los parámetros.</i>	69
Hipótesis.....	72
Determinación de la Hipótesis	72

CONCLUSIONES	75
RECOMENDACIONES	76
BIBLIOGRAFÍA	
ANEXOS	

INDICE DE ABREVIATURAS

MIT	Instituto Tecnológico de Massachusetts
INS	Inertial Navigation System
RAM	Random Access Memory
ROM	Read Only Memory
KB	Kilo Byte
GUI	Interfaz Gráfica de Usuario
RTOS	Real Time Operating System
BIOS	Basic Input & Output System
DMA	Direct Memory Access
GNU	GNU no es Unix
PDF	Portable Document Format
ISO	Organización Internacional de Estándares
PLC	Programmable Logic Controller.
I/O	Entrada/Salida de datos.
IP	Dirección del dispositivo en una red pública o privada..
RHEL	Red Hat Enterprise Linux.
VI	Virtual Instruments.

ÍNDICE DE TABLAS

Tabla 1-2.	Requerimientos Sistema Operativo	19
Tabla 2-3.	Definición de Variables	68
Tabla 3-3.	Índice de los parámetros	68
Tabla 4-3.	Definición de valorización	69
Tabla 5-3.	Conglomerado de datos	69
Tabla 6-3.	Estadístico del Indicador Distribución de Materiales.....	70
Tabla 7-3.	Conglomerado de datos	70
Tabla 8-3.	Estadístico del Indicador Velocidad Piezas.....	71
Tabla 9-3.	Conglomerado de datos Color Piezas.....	71
Tabla 10-3.	Estadístico del Indicador Color Piezas	72
Tabla 11-3.	Conglomerado de datos total	73
Tabla 12-3.	Resultados Globales	73

ÍNDICE DE FIGURAS

Figura 1-1.	Control de tráfico Aéreo	8
Figura 2-1.	Misil Minuteman II 1962	9
Figura 3-1.	Componentes de un Sistema Embebido.....	13
Figura 4-1.	Gnu/Linux	14
Figura 5-1.	Scientific Linux 6.6.....	15
Figura 6-1.	Herramienta de diseño Sistema Embebido	16
Figura 7-1.	Labview Panel Frontal.	17
Figura 8-1.	Ejemplo de un Diagrama de Bloques y Panel Frontal	18
Figura 1-2.	Descarga Sistema Operativo Scientific Linux	20
Figura 2-2.	Versión Sistema Operativo Scientific Linux	20
Figura 3-2.	Oracle VM VirtualBox	21
Figura 4-2.	Instalación Oracle VM VirtualBox.....	22
Figura 5-2.	Instalación Oracle VM Virtual box.....	22
Figura 6-2.	Icono Oracle VM VirtualBox	23
Figura 7-2.	Inicio Oracle VM Virtual Box	23
Figura 8-2.	Nueva Máquina Virtual.....	24
Figura 9-2.	Parámetro nuevo Sistema Operativo.....	24
Figura 10-2.	Configuración parámetros Máquina Virtual	25
Figura 11-2.	Asignación de Memoria RAM.....	25
Figura 12-2.	Asignación Final memoria RAM.....	26
Figura 13-2.	Creación Disco Duro Virtual	26
Figura 14-2.	Tipo de Archivo Disco Duro Virtual	27
Figura 15-2.	Asignación Espacio Disco Duro Virtual.....	27
Figura 16-2.	Ubicación y Tamaño del Sistema Operativo	28
Figura 17-2.	Instalación Sistema Operativo	29
Figura 18-2.	Ubicación imagen ISO archivo descargado	29
Figura 19-2.	Instalación Inicio Scientific Linux 6.....	30
Figura 20-2.	Opciones de instalación Scientific Linux	31
Figura 21-2.	Instalación Scientific Linux archivos primarios	31
Figura 22-2.	Instalación Scientific Linux carga procesos	32
Figura 23-2.	Instalación Scientific Linux presentación.....	32

Figura 24-2.	Instalación Scientific Linux selección idioma	33
Figura 25-2.	Instalación Scientific Linux idioma teclado	33
Figura 26-2.	Instalación Scientific Linux tipo de dispositivos	34
Figura 27-2.	Instalación Scientific Linux formato disco duro virtual	34
Figura 28-2.	Instalación Scientific Linux selección ubicación.....	35
Figura 29-2.	Instalación Scientific Linux contraseña root.....	35
Figura 30-2.	Tipo de instalación Scientific Linux	36
Figura 31-2.	Instalación Scientific Linux inicialización.....	36
Figura 32-2.	Instalación Scientific Linux finalización	37
Figura 33-2..	Instalación Scientific Linux instalación completa	37
Figura 34-2.	Instalación Scientific Linux primer inicio	38
Figura 35-2.	Inicio sistema operativo Scientific Linux	38
Figura 36-2.	Ingreso súper usuario root.....	39
Figura 37-2.	Contraseña súper usuario	39
Figura 38-2.	Pantalla principal Sistema Operativo Scientific Linux	40
Figura 39-2.	Comandos Terminal root	41
Figura 40-2.	Comando Ubicación en Directorio	41
Figura 41-2.	Comandos para la instalación de Labview.....	42
Figura 42-2.	Archivos descomprimidos Labview	43
Figura 43-2.	Archivos necesarios para la instalación	43
Figura 44-2.	Actualización archivos necesarios	44
Figura 45-2.	Actualización archivos necesarios	44
Figura 46-2.	Instalación Labview comandos.....	45
Figura 47-2.	.Archivos de Instalación Labview.....	45
Figura 48-2.	Archivos necesarios Instalación.....	46
Figura 49-2.	Archivos de Instalación.....	46
Figura 50-2.	Inicio Labview	47
Figura 51-2.	Software Labview inicio	47
Figura 52-2.	Montaje trole, tablero, motor y banda transportadora.....	48
Figura 53-2.	Módulo de distribución contenedores o magazines	49
Figura 54-2.	Módulo de Distribución Centronix bus de datos	50
Figura 55-2.	Modulo de Distribución Válvulas YONGYI	50
Figura 56-2.	Módulo de Distribución Cilindros Neumáticos AIRTAC	51
Figura 57-2.	Módulo de Distribución Motor de 24 Voltios VOSH.....	52

Figura 58.2.	Banda transportadora montaje.....	52
Figura 59.2.	Válvulas YONGYI conexión mangueras	53
Figura 60-2.	Cableado Centronix bus de datos.....	53
Figura 61.2.	Módulo de distribución pulsadores	54
Figura 62.2.	Estación de distribución cableada	55
Figura 63-2.	Cableado PLC	55
Figura 64-2.	Cableado tablero PLC	56
Figura 65-2.	Módulo de Distribución de Materiales completo.....	56
Figura 66-2.	Instalación NI Labview 2013.....	57
Figura 67-2.	Instalación NI Labview Ingreso usuario y organización	58
Figura 68-2.	Instalación Labview Seriales	58
Figura 69-2.	Instalación Ni Labview directorio de destino	59
Figura 70-2.	Instalación NI Labview Actualizaciones	59
Figura 71-2.	Instalación Ni Labview Licencias.....	60
Figura 72-2.	Instalación Ni Labview archivos	60
Figura 73-2.	Instalación Ni Labview pantalla	61
Figura 74-2.	Instalación Labview Final.....	61
Figura 75-2.	Labview Nuevo Proyecto.....	62
Figura 76-2.	Labview Diagrama de Bloques.....	62
Figura 77-2.	Sistema Embebido Panel Frontal.....	63
Figura 78.2.	Diagrama de Bloques Sistema Embebido	64
Figura 79.2.	NI OPC	65
Figura 80-2.	NI OPC Runtime.....	66
Figura 81-2.	OPC Quick Client	66
Figura 82-2.	Módulo de Distribución 1 contenedor	67
Figura 83-2.	Módulo de Distribución 3 contenedores	67

RESUMEN

La presente investigación tiene como propósito el diseño e implementación de un sistema embebido para el control de un módulo de distribución de materiales que funcionará en el Laboratorio de Automatización de la Facultad de Informática y Electrónica de la Escuela Superior Politécnica del Chimborazo. Se utilizó el Método Analítico además de los siguientes Materiales el sistema operativo GNU/Linux utilizando la distribución Scientific Linux 6.6. Implementando el software para diseño de sistemas embebidos Labview todo esto emulado sobre Oracle VM Virtual Box 4.3.12, 1 Laptop, 1 Motor de 24 Voltios, sensores, magazines, cables, piezas metálicas, banda transportadora, mangueras, tornillos, tuercas, PLC, cilindros neumáticos. Las técnicas que se aplicaron en la investigación fueron, test del sistema embebido, observación, pruebas prácticas en el módulo de distribución, análisis en tiempos de respuesta. Obteniendo resultados del 90% en la distribución de materiales de acuerdo a los test realizados. La distribución de materiales se la realiza a través de 3 magazines o contenedores los cuales tienen en su interior 4 piezas del mismo color por contenedor, estas piezas emulan materiales, tenemos piezas de color rojo, negro y plateadas las mismas que según nuestras necesidades serán empujadas por cilindros neumáticos hacia una banda transportadora de 1 metro de largo que está accionada por un motor cilíndrico de 24 Voltios a una velocidad de 1/5 m/s; todo el proceso será controlado por la aplicación Labview. El módulo de distribución original obtuvo el 30% en la distribución de materiales, debido a que está construido con un solo contenedor que contiene 12 piezas de colores en orden aleatorio. Se concluye que la adecuación de más contenedores en los módulos de distribución de materiales aumenta la productividad en un 30% además de facilitar el acceso a más piezas de colores, la estación original de un magazine tiene problemas de acceso si deseamos una pieza de un color diferente que se encuentra en la última posición. Se recomienda al director del Laboratorio de Automatización de la Facultad de Informática y Electrónica implementar un magazine o contenedor circular con el cual las piezas que no cumplan con los requerimientos puedan ser desechadas de la línea de montaje.

PALABRAS CLAVE: <PROGRAMMABLE LOGIC CONTROLLER [PLC]>, <SISTEMA OPERATIVO [GNU/LINUX]>, <CILINDROS NEUMATICOS [AIRTAC]>, <SISTEMA EMBEBIDO>, <LABORATORIO AUTOMATIZACIÓN>, <SOFTWARE [ORACLE VM VIRTUALBOX]>, <SOFTWARE [LABVIEW]>, <MODULO DE DISTRIBUCION>.

SUMMARY

The purpose of the following research is to design and implement an embedded system to control a material distribution module which will work at the Automation Laboratory in the Electronics and Computing Faculty of Escuela Superior Politécnica de Chimborazo. For the research, it was necessary to use the Analytical method as well as the following materials: GNU/Linux OS, by using Linux 6.6 scientific distribution for implementing the Labview 2013 software for embedded system design emulated with Oracle VM virtual box 4.3.12, a laptop, a 24-volt motor, sensors, containers, cables, metal pieces, conveying belt, hoses, screws, nuts, PLC, and pneumatic cylinders. The techniques used for the research were: embedded system tests, observation, practical tests in the distribution module, and response time analysis obtaining a 90% of performance in the distribution of materials according to the tests carried out. The distribution of materials is done by using three containers which have four internal pieces of the same color per container to emulate materials, there are red, black and silver pieces which will be pulled by pneumatic cylinders to a one-meter conveying belt powered by a 24-volt cylindrical motor at a speed of $1/5$ m/S, all the process will be controlled by Labview application. The original distribution module got a 30% in the material distribution, since it is made up for one containers in the materials distribution modules increases the production in a 30% and also eases the access problems if a color piece which is in the last position in required. It is recommended for the Director of Automation Laboratory to implement a circular container which rejects the piece which do not fulfill the requirements at the assembly line.

INTRODUCCIÓN

Nuestro país considerado en vías de desarrollo y con el fin de que alcance el progreso completo para el bienestar de sus ciudadanos, es de vital importancia, implementar sistemas embebidos en la industria en general para facilitar procesos de producción. Estos sistemas en la industria ayudarán al crecimiento económico de diferentes sectores del país, ya que se lograría la ayuda en la fabricación de nuestros productos de manera directa y sin necesidad de la intervención de agentes externos.

Los sistemas embebidos podemos encontrarlos en un sinnúmero de dispositivos a nuestro alcance, desde el control remoto de televisores, lavadoras, microondas; inclusive los primeros celulares funcionaban con sistemas embebidos. Estos sistemas a pesar de no ser muy conocidos en la actualidad, casi es difícil encontrar a un dispositivo que no utilice estos sistemas para su funcionamiento.

Un Sistema Embebido utiliza un microprocesador que incluye interfaces de entrada y salida, en la mayoría de estos sistemas incluye una interfaz de monitoreo para verificar su estado y hacer un diagnóstico del mismo.

Este proyecto está enfocado en diseñar e implementar un sistema embebido en un módulo de distribución de materiales ubicado en el Laboratorio de Automatización de la FIE de la Escuela Superior Politécnica del Chimborazo, y servirá como referente a los estudiantes para futuros proyectos similares. Todo esto para hacer realidad el buen vivir de los ecuatorianos.

El Capítulo I Marco Teórico Referencial, comprende los antecedentes, la justificación, los objetivos y se plantea la Hipótesis.

El Capítulo II Marco Metodológico, en este capítulo se describe los conceptos de sistema embebido su estructura sus componentes, el sistema operativo a utilizarse como la herramienta de diseño de sistemas embebidos Labview.

El Capítulo III Marco de Resultados, Discusión y análisis de los resultados el cual comprende el desarrollo del Módulo de distribución, detallara los materiales a utilizar además de la construcción del módulo de distribución de materiales, se describe también los resultados y análisis de la investigación científica, también se agrega las Conclusiones y Recomendaciones donde se detallan los aspectos más importantes en la investigación científica como también recomendaciones.

Antecedentes

Los estudiantes de la Facultad de Informática y Electrónica de la Escuela Superior Politécnica de Chimborazo, quienes, deseosos por estar al corriente de nuevas tecnologías en lo que se refiere a la automatización industrial: la mecatrónica y demás materias afines, siempre en constante investigación; decidimos realizar un proyecto el cual consiste en la adaptación e implementación de un “Módulo de distribución de materiales”.

La adaptación abaratará costos y se acoplará a la realidad nacional, colaborando de esta manera al buen vivir de los ecuatorianos. La estación se encuentra conformada por varios módulos siendo el primero a ser implementado.

El primer módulo será desarrollado por el autor de esta investigación científica. El desarrollo debe ser de manera conjunta, con los demás módulos restantes para su correcta integración, los que serán implementados por estudiantes de la escuela de electrónica.

La función que desempeñara el módulo 1 me permito describir a continuación. El primer módulo de la estación de distribución original se encuentra en la Figura 65-2.

La parte inicial de la Estación de distribución de materiales consta de tres contenedores o magazines los cuales podemos observar en la Figura 65-2. Estos contenedores se encuentran llenos con piezas las cuales emulan materiales, tenemos piezas de color rojo, negro y plateadas las mismas que según nuestras necesidades serán empujadas por cilindros neumáticos hacia una banda transportadora de 1 metro de largo que está accionada por un motor cilíndrico de 24 Voltios a una velocidad de $1/5$ m/s.

Dichas piezas pasaran al siguiente módulo. Todo el proceso será controlado por una aplicación. De Monitoreo de procesos se encuentra en la Figura 72-2

.El problema a tratar de solucionar se encuentra en el Laboratorio de Automatización de la FIE el cual consta de una estación de distribución de un solo contenedor o magazine, esta Estación de distribución de un solo contenedor ubicada en el Laboratorio de Automatización de la FIE se encuentra en la Figura 82-2.

Este Módulo cumple con la ley FIFO (primero en entrar es el primero en salir). Su único magazine contiene todas las fichas en una sola columna. Esto es un problema si se quisiera obtener la pieza negra que se encuentra en la última posición.

En el módulo existente en el Laboratorio de Automatización de la FIE se detectaron los siguientes problemas:

La tecnología existente exige que cada componente ingrese individualmente, causando dificultades en la producción y control, se aumenta el tiempo de ciclo y se disminuye la productividad.

En muchos casos la materia prima exige que el sistema de control sea más preciso y requiera menor tiempo. El sistema existente necesita un control extra para diferenciar el material.

Justificación

Justificación Teórica.

En este mundo globalizado lleno de valores materiales donde lo que realmente importa es el dinero, todo tiene su precio: el tiempo, el espacio, la mano de obra en especial.

Con estos antecedentes cualquier producto manufacturado tiene su costo, muchos de ellos escandalosamente altos, dificultando su comercialización por la enorme competencia que existe a todo nivel y desde cualquier parte del globo terráqueo, haciendo más difícil la economía familiar, social y de país.

Observamos actualmente con gran preocupación cómo se derrumban las economías del mundo, especialmente la del Ecuador, porque sus ingresos para solventar el presupuesto han apostado a un solo producto: el petróleo, al derrumbarse su precio; todo lo planificado se va al traste por la falta de recursos económicos.

El gobierno para estabilizar medianamente su economía recurre a otras medidas: el abusivo endeudamiento externo, nuevos aranceles, abolición de compromisos estatales en vigencia desde hace decenas de años, etc. donde la ciudadanía es perjudicada directamente.

La gran apuesta del régimen es el cambio de la matriz productiva y en ese sentido –creo yo-, es la única política que permitirá a nuestro pueblo enrumbarse en días de bienestar y progreso, porque la economía estatal estará encaminada a la diversificación tanto en su producción, los tipos de esta, su calidad, la rapidez y los bajos precios con que los productos se consigan.

Es tarea de cada uno de los ecuatorianos aportar positivamente en la búsqueda de alternativas que permitan a nuestra gente alcanzar niveles de desarrollo donde sus familiares, la sociedad en general tengan perspectivas de desarrollo para alcanzar el buen vivir.

Es responsabilidad mayor para las instituciones de educación superior de gran prestigio como la Escuela Politécnica de Chimborazo presentar estas alternativas. Pero aún más, a los estudiantes que hemos terminado una carrera de estudios, como pago a la sociedad, al estado, por habernos permitido forjarnos en una carrera que está destinada al servicio social.

En este marco referencial, pongo en consideración la siguiente investigación científica:

“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA EMBEBIDO PARA EL CONTROL DE UN MÓDULO DE DISTRIBUCIÓN DE MATERIALES. CASO PRÁCTICO ESPOCH”

Que está sustentado en el principio de Kanban. Es un sistema de información que controla de modo armónico la fabricación de los productos en cantidad y tiempo necesarios en cada uno de los procesos. Al aplicar este principio entre la estación de distribución actual de la FIE y la estación propuesta en esta investigación científica se abarataran costos con relación a la máquina existente en el mercado y se permite disminuir tiempos de distribución de materiales.

La implementación de esta técnica en la producción, será sin duda una herramienta básica en el cambio de la matriz productiva que tanto promueve el gobierno, incidiendo directamente en el bienestar común y en consecuencia el buen vivir.

Justificación Metodológica.

El realizar un análisis comparativo entre el módulo actual de la FIE y el módulo de distribución de materiales propuesta en esta investigación científica, permite verificar los resultados obtenidos entre estos dos módulos de distribución para definir cuál de estos será el más óptimo según la complejidad de procesos requeridos en el Laboratorio de Control de procesos FIE.

Justificación Práctica.

El construir dicha estación de distribución desde cero, fomentará en los estudiantes el estudio de técnicas para la realización de futuros proyectos similares. Ayudará a entender de mejor manera el mundo de la automatización industrial, ya que estará implementado en el Laboratorio de Automatización de la FIE.

Para realizar este proyecto se ha tomado en cuenta diferentes temas de tesis elaborados dentro y fuera de la ESPOCH.

Estudios Realizados dentro y fuera de la ESPOCH.

- Automatización de la línea de envasado de productos cosméticos de laboratorios WINDSOR S.A (ESPE).
- Automatización y monitoreo de un brazo robótico para la manipulación, transporte y clasificación de piezas en un Área de trabajo. (USFQ)
- Sistema de Control con PLC de un Módulo Industrial de envasado automático con fines didácticos, en el laboratorio OMRON de la FISEI/UTA. (UTA).

- Diseño y construcción de un PLC modular a Base de micros controladores. (ESPE-LATACUNGA).
- Diseño e Implementación de un sistema de SCADA para control del proceso de un Módulo didáctico de montaje FESTO Utilizando PLC y una pantalla HMI, caso práctico: en el laboratorio de automatización de la FIE (ESPOCH).
- Propuesta metodológica para la programación de PLC en GRAFSET para las competencias de mecatrónica world skill. caso práctico: laboratorio de mecatrónica (EIS). (ESPOCH).
- Repotenciación de un sistema de control semiautomático para la cortadora de perfiles utilizando PLC para empresa IMAC. (ESPOCH).

Todos los estudios realizados dentro y fuera de la ESPOCH referente al tema son considerados a la hora de la construcción del módulo de distribución de materiales.

Algunos lineamientos fueron considerados como:

LINEAS Y PROGRAMA DE LA ESPOCH

- LINEA: V. Tecnologías de la información, comunicación y procesos industriales.
- PROGRAMA: d. Programa para el desarrollo de automatización y control de procesos industriales.

Objetivos

Objetivo General

- Diseño e Implementación de un sistema embebido para el control de un módulo de distribución de materiales

Objetivos Específicos

- Establecer las características, dimensionamiento y componentes de la estación distribución.
- Implementar el módulo de distribución en el Laboratorio de Automatización de la FIE
- Aplicar los principios de Kanban en la programación del módulo.
- Diseñar e implementar un sistema de visualización
- Investigar e implementar un sistema de comunicación entre el sistema de visualización y el módulo de distribución de materiales.

Hipótesis

La implementación de un sistema embebido para el control de un módulo de distribución de materiales, para el sistema de producción modular, optimizará el proceso de manejo de materiales del sistema de montaje

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

En este capítulo se realiza un estudio sobre los sistemas embebidos, su definición la estructura, sus componentes además de una introducción al sistema operativo a utilizarle para la realización de esa investigación científica, como también la herramienta de diseño de sistemas embebidos de control software Labview.

1.1 Generalidades de los Sistemas Embebidos

1.1.1 Introducción a los Sistemas Embebidos

Se entiende por sistema embebido, empotrado, insertado, incrustado a una constitución de hardware y software de computador, también se pueden adicionar algunas piezas mecánicas o de otras funciones, su fin es realizar una tarea específica o única función. Estos dispositivos se encuentran en un sin número de lugares en nuestra vida diaria, como por ejemplo el control remoto de nuestros televisores, las lavadoras o en el microondas.

La mayoría de personas no están conscientes de que existe un circuito integrado y líneas de código ejecutándose que les permite trabajar.

Esto brinda una diferencia bien marcada con los ordenadores con los cuales trabajamos, estudiamos o utilizamos por diversión, que también están formados por una combinación de hardware y software más algunas piezas como por ejemplo el ratón, una cámara web, una impresora los discos duros o una memoria flash.

Sin embargo las computadoras no son diseñadas para realizar una sola tarea, estos artefactos están diseñados para darles varias ocupaciones al mismo tiempo como por ejemplo podemos escuchar música mientras imprimimos un archivo o editamos una hoja de texto.

Podemos encontrar sistemas embebidos en diferentes elementos de sistemas más complejos, como por ejemplo el sistema de navegación o el GPS en automóviles actuales son sistemas embebidos,

Esta mezcla de software y hardware en varios casos puede ser substituida por un circuito integrado que realice el mismo trabajo, este circuito podría ser creado, diseñado o implementado especificando las necesidades del sistema.

Una de las ventajas de los sistemas embebidos es su flexibilidad. Esto quiere decir que a la hora de realizar algún cambio podemos hacerlo solo modificando líneas de código al software del sistema embebido, esto sería más factible que construir un nuevo circuito integrado.

En la actualidad los sistemas embebidos se encuentran frecuentemente en los sistemas de tiempo real, concebimos el término sistemas de tiempo real a aquellos sistemas que dependen exclusivamente del tiempo en su funcionamiento.

Un ejemplo claro de sistemas embebidos de tiempo real son los que encontramos en las fábricas de procesamiento de productos, si en la línea de montaje fallara el dispositivo de empaquetamiento, toda la línea de montaje quedaría colapsada con productos acumulándose en la línea de montaje sin ser procesados, esto generaría grandes pérdidas económicas a la empresa.

El control de tráfico aéreo es otro ejemplo claro de sistemas embebidos de tiempo real este sistema es de vital importancia para evitar accidentes y tener un control total del tráfico aéreo en cada país estos sistemas tienen una interfaz de monitoreo en la cual se puede observar en tiempo real el desplazamiento de cada vuelo en una cantidad de tiempo.

Un sistema de tiempo real debe satisfacer restricciones explícitas en el tiempo de respuesta o arriesgarse a severas consecuencias, incluida la falla.



Figura 1-1. Control de tráfico Aéreo

Fuente: <https://www.fayerwayer.com/2008/09/trafico-aereo-mundial-en-24-horas/>

Uno de los primeros sistemas embebidos registrados en la era actual fue el que se encontraba en las Naves para misiones Apolo hacia Luna mejorado por el laboratorio de desarrollo del Instituto Tecnológico de Massachusetts **MIT**.

Cada vuelo realizado hacia la luna tenía en sus módulos de excursión lunar dos sistemas embebidos que servían como dispositivos guía de navegación inercial o **INS** estos sistemas determinaban la ubicación dirección y velocidad sin la intervención de agentes externos de los módulos de excursión.

En la parte inicial del proyecto fue considerado como el mecanismo que más riesgo presentaba en los viajes Apolo, este sistema de cómputo fue el originario en utilizar circuitos integrados y utilizaba una memoria **RAM** magnética, con un tamaño de 16 bits.

El software fue estructurado en lenguaje ensamblador y a su vez este era el sistema operativo básico quien controlaba los módulos de excursión lunar, capaz de soportar hasta ocho tareas simultáneas.

Uno de los primeros sistemas embebidos producidos en masa, fue el computador guía del misil norteamericano Minuteman II en 1962. El primordial aspecto de diseño del computador fue que además de estar montado con circuitos integrados, admitía reprogramar los algoritmos de guía del misil para la disminución de errores, y permitía realizar pruebas sobre el misil ahorrando así el peso de los cables y conectores además de ahorrar la construcción de un circuito integrado si se lo requería, ya que cada cambio se lo hacía mediante líneas de código.



Figura 2-1. Misil Minuteman II 1962

Fuente: <http://www.airliners.net/photo/USA---Air/Boeing-LGM-30-Minuteman/0267372/L/>

1.1.2 Estructura de un Sistema Embebido

Un Sistema Embebido está estructurado por 3 componentes principales.

- Hardware
- Software primario o aplicación principal. Este software o aplicación ejecuta una tarea particular, o en algunas ocasiones una serie de tareas
- Un sistema operativo que permite controlar la(s) aplicación(es), además de proveer los mecanismos para la ejecución de procesos. En muchos sistemas embebidos es necesario que el sistema operativo posea características en tiempo real.

Es significativo destacar que el software que se ejecuta en un sistema embebido es creado bajo algunas limitaciones importantes, ya sea en las cantidades pequeñas de memoria generalmente en el orden de los **KB**, capacidades limitadas de procesamiento, la necesidad de limitar el consumo de energía en cualquier instante bien sea en estado de ejecución o no.

Estas características hacen a los sistemas embebidos, herramientas tecnológicas útiles en el desarrollo de dispositivos que son de gran ayuda en la automatización industrial o en nuestra vida cotidiana.

Las principales diferencias de los sistemas embebidos con los computadores son las siguientes:

- Un valor más bajo al construirlos
- Pueden ir implementados en diferentes arquitecturas de procesadores
- Su desarrollo involucra un diseño específico de hardware y software con el fin de satisfacer la tarea final a cumplir
- Recursos limitados de hardware, por ejemplo a la memoria RAM, ROM y a la interacción de dispositivos de Entrada/Salida (E/S).
- El bajo consumo de energía
- Los Sistemas Embebidos poseen limitaciones de tiempo

1.1.3 Elementos de un Sistema Embebido

Las esenciales características de un sistema embebido para su fabricación son el bajo costo y consumo de potencia. Esta consideración se la toma por que la gran mayoría de estos sistemas son perfeccionados para ser producidos en cientos de unidades, el costo que conlleva la fabricación de un sistema embebido por unidad se la debe tener muy en cuenta al diseñar estos sistemas.

Generalmente estos sistemas recurren a procesadores que no demandan muchos recursos, además de memorias pequeñas para minimizar los costos en su fabricación.

En los sistemas embebidos la rapidez en su desenvolvimiento únicamente no depende de la velocidad del reloj del procesador, también interactúa la arquitectura y demás componentes si los tuviera.

Anteriormente se consideró, que un sistema embebido enfrenta comúnmente fuertes restricciones de recursos como por ejemplo poca memoria o número de tareas a realizar, para todas estas limitaciones es recomendable utilizar sistemas operativos de tiempo real o **RTOS** para su correcto funcionamiento.

Los sistemas embebidos se ejecutan disminuyendo tiempos en los que las tareas no se ejecutan o el sistema está sin tareas asignadas además desafían restricciones de hardware extremas como la falta de un dispositivo para almacenamiento de datos, usa una interfaz LCD en muchos casos para presentar datos.

Para resumir el concepto de sistema embebido podríamos denotar qué está principalmente estructurado por un software que se ejecuta sobre un procesador. Algo muy importante es que esta aplicación o sistema embebido necesita un espacio donde poder almacenarse luego de interactuar con el procesador. A este procedimiento de almacenamiento de memoria lo nombraremos memoria **ROM**.

Un sistema embebido debe estar estructurado con entradas y salidas fundamentales para comunicarse con el exterior, estas entradas o salidas pueden estar conectadas con sensores, motores, cilindros neumáticos o dispositivos que interactúen con el sistema.

Debido a que las instrucciones realizadas por sistemas embebidos no requieren de la utilización de todos los recursos de un procesador los más usados cuentan con registros de 8 o 16 bits.

En un sistema embebido la memoria del procesador debería utilizarse exclusivamente para almacenar las líneas de código de la aplicación.

Las principales características que deben tener los sistemas embebidos fueron ya mencionadas las mismas que varían de acuerdo a las necesidades o requerimientos de cada aplicación.

1.1.4 Arquitectura básica empleada

Un sistema embebido posee una arquitectura semejante a la de un computador personal. A continuación se detallan las partes más importantes.

MICROPROCESADOR

Realiza las operaciones fundamentales y primarias de cálculo del sistema. Principal responsable de la ejecución de las líneas de código existentes en una tarea determinada dirige además el trabajo o puesta en marcha de los demás dispositivos que actúan directamente sobre él como por ejemplo el control o puesta en marcha de un motor en un módulo de distribución.

MEMORIA

En la memoria encontramos incrustado las líneas de código necesarias para que el sistema se ejecute además de los datos utilizados por el sistema. Su principal función es que mediante velocidades muy altas tener libre entrada a la lectura y escritura del microprocesador esto le es de mucha ayuda al procesador para que este no despilfarre recursos y pierda el tiempo en tareas que no tienen que ver con tareas primarias de cálculo.

Al ser volátil la memoria esto indica que se pierde la información si se corta el flujo eléctrico una parte fundamental del sistema es que siempre necesitara un lugar donde almacenarse sin depender de la corriente eléctrica si este llegara a fallar o a faltar.

CACHÉ

A diferencia de la memoria principal esta es más rápida ya que acá se encuentran los datos accedidos recientemente y corren el riesgo de perderse si se corta o falla el flujo eléctrico.

DISCO DURO

En el disco duro la información no es volátil es decir no se pierde si se corta el flujo de energía y además puede conseguir capacidades o tamaños muy elevadas. A diferencia de las memorias que son de estado sólido éste suele albergar la información de forma magnética. Si utilizamos sistemas embebidos necesitaremos discos duros con almacenamiento y capacidades muy limitadas.

En la Figura 3-1. Tenemos un diagrama de bloques con los componentes fundamentales de un sistema embebido.

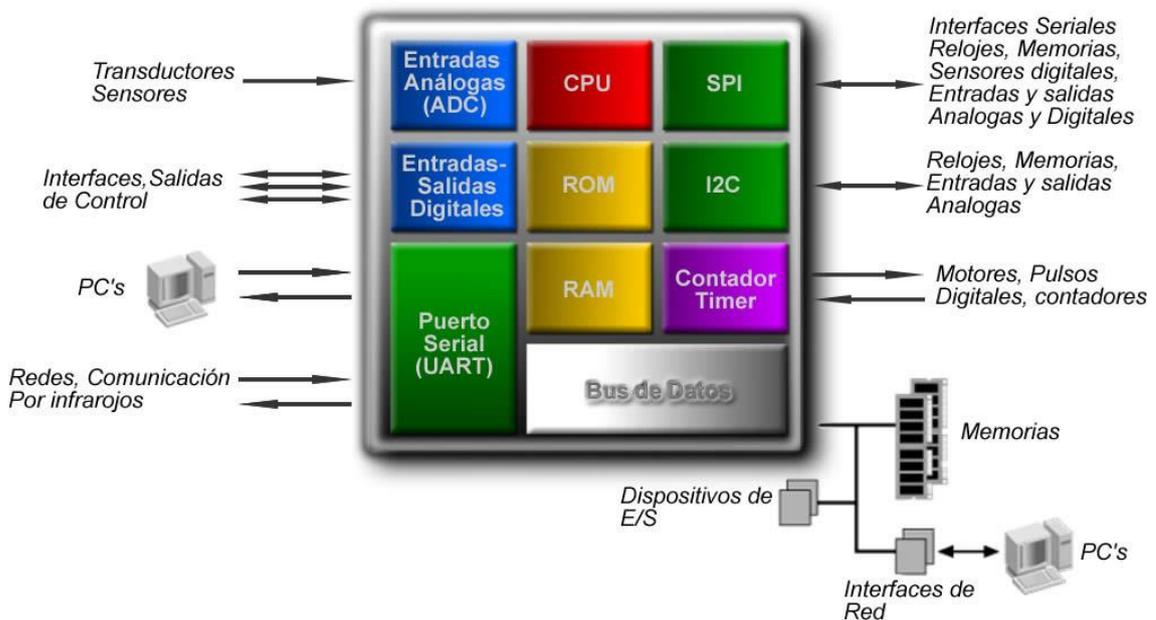


Figura 3-1. Componentes de un Sistema Embebido

Fuente: http://datateca.unad.edu.co/contenidos/208006/sistemas_embebidos_contenido/leccin_no_6_diagrama_de_bloques_de_un_sistema_embebido.html

1.2 Gnu/Linux

Se denomina a Linux a un sistema operativo que a su vez es un conglomerado de software, aplicaciones o programas que envían y reciben órdenes de un ordenador o computadora un sistema operativo además pone en marcha otros programas como editores de texto, editores de fotografías navegadores, entre otros.

El núcleo en un sistema operativo es la parte más importante pero a su vez no es la única parte que conforma un sistema operativo, además este tiene otra parte fundamental que son los programas o aplicaciones con las que viene por defecto como los que son escritos por el proyecto GNU o desarrollados para este.

Si solo tenemos el núcleo no podemos decir que es un sistema operativo funcional, para que sea un sistema funcional se necesita que se integren las demás aplicaciones es decir la manera correcta para definir a este sistema operativo sería GNU + Linux, la mayoría de usuarios cometen el error de llamarlo de manera coloquial solo Linux, la forma correcta sería GNU/LINUX.

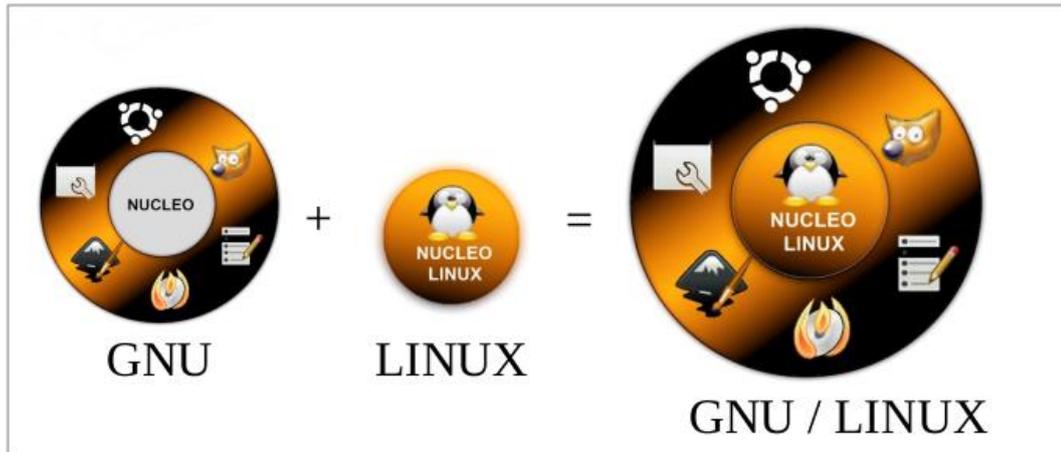


Figura 4-1. Gnu/Linux

Fuente: <https://xixirin.wordpress.com/gnulinux/>

Cabe recalcar algunas diferencias entre los sistemas operativos más importantes en la actualidad, por ejemplo GNU/LINUX no tiene dueño, como otros sistemas operativos privativos, GNU/LINUX es continuamente modificado y mejorado por programadores alrededor el mundo de manera voluntaria.

Todos los usuarios que utilizan GNU/LINUX tienen la ventaja de elegir software creado por distintas personas en todo el mundo, están disponibles cada día diferentes versiones y nuevos programas

Una ventaja de este sistema operativo es que no es tan vulnerable a los virus como otros sistemas, estos sistemas difícilmente colapsan y se pueden ejecutar varias tareas o ejecutar varios programas al mismo tiempo, además de la seguridad y estabilidad que ofrece.

Todas estas características hacen que este sistema operativo sea uno de los más importantes en la actualidad, son elegidos mayormente por usuarios domésticos y por grandes empresas reemplazan ya en la actualidad a los sistemas operativos convencionales

1.2.1 Scientific Linux

Scientific Linux es un sistema operativo que tiene relación directa con Red Hat Enterprise Linux recompilado, co-desarrollado por el Laboratorio Nacional Fermilab y la Organización Europea para la Investigación Nuclear (CERN en inglés). A pesar de que apunta a ser plenamente compatible con Red Hat Enterprise Linux, también provee paquetes adicionales no encontrados en la distribución de que proviene.

Scientific Linux por lo general está compuesto en su totalidad, por Software libre que es el software que respeta la libertad de los usuarios y la comunidad. A grandes rasgos, significa que

los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software. Es decir, el software libre es una cuestión de libertad, no de precio.

Para entender el concepto, debemos pensar en libre como en libre expresión. En inglés a veces traducimos free software como software gratis y esta erróneo porque en realidad el free debemos traducirlo como libertad, la libertad de elección, libertad de copiar el software, la libertad de estudiar el software, o de redistribuido libremente.

Para el desarrollo de esta investigación se va a utilizar la versión 6.6 de Scientific Linux, por compatibilidad con la herramienta de diseño de sistemas embebidos de nombre Labview.



Figura 5-1. Scientific Linux 6.6

Fuente: Scientific Linux 6.6.

Aunque a menudo una distribución incorpora aplicaciones o controladores propietarios. Scientific Linux utiliza además software libre como aplicaciones secundarias.

1.3 Herramienta de diseño de Sistemas Embebidos de control Software

Para el presente proyecto de investigación se empleará el Software Labview para el diseño del sistema embebido, porque posee una interfaz de monitoreo gráfica ideal para el control del módulo de distribución de materiales y todos los componentes que lo conforman.

1.3.1 Labview

Al diseñar sistemas embebidos para controlar módulos de distribución de materiales, es muy importante tener un entorno de desarrollo gráfico para controlar el proceso en distintas etapas. Labview nos ofrece un entorno de desarrollo que nos brinda todas estas opciones al diseñar e implementar sistemas embebidos, acelera la productividad ya que se pueden hacer pruebas en tiempo real además de integrar un lenguaje gráfico mediante el cual tenemos un control total de todo el proceso y de los componentes que actúan.



Figura 6-1. Herramienta de diseño Sistema Embebido

Fuente: <http://www.ni.com/labview/requirements/esa/>

Cuando se desarrolla programas utilizando el diseñador de sistemas embebidos Labview estos reciben el nombre de Instrumentos virtuales VLS, o Virtual Instruments por sus siglas en inglés, gracias a los bloques prediseñados que contiene este entorno de desarrollo los programas no se desarrollan mediante la digitación de código, si no que se dibujan.

Estos instrumentos virtuales o programas están compuestos de dos partes, el Panel Frontal y el Diagrama de Bloques.

El Panel Frontal es la interfaz directa con el usuario en la cual se pueden insertar controles tanto entradas como salidas, un ejemplo en entrada podríamos ingresar botones y en salidas o indicadores ingresamos una interfaz de monitoreo, todo esto podemos hacerlo cuando se está ejecutando el programa utilizando la barra de herramientas o la paleta de colores.

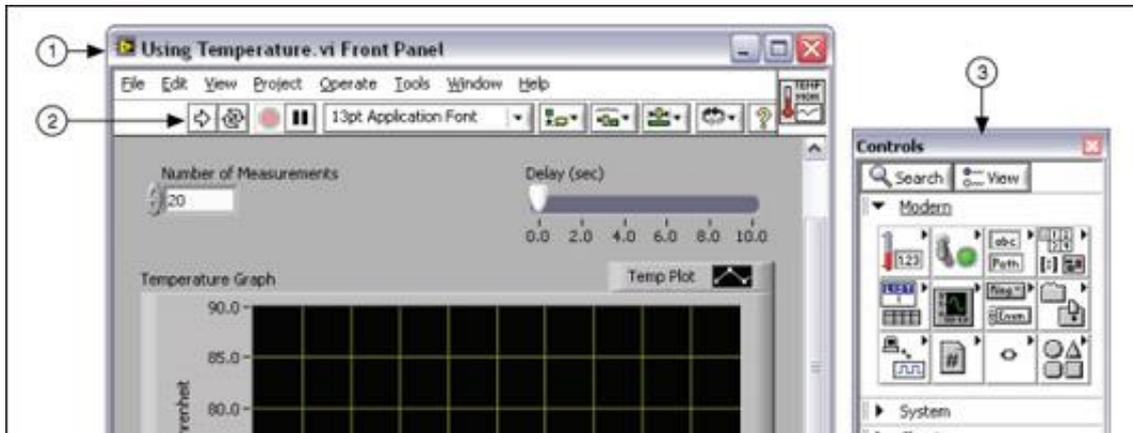


Figura 7-1. Labview Panel Frontal.

(1) Ventana de Panel Frontal | (2) Barra de Herramientas | (3) Paleta de Controles

Fuente: <http://www.ni.com/getting-started/labview-basics/esa/environment>

El Diagrama de Bloques se lo puede definir como el programa en sí o el lugar donde vamos a desarrollar todos los requerimientos que necesitamos para realizar las funciones que llevara a cabo el módulo de distribución de materiales, donde se define su funcionalidad es decir aquí se programaran las líneas de condigo del sistema embebido pero de manera gráfica.

En la Figura 8-1. Podemos observar un diagrama de bloques con el panel frontal y como se relacionan.

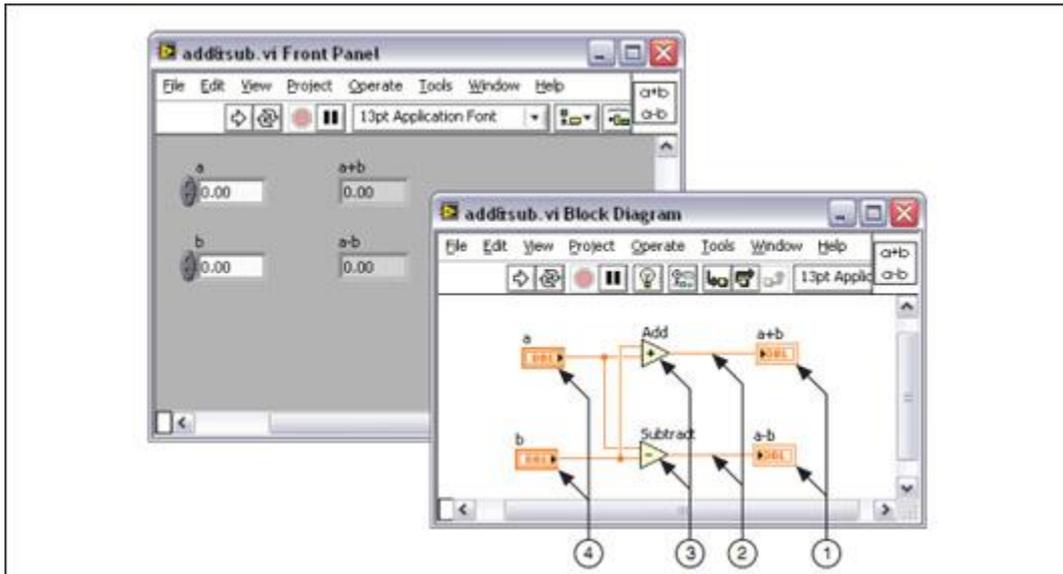


Figura 8-1. Ejemplo de un Diagrama de Bloques y Panel Frontal

(1) Terminales de Indicador | (2) Cables | (3) Nodos | (4) Terminales de Control

Fuente: <http://www.ni.com/getting-started/labview-basics/esa/environment>.

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1 Desarrollo del módulo de distribución de materiales

Para el correcto acoplamiento con el módulo de distribución de materiales como parte inicial en este capítulo, se procederá a la instalación del sistema operativo a utilizarse para la realización del proyecto.

2.1.1. Instalación Sistema Operativo Scientific Linux

Se procede a la instalación del sistema operativo Scientific Linux cuya versión seleccionada para este proyecto es la 6.6 en base a los requerimientos del software para el diseño de sistemas embebidos Labview, tal como indica en la tabla de requisitos.

Tabla 1-2. Requerimientos Sistema Operativo

Linux	Run-Time Engine	Entorno de Desarrollo
Procesador	Pentium III/Celeron 866 MHz o equivalente	Pentium 4/M o equivalente
RAM	256 MB	1 GB
Resolución de Pantalla	1024 x 768 píxeles	1024 x 768 píxeles
SO	Linux kernel 2.4x, 2.6x o 3.x y GNU C Library (glibc) Versión 2.5.1 para la arquitectura Intel x86_64 La <i>Guía de Instalación de LabVIEW</i> omite de un modo incorrecto Linux kernel 3.x de esta lista.	Red Hat Enterprise Linux Desktop + Workstation 6 o posterior, open SUSE 12.3 o 13.1 o Scientific Linux 6 o posterior.
Espacio en Disco	115 MB (32 bits) 131 MB (64 bits)	1.2 GB para la instalación completa de cada bitness 1.4 GB para la instalación completa de LabVIEW de 32 y 64 bits

Fuente: <http://www.ni.com/labview/requirements/esa/>

Realizado por: Avalos Edgar, 2015

Nos dirigimos a la página oficial de Scientific Linux para descargar la versión correspondiente tal como nos indica la Tabla 1-2.

Latest Releases of Scientific Linux:

Scientific Linux 7
Install and Live Media – x86_64

Scientific Linux 6 ←
Install and Live Media – x86_64 Install and Live Media – i386

Scientific Linux 5
Install Media – x86_64 Install Media – i386

Figura 1-2. Descarga Sistema Operativo Scientific Linux

Fuente: <https://www.scientificlinux.org/downloads/>

Realizado por: Avalos Edgar, 2015

Ahora debemos seleccionar la versión más apropiada y se despliega la siguiente página donde seleccionamos.



Scientific Linux

<u>Name</u>	<u>Last modified</u>	<u>Size</u>
 Parent Directory		-
 README	21-Jan-2014 17:14	1.6K
 SHA1SUM	12-Nov-2014 12:30	573
 SHA1SUM.gpgsigned	12-Nov-2014 12:30	818
 SHA256SUM	12-Nov-2014 12:30	741
 SHA256SUM.gpgsigned	12-Nov-2014 12:30	1.0K
 SL-6-x86_64-2014-11-05-netinstall.iso	28-Oct-2014 14:08	224M
 SL-6.6-x86_64-2014-11-05-Everything-DVD1.iso	05-Nov-2014 16:47	4.0G
 SL-6.6-x86_64-2014-11-05-Everything-DVD2.iso	05-Nov-2014 16:47	1.6G
 SL-6.6-x86_64-2014-11-05-Install-DVD.iso	05-Nov-2014 15:43	4.0G
 SL-66-x86_64-2014-11-09-LiveCD.iso	12-Nov-2014 11:40	698M
 SL-66-x86_64-2014-11-09-LiveDVD.iso	12-Nov-2014 11:40	2.5G
 SL-66-x86_64-2014-11-09-LiveMiniCD.iso	12-Nov-2014 11:40	581M

Figura 2-2. Versión Sistema Operativo Scientific Linux

Fuente: http://ftp.scientificlinux.org/linux/scientific/6x/x86_64/iso/

Realizado por: Scientific Linux

Procedemos a descargar la versión ISO, SL-6.6-x86_64-2014-11-05-Everything-DVD1.iso y la versión SL-6.6-x86_64-2014-11-05-Everything-DVD2.iso como podemos observar esta versión

tiene un aproximado de 4.0GB y 1.6GB correspondientemente, procedemos a la descarga de esta versión por motivos de compatibilidad ya que es un versión completa

Una vez completada la descarga de estos archivos en imágenes iso, procedemos a la instalación del sistema operativo que se la hará utilizando un software de virtualización de nombre Oracle VM VirtualBox.

2.1.2. Configuración Oracle VM VirtualBox con Scientific Linux 6.6

Para esta investigación científica se procede a utilizar el software para virtualización de sistemas operativos de nombre Oracle VM VirtualBox, mediante este software podemos emular sistemas operativos dentro de otro, es decir si se tiene instalado Windows 8/7 en nuestro computador, mediante este software podemos instalar un nuevo sistema operativo como GNU/LINUX u otro que necesitemos de manera virtual

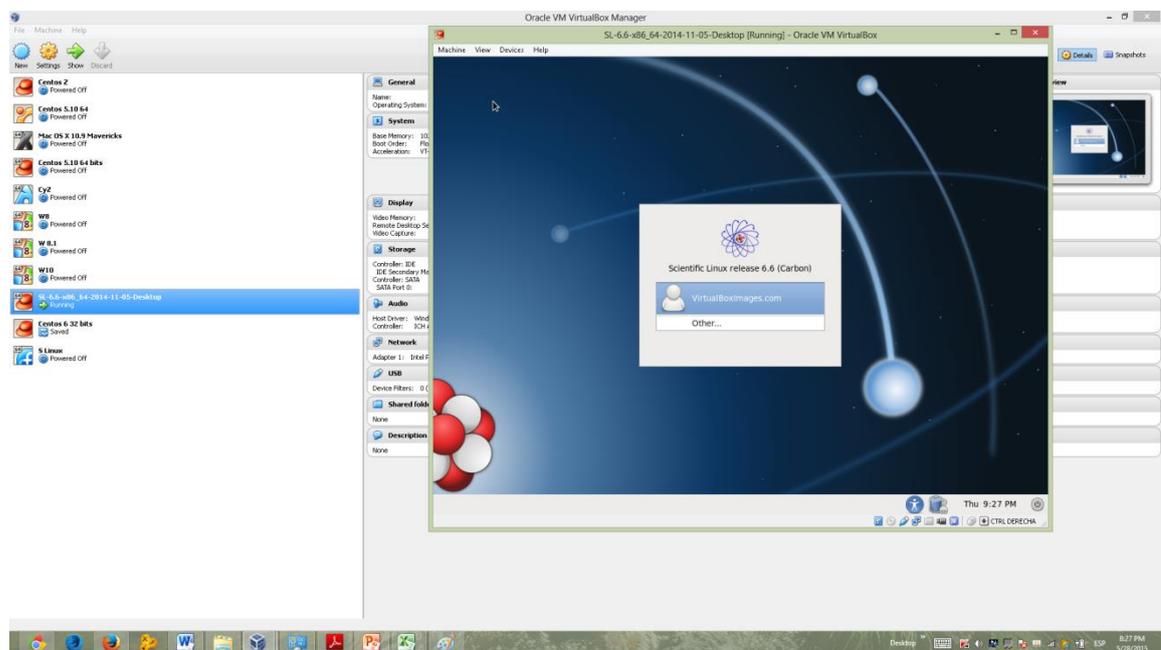


Figura 3-2. Oracle VM VirtualBox

Fuente: Software Virtual Box.

Para instalar Oracle VM VirtualBox nos dirigimos a la página oficial y seleccionamos la versión para Windows tal como indica la Figura 4-2.



Figura 4-2. Instalación Oracle VM VirtualBox

Fuente: <https://www.virtualbox.org/wiki/Downloads>

Una vez descargado el software procedemos a la instalación, damos doble clic en el archivo ejecutable que acabamos de descargar y seguimos los siguientes pasos

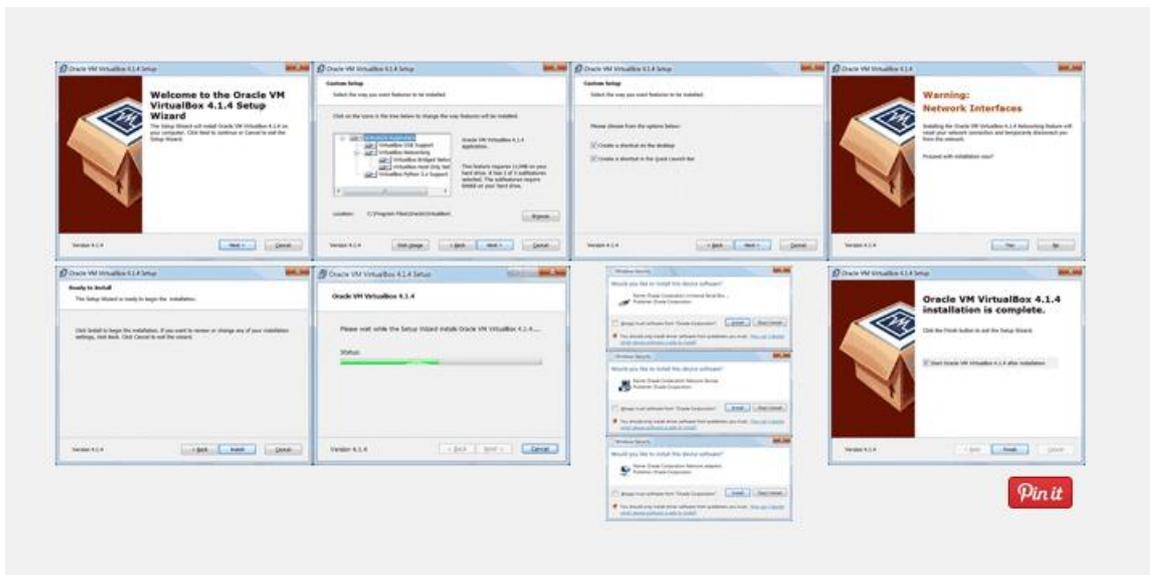


Figura 5-2. Instalación Oracle VM Virtual box

Fuente: http://windowsespanol.about.com/od/AccesoriosYProgramas/ss/Virtualbox-En-Windows_2.htm#step-heading

Una vez instalado la versión compatible de VirtualBox en el ordenador procedemos a la instalación del sistema operativo Scientific Linux previamente ya descargado en imagen .iso inicializamos el software presionando doble clic en el icono Oracle VM Virtual Box como se indica en la Figura 6-2.



Figura 6-2. Icono Oracle VM VirtualBox

Fuente: Software Virtual Box.

Ahora procedemos a cargar la imagen .iso de nuestro sistema operativo para proceder a la instalación.

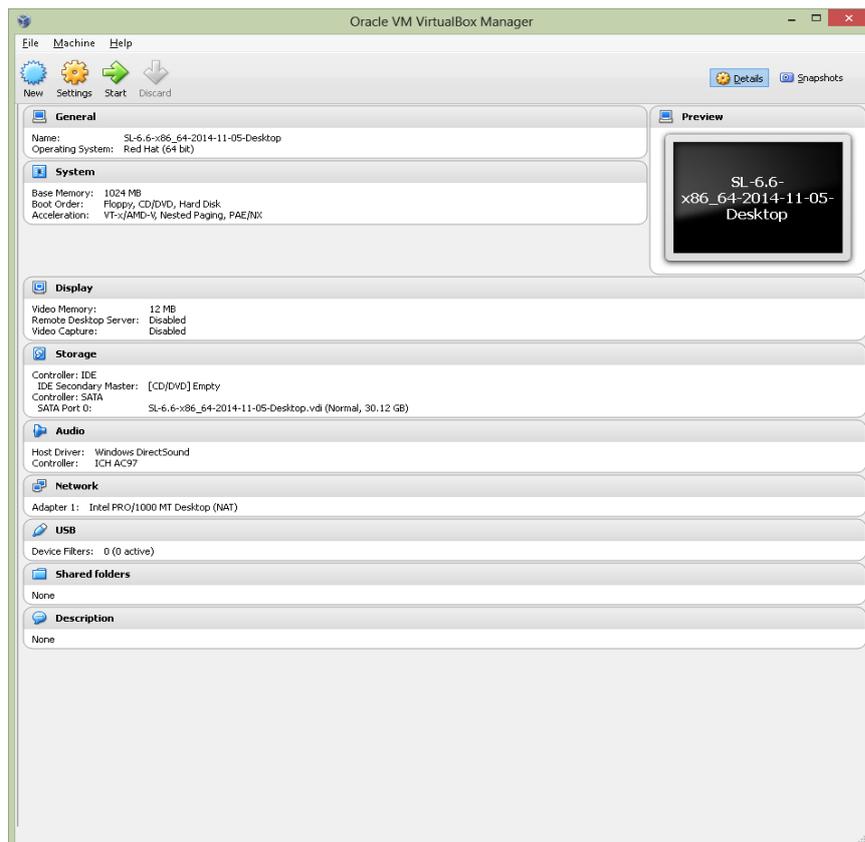


Figura 7-2. Inicio Oracle VM Virtual Box

Fuente: Software Virtual Box.

Como vamos a crear una nueva máquina virtual para proceder a instalar Scientific Linux seleccionamos la opción New como se indica en la Figura 8-2.

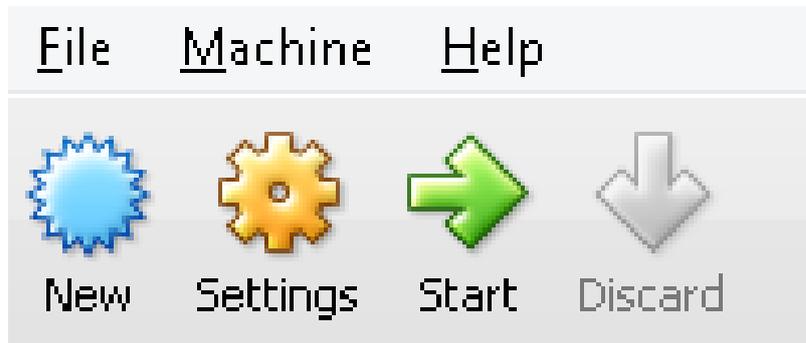


Figura 8-2. Nueva Máquina Virtual

Fuente: Software Virtual Box.

Se despliega la siguiente pantalla donde vamos a proceder a configurar la instalación del nuevo sistema operativo.

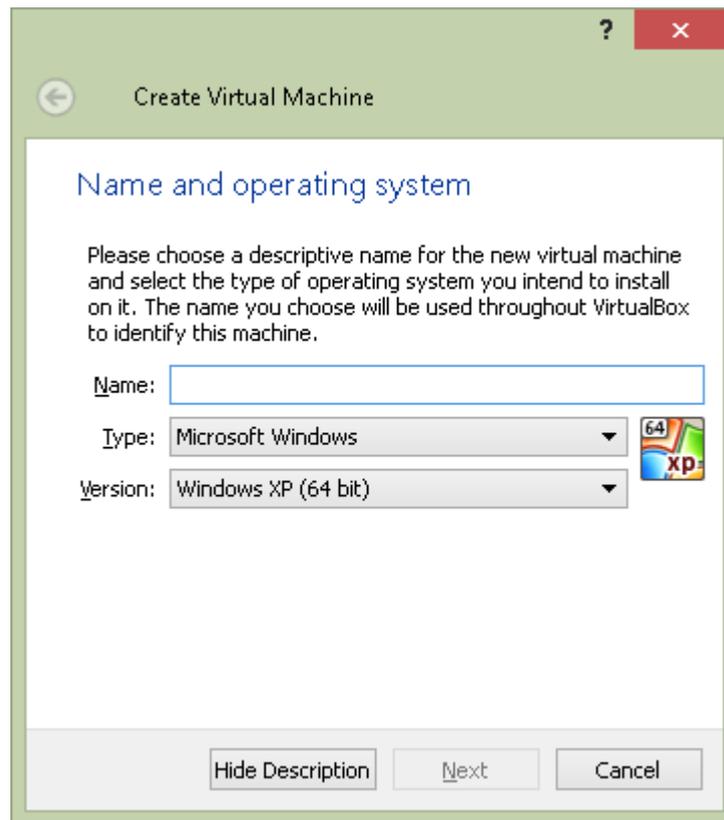


Figura 9-2. Parámetro nuevo Sistema Operativo

Fuente: Software Virtual Box.

Procedemos a modificar tanto el Nombre que queremos que tenga el nuevo sistema operativo de que tipo es y la versión de tal manera que al modificar estas opciones se ajusten a las requeridas

y no tengamos problemas en la instalación. A continuación modificamos los parámetros necesarios con la siguiente información. Y presionamos la opción Next.

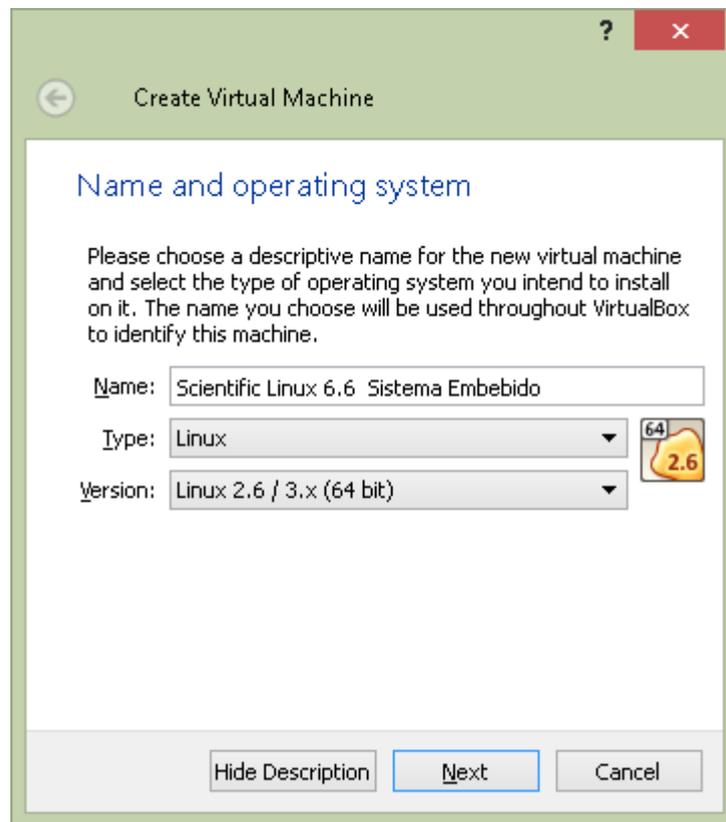


Figura 10-2. Configuración parámetros Máquina Virtual

Fuente: Software Virtual Box.

Como siguiente paso se asigna la memoria RAM con el valor que nosotros creamos conveniente para la instalación del sistema operativo colocamos 4096 Mb que es igual a 4 GB de RAM.

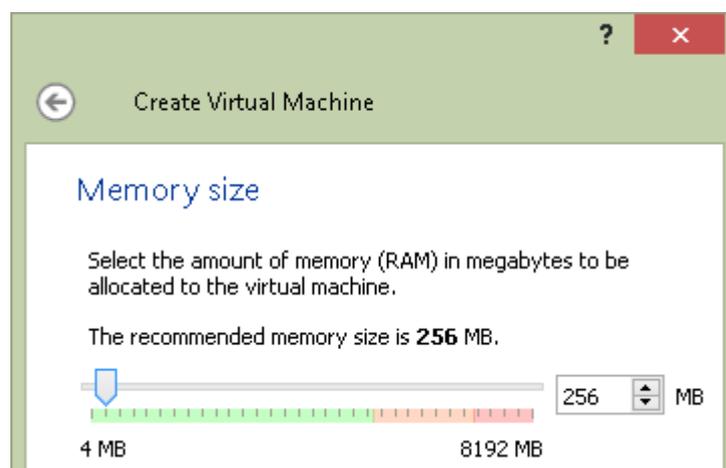


Figura 11-2. Asignación de Memoria RAM

Fuente: Software Virtual Box.

Procedemos a modificar hasta que tengamos la cantidad requerida de memoria RAM 4094 Mb este valor está en la Figura 12-2.

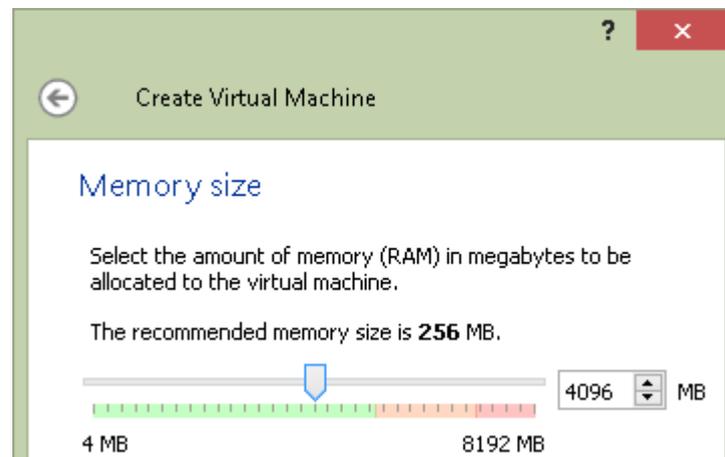


Figura 12-2. Asignación Final memoria RAM

Fuente: Software Virtual Box.

En la siguiente pantalla procedemos a configurar donde se almacenara el sistema operativo, por lo que creamos un nuevo disco virtual, este disco emulara a un disco duro físico, seleccionamos la opción detallada en la Figura 13-2.

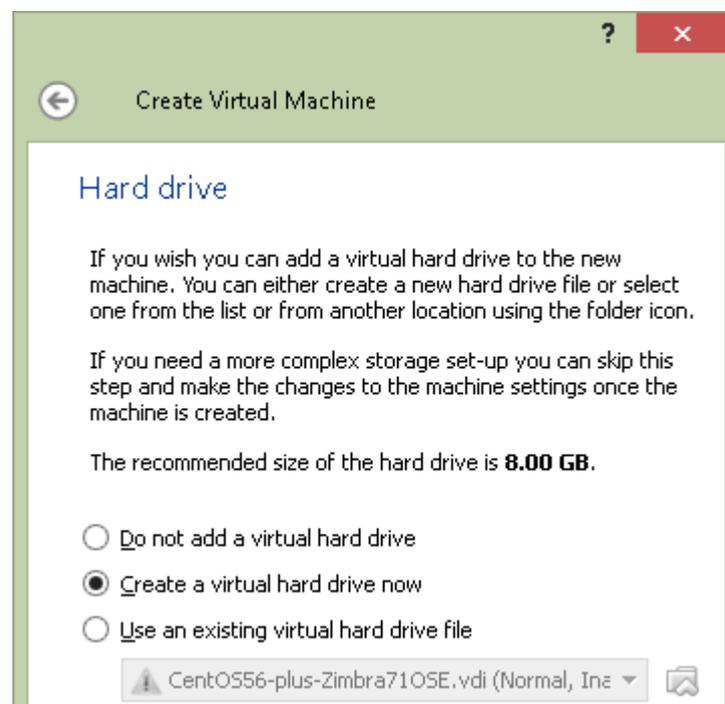


Figura 13-2. Creación Disco Duro Virtual

Fuente: Software Virtual Box.

Procedemos a configurar el tipo de archivo que necesitamos para el disco duro virtual, seleccionamos la opción que esta por defecto y presionamos Next.

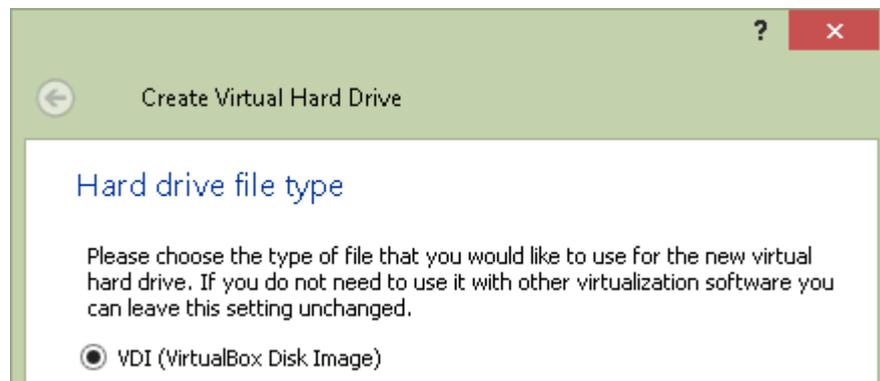


Figura 14-2. Tipo de Archivo Disco Duro Virtual

Fuente: Software Virtual Box.

Para una mejor emulación del sistema operativo seleccionamos la opción que nos da por defecto el software, para asignar el espacio del disco duro virtual

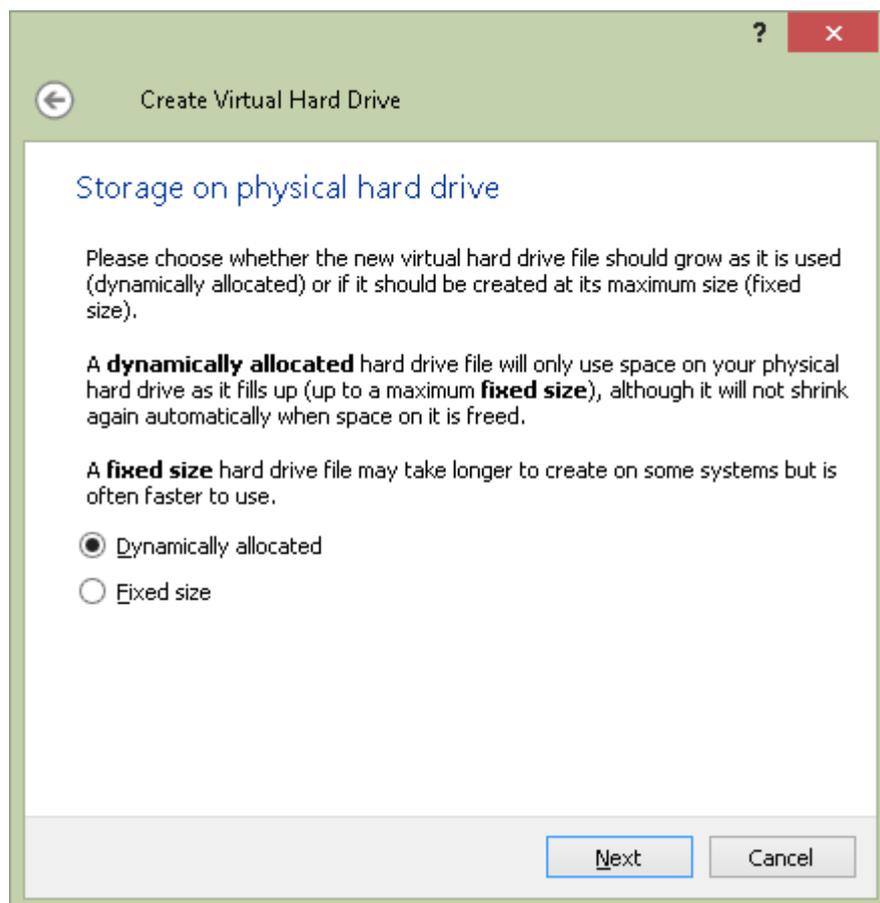


Figura 15-2. Asignación Espacio Disco Duro Virtual

Fuente: Software Virtual Box.

Seleccionamos la opción (Dynamically allocated), el espacio que ocupa el disco duro virtual ira creciendo conforme vayamos guardando archivos, caso contrario con lo que pasaría si seleccionamos la opción Fixed size que se refiere al tamaño fijo. Como siguiente paso seleccionamos la ubicación del sistema operativo virtual y le asignamos un tamaño 8.00 GB

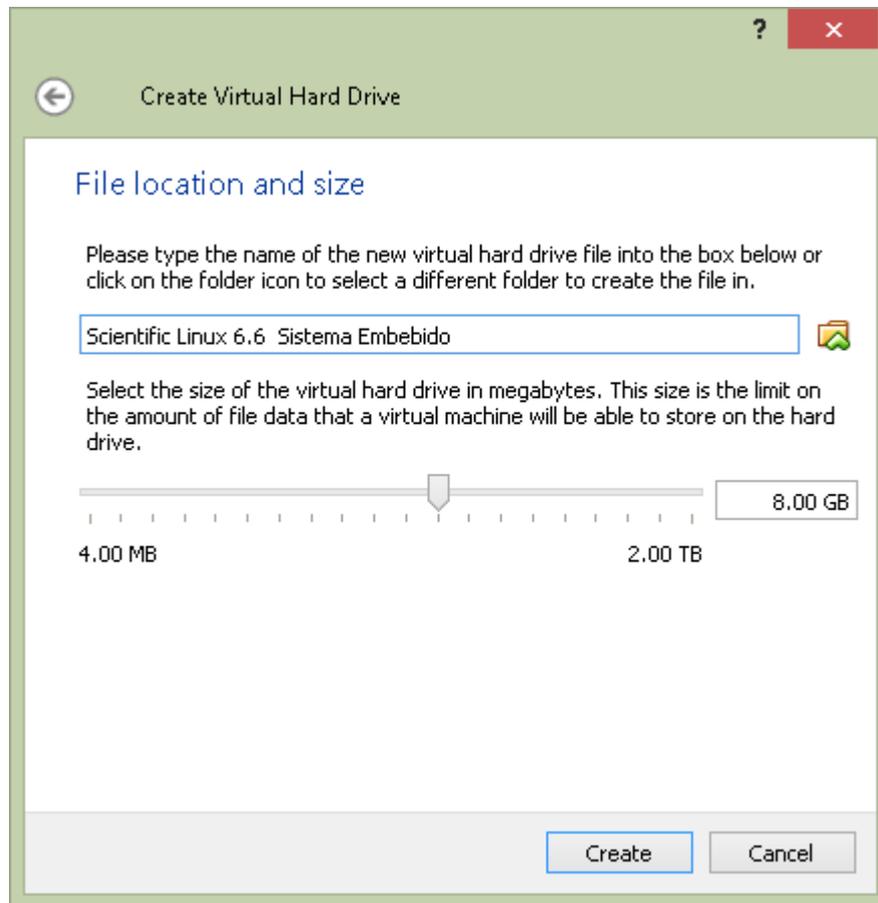


Figura 16-2 Ubicación y Tamaño del Sistema Operativo

Fuente: Software Virtual Box.

Presionamos la opción Start para iniciar con la instalación del sistema operativo esta opción que esta detallada con una flecha azul en la Figura 17-2, pone en marcha el nuevo sistema operativo virtual con las mismas funciones y características de instalación de uno real o físico.

Estas cualidades que nos ofrece el software virtual box son de gran ayuda a estudiantes quienes dependiendo las capacidades de la computadora donde se albergara las máquinas virtuales pueden emular diferentes sistemas operativos de diferentes arquitecturas y fabricantes, donde se pueden hacer pruebas en red además de diferentes tareas como si se estuviera trabajando en varios equipos físicos, solo que mediante la emulación o virtualización se ahorra muchos recursos sin la necesidad de comprar nuevos equipos.

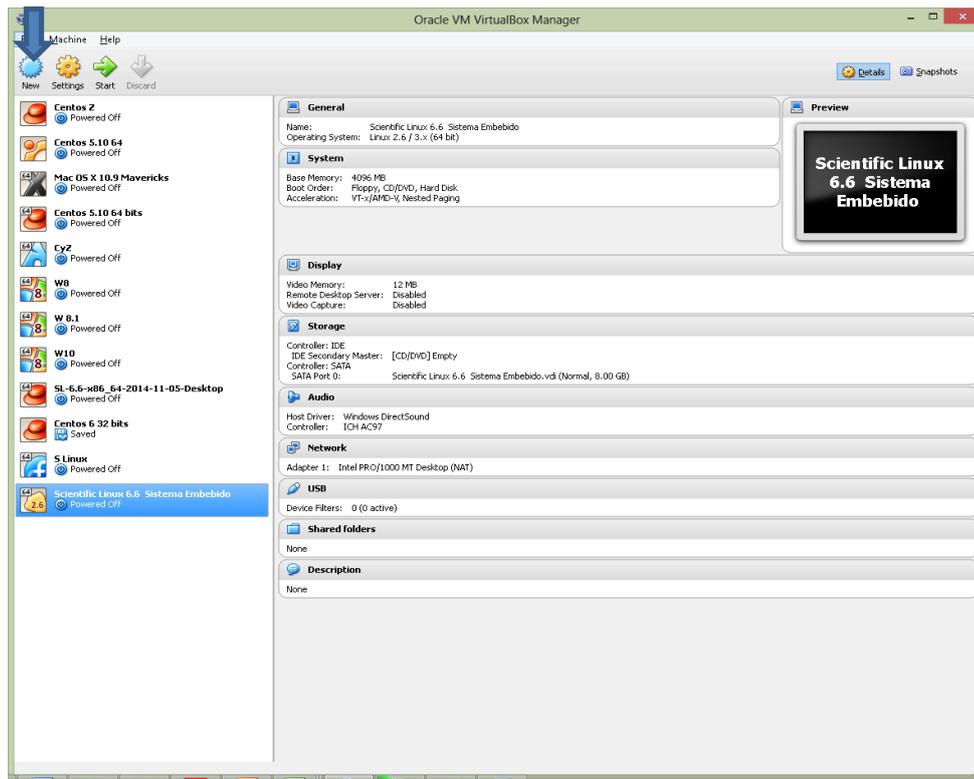


Figura 17-2. Instalación Sistema Operativo

Fuente: Software Virtual Box.

Seleccionamos la ubicación de la imagen ISO del Sistema Operativo descargado, y presionamos la opción Start como siguiente paso.

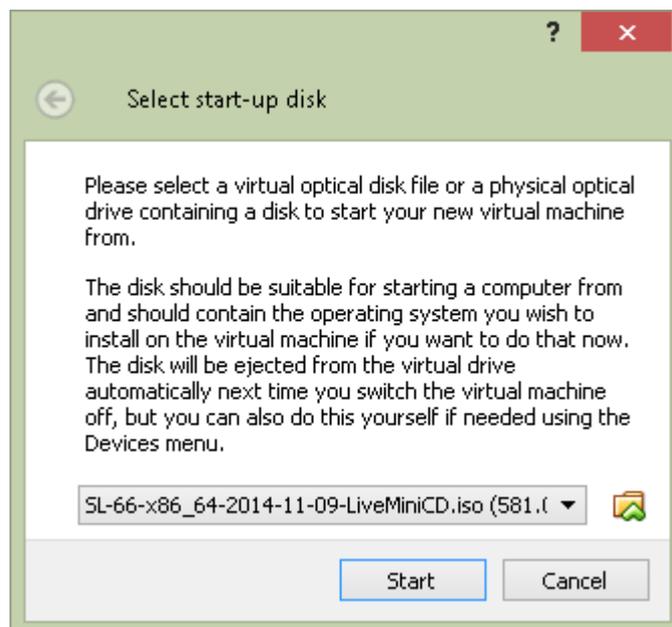


Figura 18-2. Ubicación imagen ISO archivo descargado

Fuente: Software Virtual Box.

Procedemos a seleccionar la imagen ISO del sistema operativo descargado, para empezar con la instalación del mismo.



Figura 19-2. Instalación Inicio Scientific Linux 6

Fuente: Scientific Linux 6.6.

El sistema operativo se ha cargado con éxito, esperamos a que el contador termine y presionamos enter, tenemos las siguientes opciones de instalación. Seleccionamos la opción install y continuamos.

El menú nos despliega además más opciones como el Boot, donde seleccionamos la ubicación del sistema operativo a instalarse, además de la opción Boot (Text Mode), donde seleccionamos y con comandos como deseamos arrancar la instalación.

La opción Boot (Basic Video) carga los controladores básicos de video, esta opción de la puede utilizar para computadores de pocos recursos, si seleccionamos la opción Install (Text Mode) debemos ingresar mediante comando la dirección donde se encuentra los archivos necesarios para la instalación, si en lugar de esa opción elegimos Memory Test, se realizara una comprobación de todos los recursos necesarios para la instalación, y para instalar este sistema operativo necesitamos realizarlo desde un disco o memoria externa utilizamos para este procedimiento la opción, Boot from local drive.

Cuya traducción más adecuada es cargar desde un disco local, es decir los archivos de instalación estarían almacenados en un disco extraíble y de ahí empezaría la instalación.

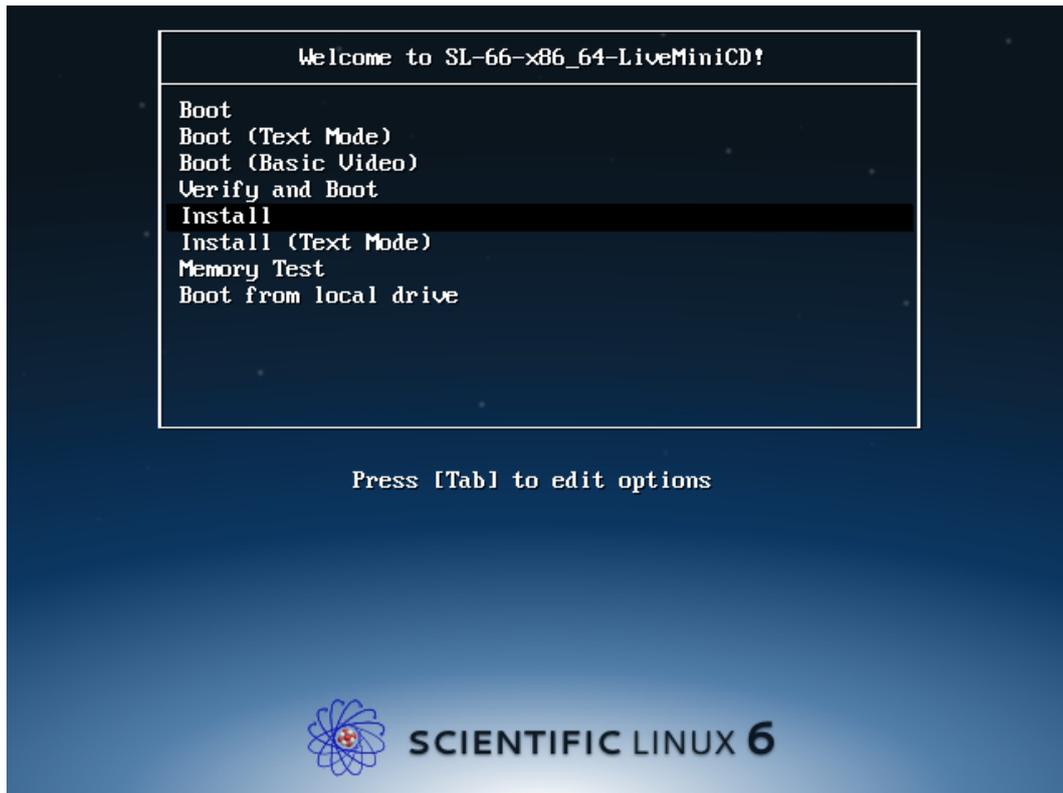


Figura 20-2. Opciones de instalación Scientific Linux

Fuente: Scientific Linux 6.6.

Hasta este paso de la instalación no se ha registrado ningún error, empiezan a cargarse los archivos necesarios que se necesita para instalar la distribución.

```

ahci 0000:00:0d.0: AHCI 0001.0100 32 slots 1 ports 3 Gbps 0x1 impl SATA mode
ahci 0000:00:0d.0: flags: 64bit ncq stag only ccc
scsi2 : ahci
ata3: SATA max UDMA/133 abar m819200xf0806000 port 0xf0806100 irq 21
ata3: SATA link up 3.0 Gbps (SStatus 123 SControl 300)
ata3.00: ATA-6: UBOX HARDDISK, 1.0, max UDMA/133
ata3.00: 16777216 sectors, multi 128: LBA48 NCQ (depth 31/32)
ata3.00: configured for UDMA/133
scsi 2:0:0:0: Direct-Access    ATA                UBOX HARDDISK    1.0  PQ: 0 ANSI: 5
sr0: scsi3-mmc drive: 32x/32x xa/form2 tray
Uniform CD-ROM driver Revision: 3.20
STARTING CRC_T10DIF
sd 2:0:0:0: [sda] 16777216 512-byte logical blocks: (8.58 GB/8.00 GiB)
sd 2:0:0:0: [sda] Write Protect is off
sd 2:0:0:0: [sda] Write cache: enabled, read cache: enabled, doesn't support DPO
or FUA
sda: unknown partition table
sd 2:0:0:0: [sda] Attached SCSI disk
squashfs: version 4.0 (2009/01/31) Phillip Lougher
EXT4-fs (dm-0): mounted filesystem with ordered data mode. Opts:
dracut: Mounted root filesystem /dev/mapper/live-rw
dracut: Loading SELinux policy
type=1404 audit(1432777126.466:2): enforcing=1 old_enforcing=0 auid=4294967295 s
es=4294967295

```

Figura 21-2. Instalación Scientific Linux archivos primarios

Fuente: Scientific Linux 6.6.

Continúa la carga de archivos esta vez se genera un mensaje OK que indica que todos los archivos necesarios para la instalación se cargan de manera de correcta.

```
iptables: Applying firewall rules: ip_tables: (C) 2000-2006 Netfilter Core Team [ OK ]
Starting system logger: [ OK ]
Starting irqbalance: [ OK ]
Starting rpcbind: [ OK ]
Starting lldpad: [ OK ]
Starting FCoE initiator service: 802.1Q VLAN Support v1.8 Ben Greear <greearb@candelatech.com>
All bugs added by David S. Miller <davem@redhat.com>
bnx2fc: Broadcom NetXtreme II FCoE Driver bnx2fc v2.4.2 (Dec 11, 2013) [ OK ]
netlink: 12 bytes leftover after parsing attributes.
netlink: 12 bytes leftover after parsing attributes.
netlink: 12 bytes leftover after parsing attributes.
Starting system message bus: [ OK ]
Setting network parameters... [ OK ]
Starting NetworkManager daemon: [ OK ]
e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
ADDRCONF(NETDEV_UP): eth0: link is not ready
8021q: adding VLAN 0 to HW filter on device eth0
ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Starting NFS statd: [ OK ]
```

Figura 22-2. Instalación Scientific Linux carga procesos

Fuente: Scientific Linux 6.6.

Finalizando la pantalla anterior podemos observar que se ha cargado la primera imagen



Figura 23-2. Instalación Scientific Linux presentación

Fuente: Scientific Linux 6.6.

Configuramos el idioma a utilizar durante la instalación, como podemos observar tenemos varias opciones, seleccionamos idioma español.

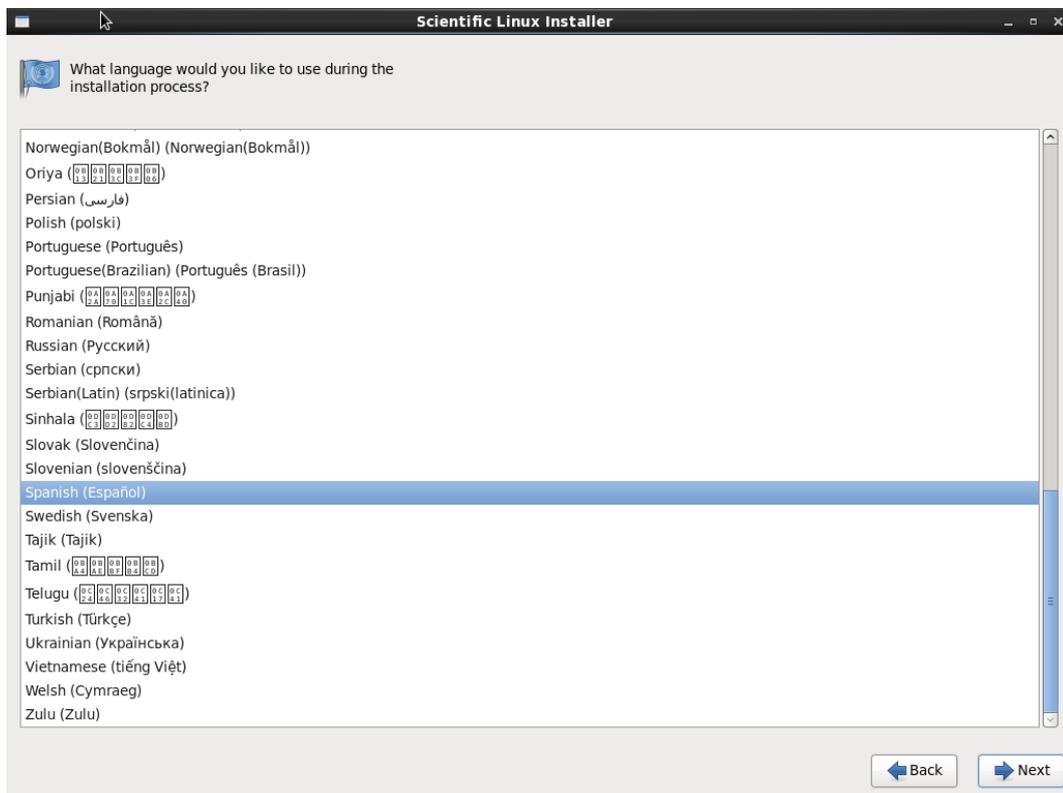


Figura 24-2. Instalación Scientific Linux selección idioma

Fuente: Scientific Linux 6.6.

Procedemos a configurar el idioma del teclado, se muestra varias opciones, escogemos Latin América, presionamos además la tecla Next, para el siguiente paso.



Figura 25-2. Instalación Scientific Linux idioma teclado

Fuente: Scientific Linux 6.6.

La siguiente pantalla de instalación asignamos el tipo de dispositivo de almacenamiento del sistema operativo, es importante aclarar que debemos seleccionar la opción que nos da por defecto para obtener un instalación sin errores

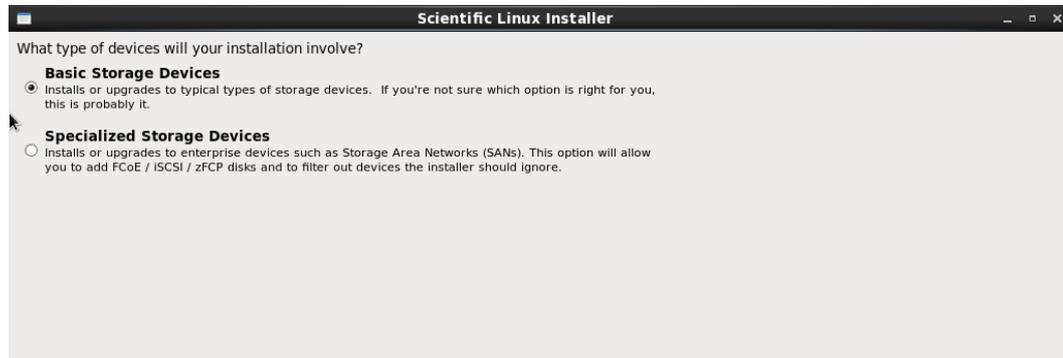


Figura 26-2. Instalación Scientific Linux tipo de dispositivos de almacenamiento

Fuente: Scientific Linux 6.6.

Como siguiente paso se procede a seleccionar el tipo de dispositivo de almacenamiento donde se instalara el sistema operativo.



Figura 27-2. Instalación Scientific Linux formato disco duro virtual

Fuente: Scientific Linux 6.6.

Para lograr una instalación limpia procedemos a descartar los datos almacenados en el disco, seleccionamos Yes, discard any data seleccionamos la opción Next y continuamos. Determinamos la ubicación en la cual nos encontramos, elegimos Guayaquil, Ecuador

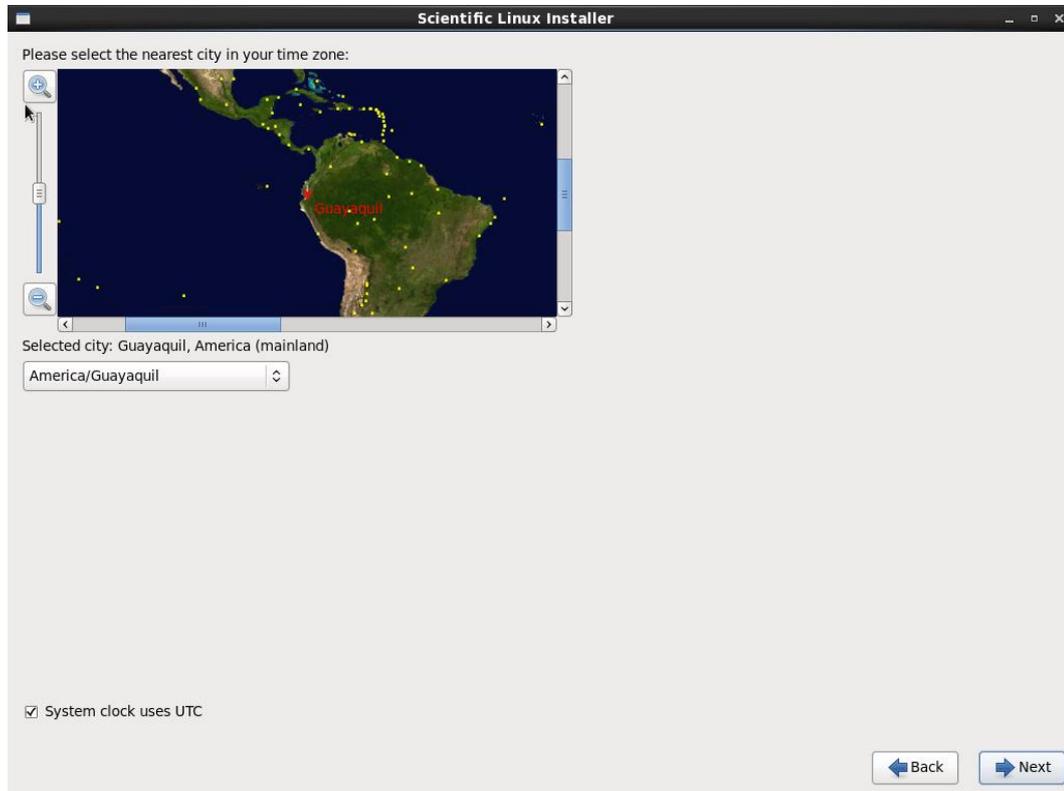


Figura 28-2. Instalación Scientific Linux selección ubicación

Fuente: Scientific Linux 6.6.

Configuramos a continuación la contraseña para el súper Usuario o root este usuario tendrá privilegios totales para realizar tareas complejas como actualizar cometidos o la instalación de programas



Figura 29-2. Instalación Scientific Linux contraseña root

Fuente: Scientific Linux 6.6.

Usamos todo el espacio disponible para la instalación con la opción que nos muestra por defecto.

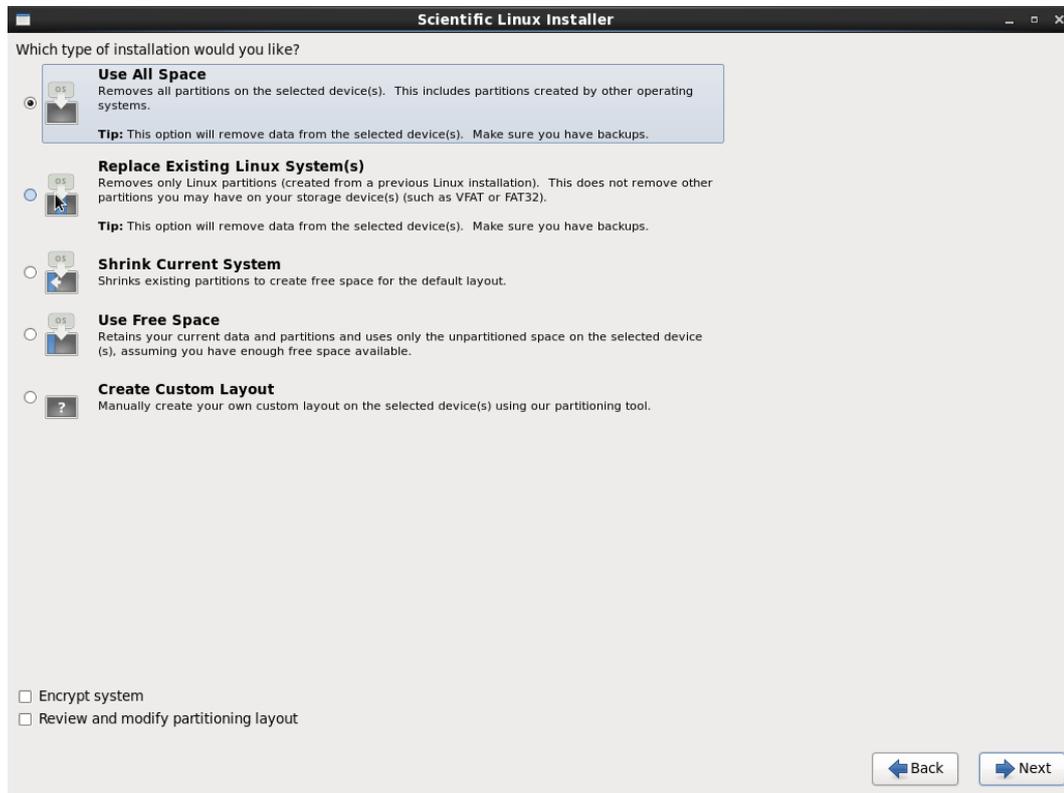


Figura 30-2. Tipo de instalación Scientific Linux

Fuente: Scientific Linux 6.6.

Se cargan todos los archivos necesarios para iniciar con la instalación

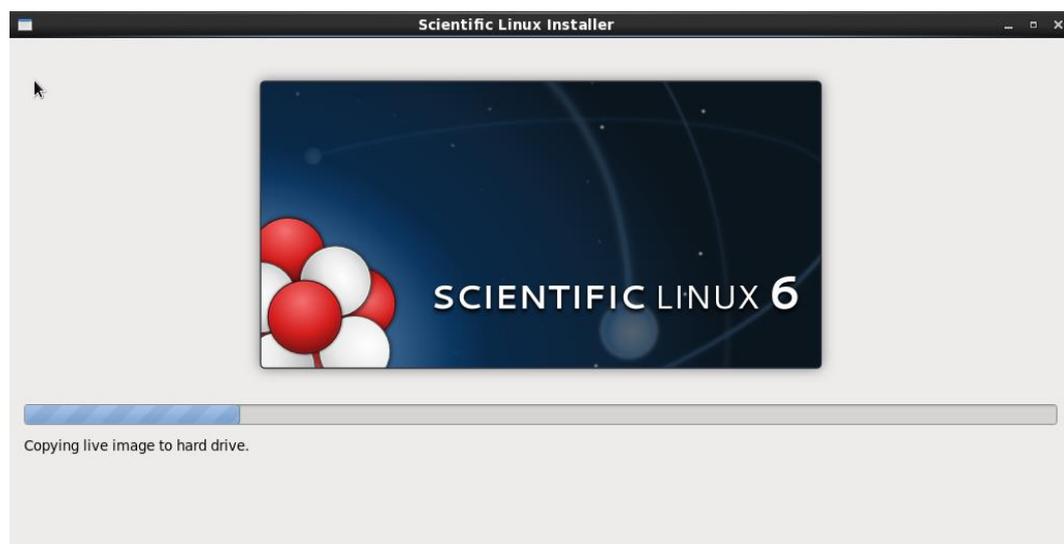


Figura 31-2. Instalación Scientific Linux inicialización

Fuente: Scientific Linux 6.6.

Terminando con los archivos necesarios para el primer arranque del sistema operativo.

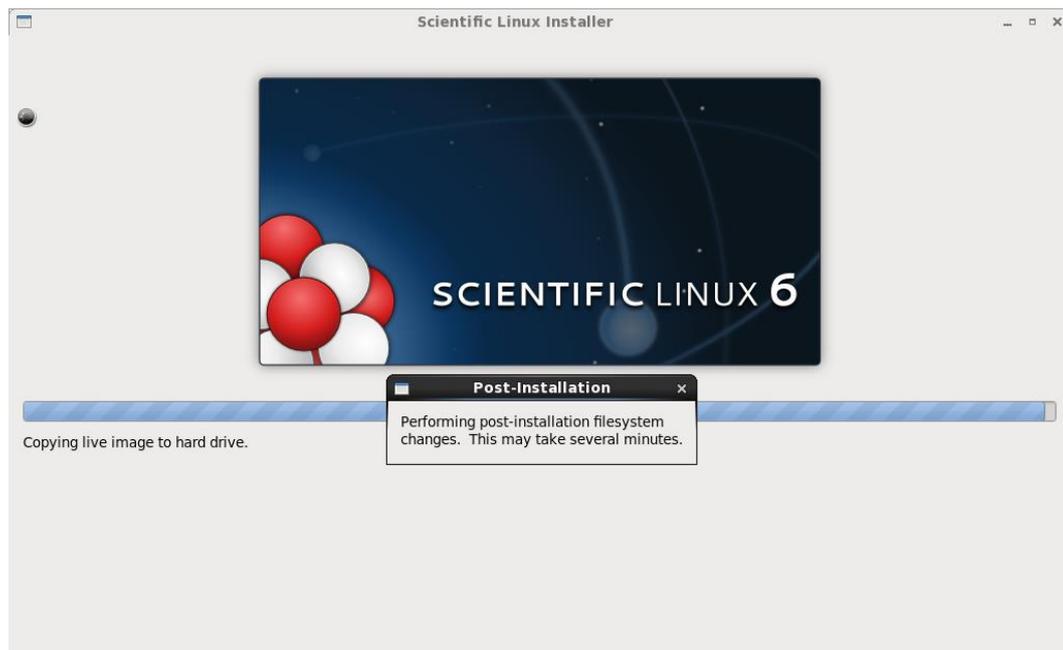


Figura 32-2. Instalación Scientific Linux finalización

Fuente: Scientific Linux 6.6.

Termina la instalación con el siguiente mensaje el cual nos indica que debemos reiniciar.

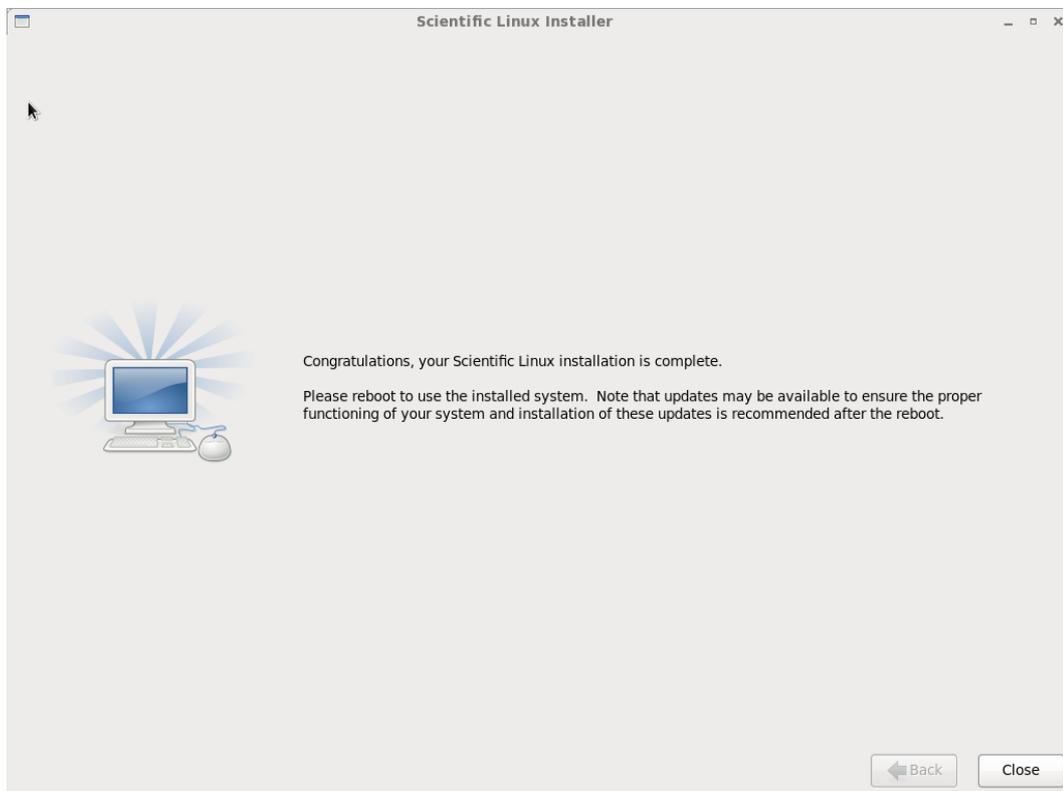


Figura 33-2. Instalación Scientific Linux instalación completa

Fuente: Scientific Linux 6.6.

Empieza el primer inicio del sistema operativo



Figura 34-2. Instalación Scientific Linux primer inicio

Fuente: Scientific Linux 6.6.

Terminando de cargar archivos necesarios para el primer inicio del sistema



Figura 35-2. Inicio sistema operativo Scientific Linux

Fuente: Scientific Linux 6.6.

Ingresamos la contraseña del súper usuario root la cual la definimos en la instalación.

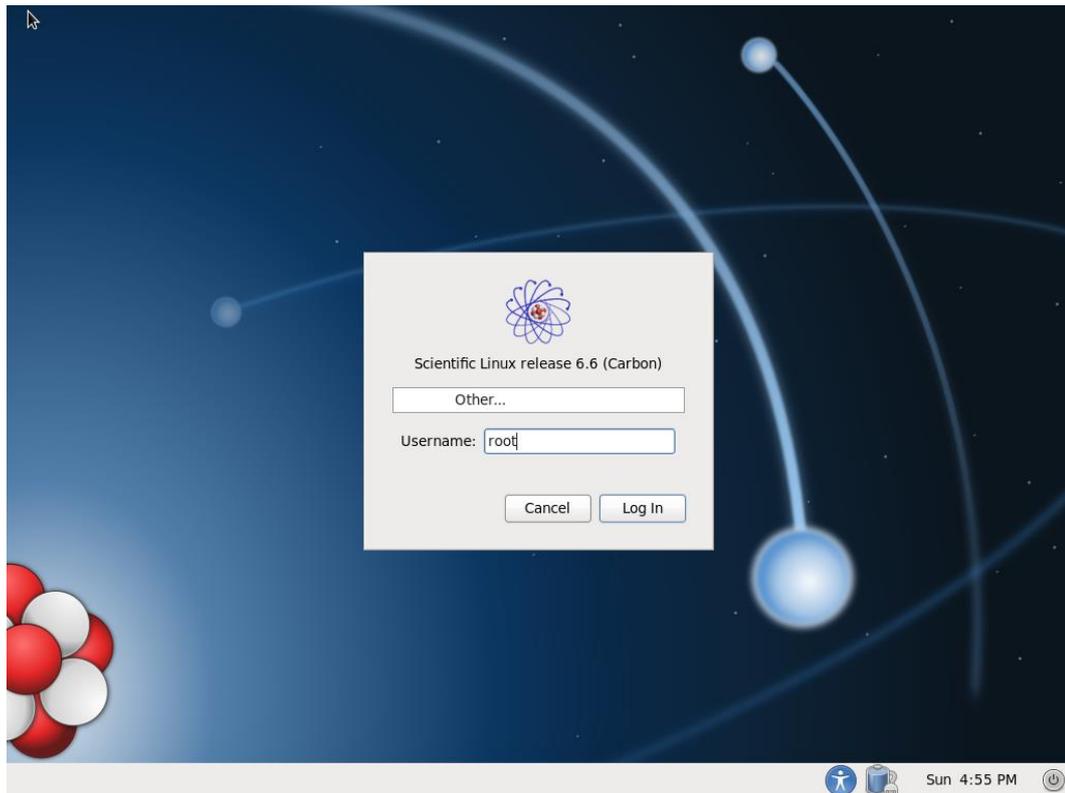


Figura 36-2. Ingreso súper usuario root

Fuente: Scientific Linux 6.6.

Ingresamos la contraseña además presionamos el botón Log In

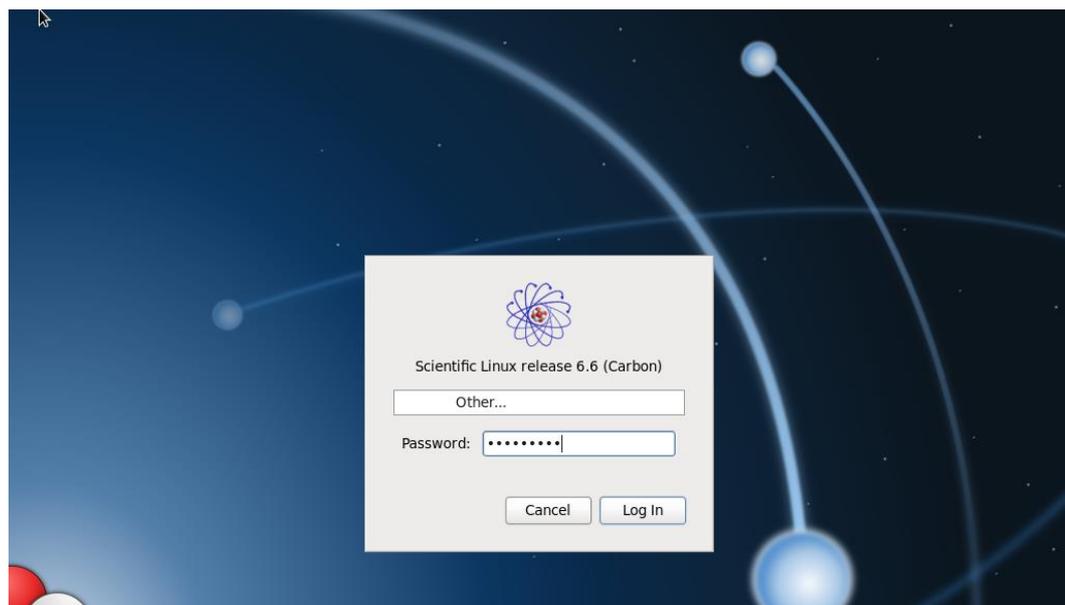


Figura 37-2. Contraseña súper usuario

Fuente: Scientific Linux 6.6.

Pantalla principal del primer inicio del sistema operativo encontramos opciones como la fecha de inicio, el usuario con el que ingresamos, el icono de navegador por defecto, la consola de comandos entre otras opciones importantes.

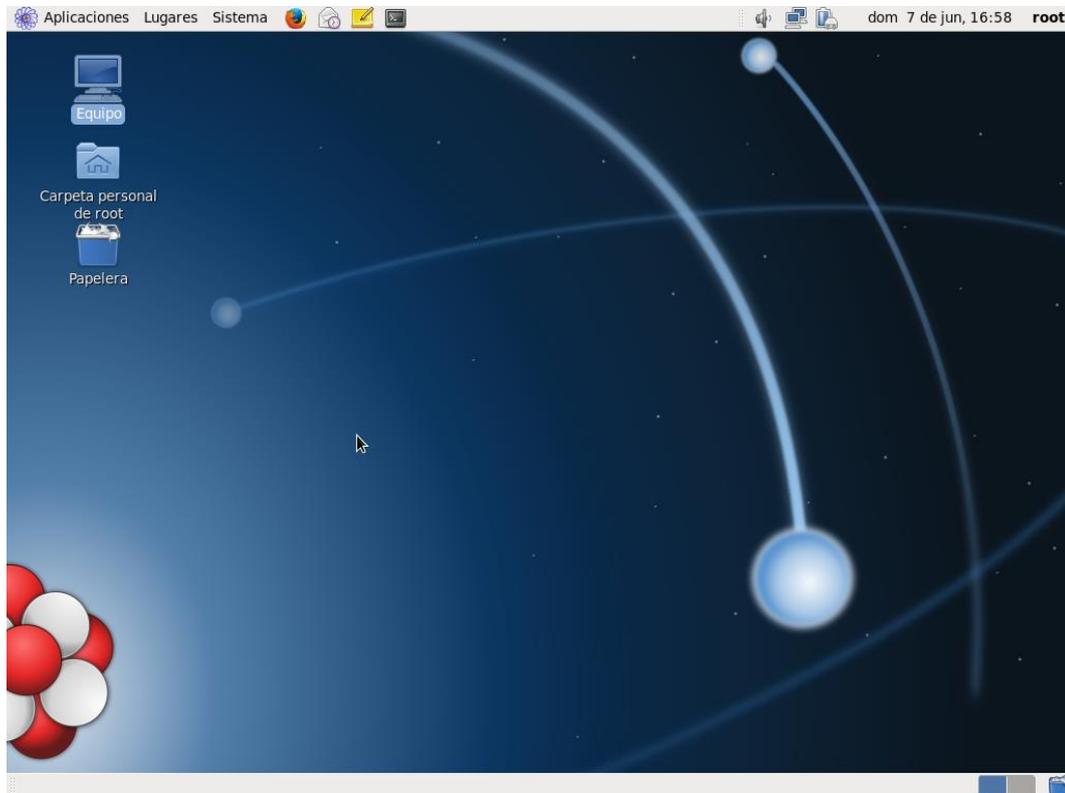


Figura 38-2. Pantalla principal Sistema Operativo Scientific Linux

Fuente: Scientific Linux 6.6.

2.1.3. Instalación Labview sobre Scientific Linux

Para la instalación de la herramienta de diseño se sistemas embebidos sobre Linux, utilizamos la consola de comandos o terminal, para instalar programas debemos ingresar con la cuenta de súper usuario o root, .la contraseña de esta cuenta la definimos en la instalación.

Nos colocamos en el terminal con el comando `cd` para cambiarnos de directorio de ser el caso tan como se indica en a Figura 39-2. Con este comando podemos acceder a una carpeta específica, o que se encuentra en el directorio actual, esta consola de comando es de mucha ayuda ya que por medio de líneas de código podemos hacer las mismas opciones que se las hace de manera gráfica, como por ejemplo abrir una carpeta, entre otras tareas adicionales.

Esta forma de navegar por medio de comandos es de mucha ayuda para cuando nos encontramos en sistemas operativos sin una interfaz gráfica y la única manera de movernos por él sistema es por medio de comandos.

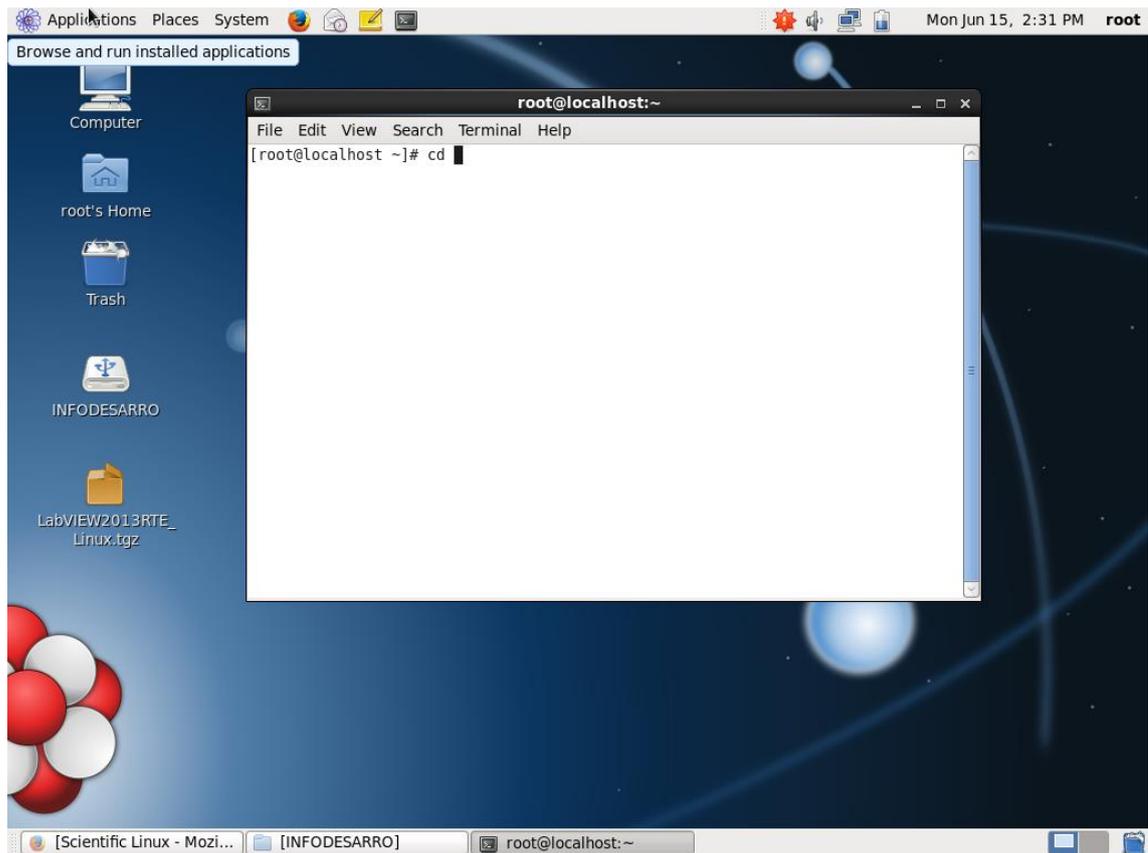


Figura 39-2. Comandos Terminal root

Fuente: Scientific Linux 6.6.

Con el comando `pwd` imprime el directorio en el que no encontramos es de mucha ayuda este comando si no sabemos en qué lugar o en que directorio estamos, presionamos la tecla enter para ejecutar el comando.

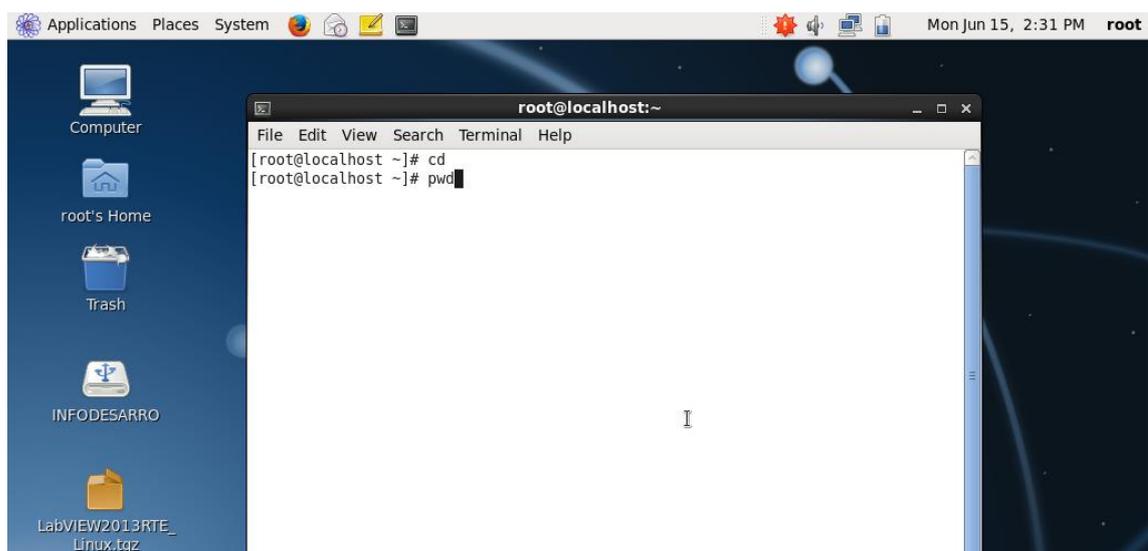


Figura 40-2. Comando Ubicación en Directorio

Fuente: Scientific Linux 6.6.

Nos ubicamos en el directorio donde se encuentra nuestro archivo de instalación con el comando ls, este comando nos despliega todos los directorios disponibles en la ubicación determinada seleccionamos escritorio porque ahí se encuentra el archivo descargado con el software para diseño de sistemas embebidos Labview para Linux.

Con el comando cd Escritorio/ nos ubicamos en el directorio ejecutamos con enter, listamos otra vez los archivos con el comando ls, observamos en la Figura 41-2 que con letras de color verde hemos encontrado el archivo con el nombre labVIEW2013RTE_Linux.tgz, procedemos a continuación a descomprimirlos con los comandos tar zxvf labVIEW2013RTE_Linux.tgz ejecutamos con enter.

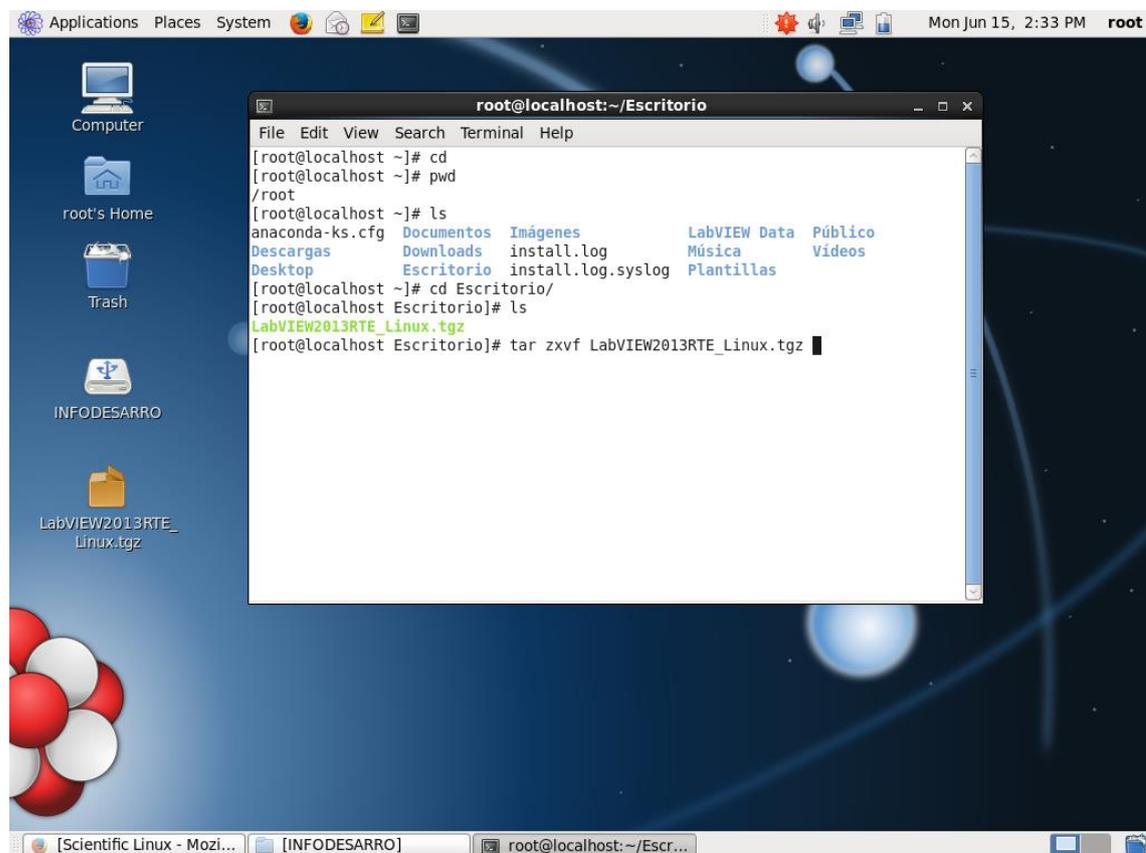


Figura 41-2. Comandos para la instalación de Labview

Fuente: Scientific Linux 6.6.

Al ejecutar con enter la sentencia para descomprimir el archivo de instalación, observamos en la Figura 42-2, el número de archivos con el cual está conformado el diseñador de sistemas embebidos Labview.

Como siguiente punto de la instalación debemos ir de manera ordenada instalando todos los archivos siguientes que hemos descomprimido, notar que todos los archivos con extensión .rpm Son los que vamos a instalar.

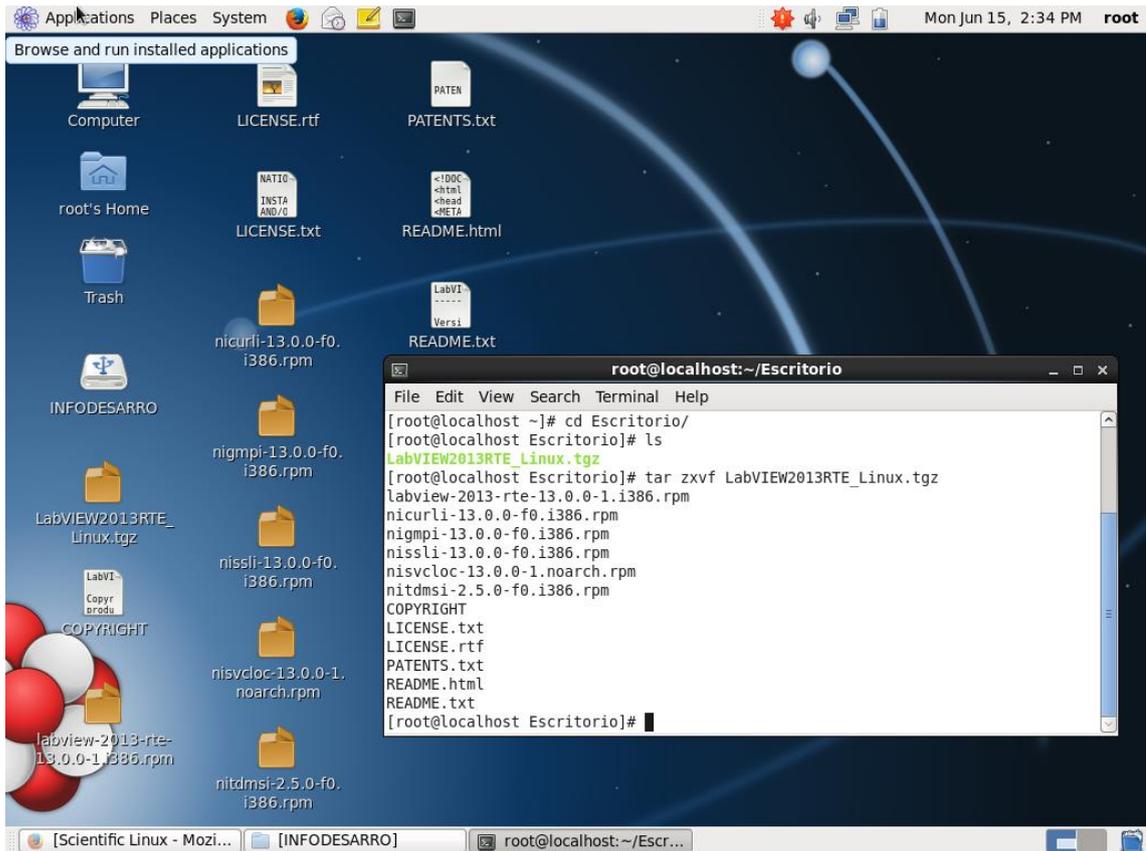


Figura 42-2. Archivos descomprimidos Labview

Fuente: Scientific Linux 6.6.

El primer archivo que necesitamos para empezar con la instalación es mesa-libGL.1686 el cual es una librería grafica necesaria para poder visualizar varios componentes del software a instalarse para ejecutar la sentencia presionamos la tecla enter.

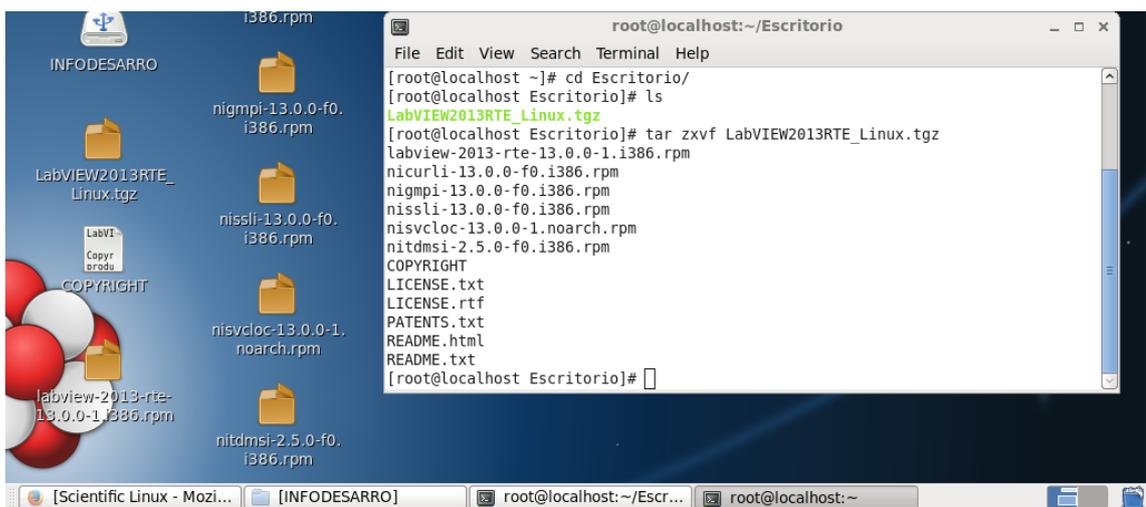


Figura 43-2. Archivos necesarios para la instalación

Fuente: Scientific Linux 6.6.

En este punto ya hemos actualizado todos los archivos necesarios, procedemos ahora a la instalación con la siguiente sentencia en la cual se encuentra ya el software de diseño para sistemas embebidos, **rpm -i Labview-2013-rte-13.0.0-1.i386.rpm** tal como se observa en la Figura 46-2.

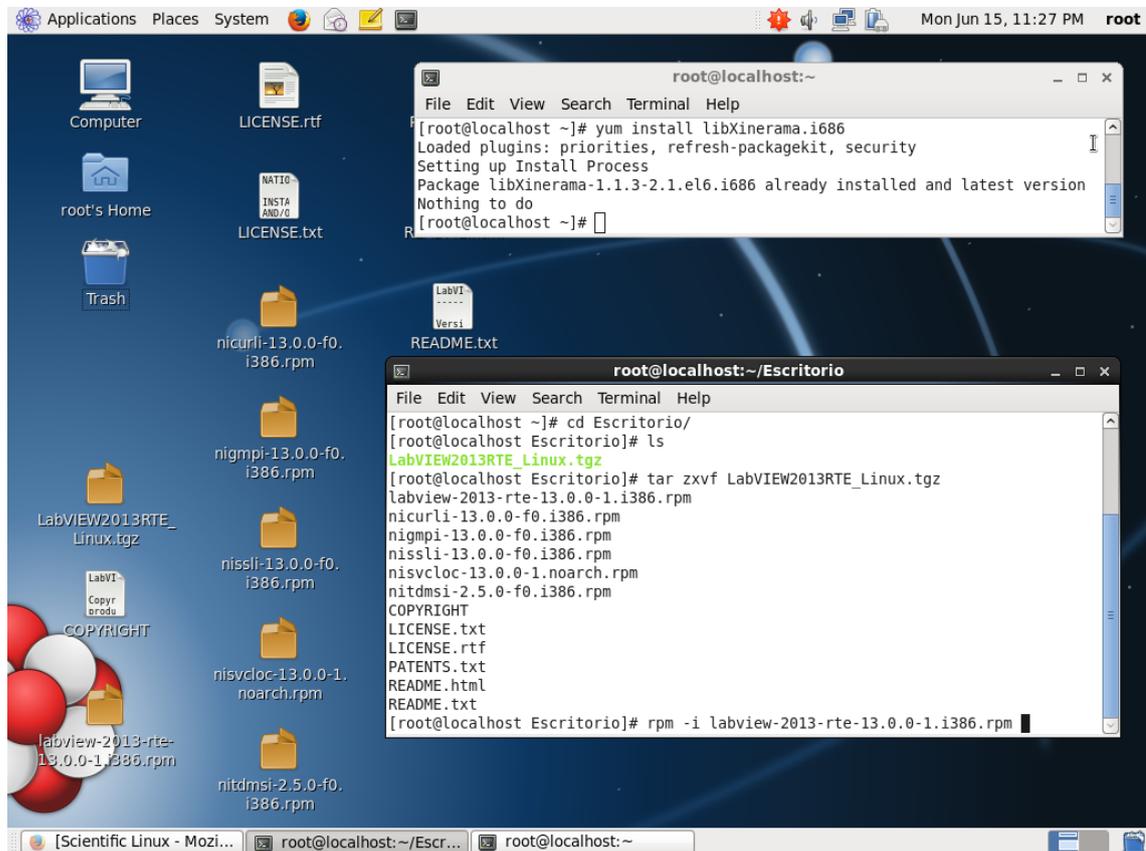


Figura 46-2. Instalación Labview comandos

Fuente: Scientific Linux 6.6.

Ejecutamos la anterior sentencia, observamos cómo se ejecuta la instalación y termina con un 100% de éxito como se observa en la Figura 47-2.

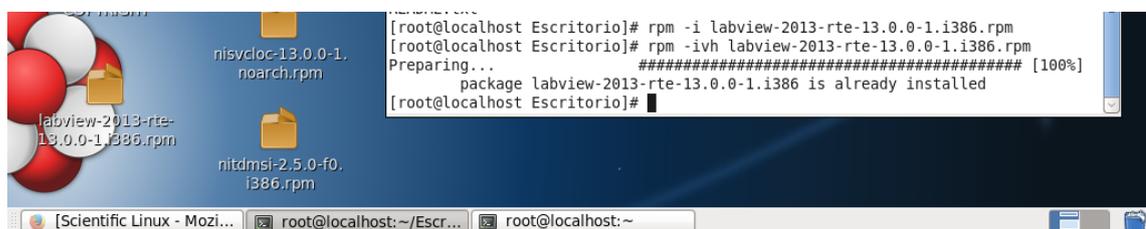


Figura 47-2. Archivos de Instalación Labview

Fuente: Scientific Linux 6.6.

Se continúa con los siguientes archivos necesarios para que se ejecute el software, instalamos el archivo siguiente con la sentencia **rpm -ivh nissli-13.0.0-f0.i386.rpm** y lo ejecutamos.

Continuamos con la siguiente sentencia **rpm -ivh nisvcloc-13.0.0-1.noarch.rpm**, observamos en la Figura 48-2 como se instalan los archivos.

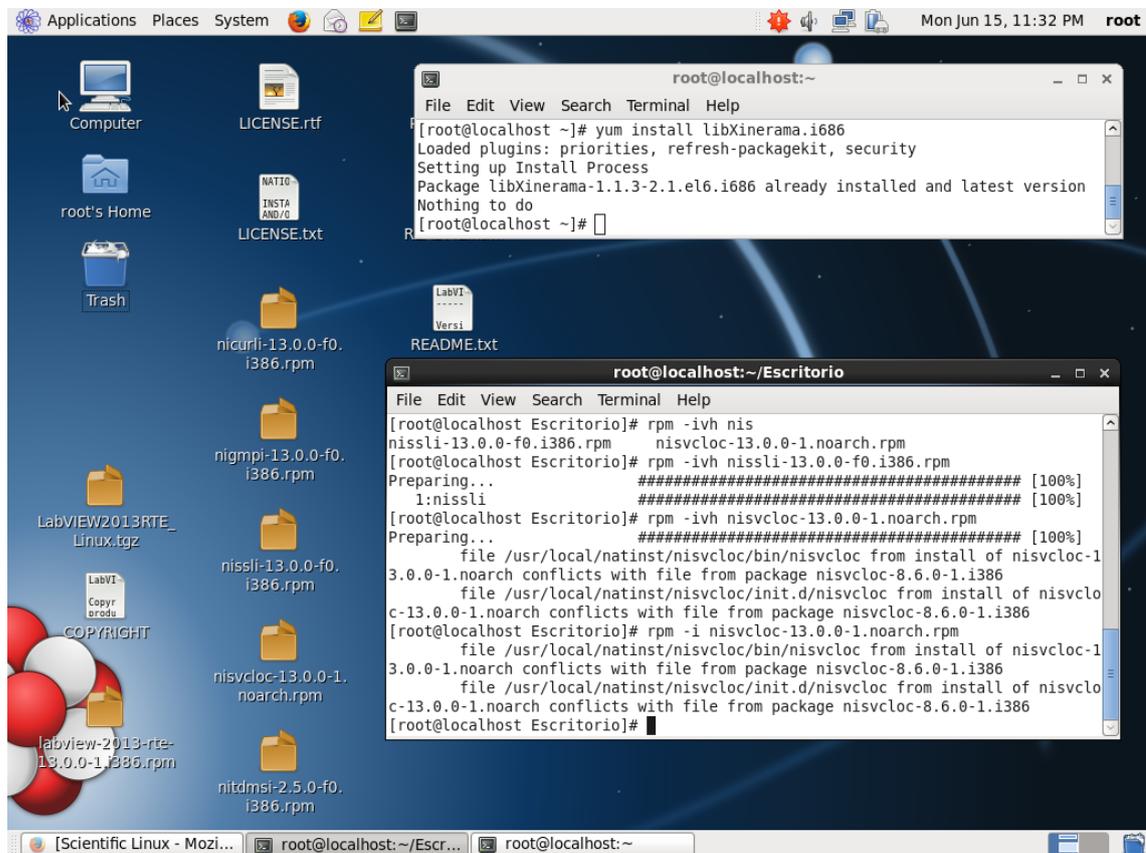


Figura 48-2. Archivos necesarios Instalación

Fuente: Scientific Linux 6.6.

Instalamos el ultimo archivo **rpm -ivh nitdmsi-2.5.0-f0.i386.rpm** observando el avance de 100% que nos indica que todo ha terminado podemos encontrar este procedimiento en la Figura 49-2.

Instalados todos estos archivos procedemos a ejecutar por primera vez el software, todos los archivos para la instalación de Labview se han ejecutado sin ningún mensaje de error.

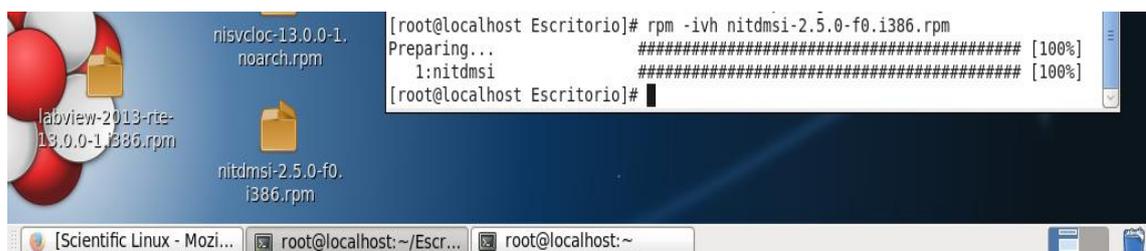


Figura 49-2. Archivos de Instalación

Fuente: Scientific Linux 6.6.

2.1.4. Labview sobre Scientific Linux

Ejecutamos Labview en Scientific Linux por primera vez de la siguiente manera. En la barra de tareas en la opción Applications, seleccionamos National Instruments, podemos observar ya el logo de Labview damos clic sobre el icono y esperamos a que cargue el software.



Figura 50-2. Inicio Labview

Fuente: Scientific Linux 6.6.

A continuación tenemos el primer inicio de Labview. La versión 8.6 es la versión más adecuada para la Scientific Linux 6.6, se procede a la primera vez que se ejecuta el software para edición de sistemas embebidos

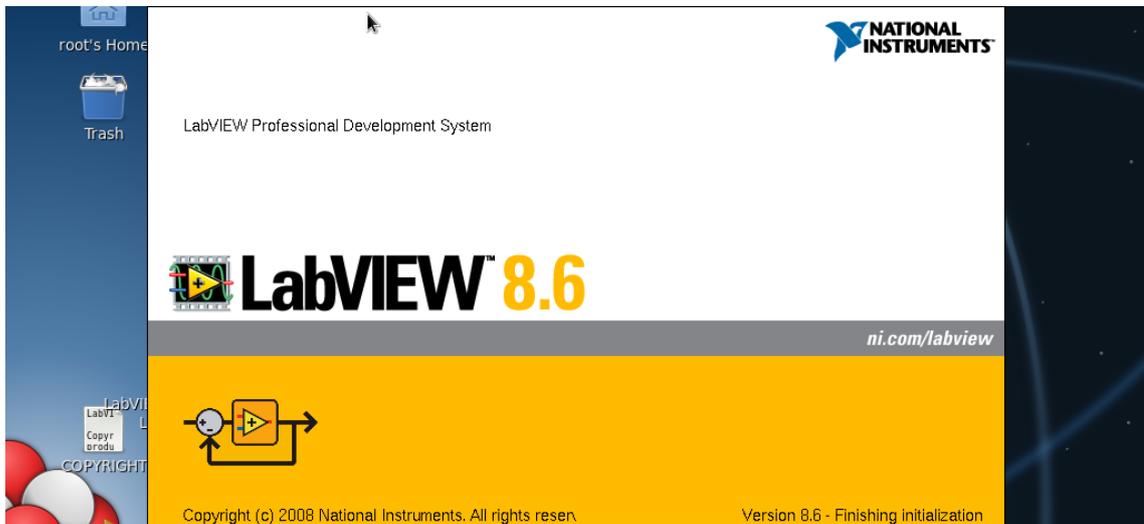


Figura 51-2. Software Labview inicio

Fuente: Scientific Linux 6.6.

Siguiendo el método Analítico, para que todos los componentes de esta investigación Científica funcionen de manera óptima como siguiente paso se construirá el módulo de distribución de materiales y una vez armado correctamente se procederá a la programación del sistema embebido.

2.2. Construcción del módulo de distribución de materiales

Procedemos al armado del módulo de distribución de materiales con las siguientes partes:

El Trole donde irán todas las partes que conforman el módulo de distribución, adicional a esto integramos el tablero que es la base donde mediante tuercas y tornillos se irán integrando todas las piezas, como podemos observar en la Figura 52-2.



Figura 52-2. Montaje trole, tablero, motor y banda transportadora

Realizado por: Avalos Edgar, 2015.

Como pieza de referencia para el módulo de distribución, se procede a colocar la base de la banda transportadora de 1 metro de largo, además del motor Bosch de 24 Voltios que se utilizara para que la banda transportadora pueda accionarse y transportar las piezas necesarias para la distribución de los materiales.



Figura 53-2. Módulo de distribución contenedores o magazines

Realizado por: Avalos Edgar, 2015.

Como siguiente paso al armando del módulo de distribución, instalamos los contenedores o magazines, notar que estos son 3 y deben colocarse cerca de la banda transportadora, se procede además al montaje del bus de datos Centronix el cual ayudara en la comunicación del PLC y el módulo de distribución

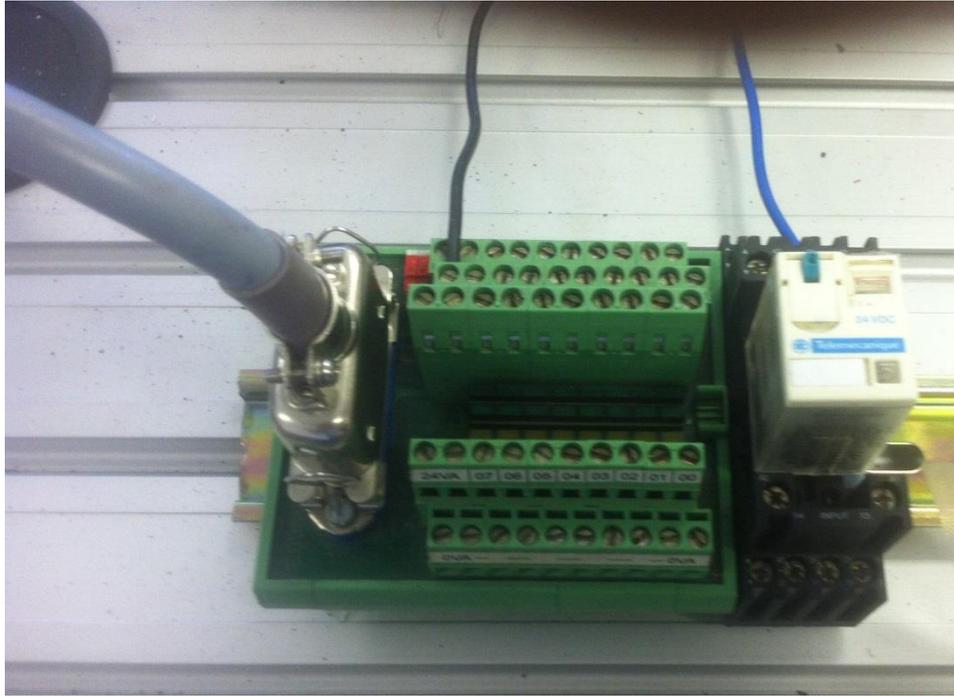


Figura 54-2. Módulo de Distribución Centronix bus de datos

Realizado por: Avalos Edgar, 2015.

Los cables deben ser colocados de manera consecutiva a las necesidades en la distribución de materiales, adicional a esto se necesita de Válvulas que utilicen los motores neumáticos necesarios para la distribución, estas válvulas son accionadas por aire comprimido, el cual se dirige hacia las válvulas YONGYI para luego ser distribuidas a los 3 cilindros neumáticos ubicados por debajo de las magazines o contenedores



Figura 55-2. Módulo de Distribución Válvulas YONGYI

Realizado por: Avalos Edgar, 2015.

Estas válvulas deben ser instaladas a una distancia no mayor a 50 cm de los cilindros neumáticos debido a que las mangueras azules transportan aire comprimido y accionan a los mismos no deben estar muy alejadas, dichas mangueras azules termina en los cilindros neumáticos, los cuales están conectados por medio del cable plomo a la tarjeta Centronix y a su vez al PLC el cual es controlado por el sistema embebido, esto se lo puede observar en la Figura 56-2.

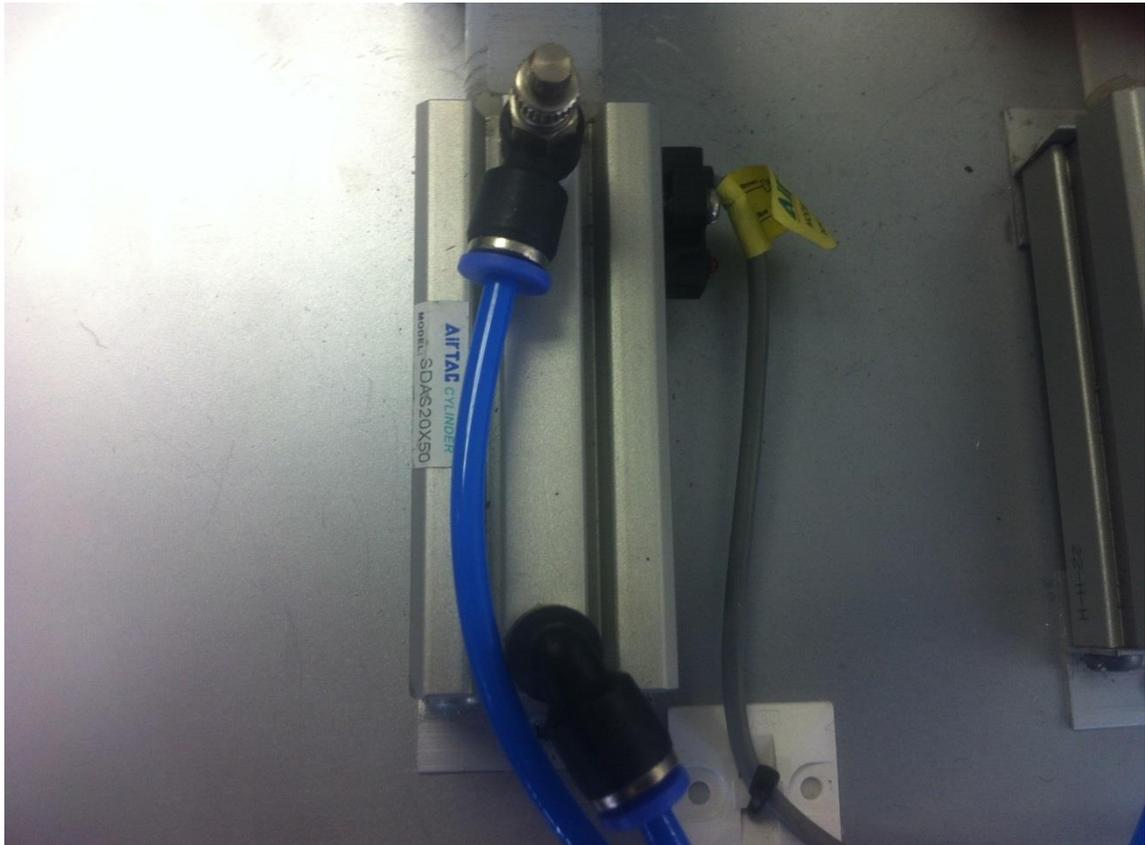


Figura 56-2. Módulo de Distribución Cilindros Neumáticos AIRTAC

Realizado por: Avalos Edgar, 2015.

Los Cilindros neumáticos utilizados para construir el módulo de distribución de materiales son tres y se encuentran en la parte posterior a los magazines, su función son empujar las piezas que emulan materiales, hacia la banda transportadora.



Figura 57-2. Módulo de Distribución Motor de 24 Voltios VOSH

Realizado por: Avalos Edgar, 2015.

La banda transportadora es accionada por un motor de 24 voltios marca VOSH este motor está sujeto a la base de la estación de distribución por tuercas y tornillos al igual que los demás componentes electrónicos este se encuentra conectado por medio de cables a la tarjeta CENTRONIX y esta a su vez por un conector al PLC.

En la figura 58-2 se puede observar el mecanismo que hace que funcione la banda transportadora, consiste en tres rodillos que se encuentran colocados de tal forma que la banda transportadora pueda mantenerse en continuo movimiento.



Figura 58.2. Banda transportadora montaje

Realizado por: Avalos Edgar, 2015.

Como siguiente punto en la construcción del módulo de distribución seguimos instalando las mangueras azules usando válvulas YONGYI usamos mangueras azules que son los conductores de aire comprimido fundamentales para que los cilindros neumáticos ejerzan movimiento.

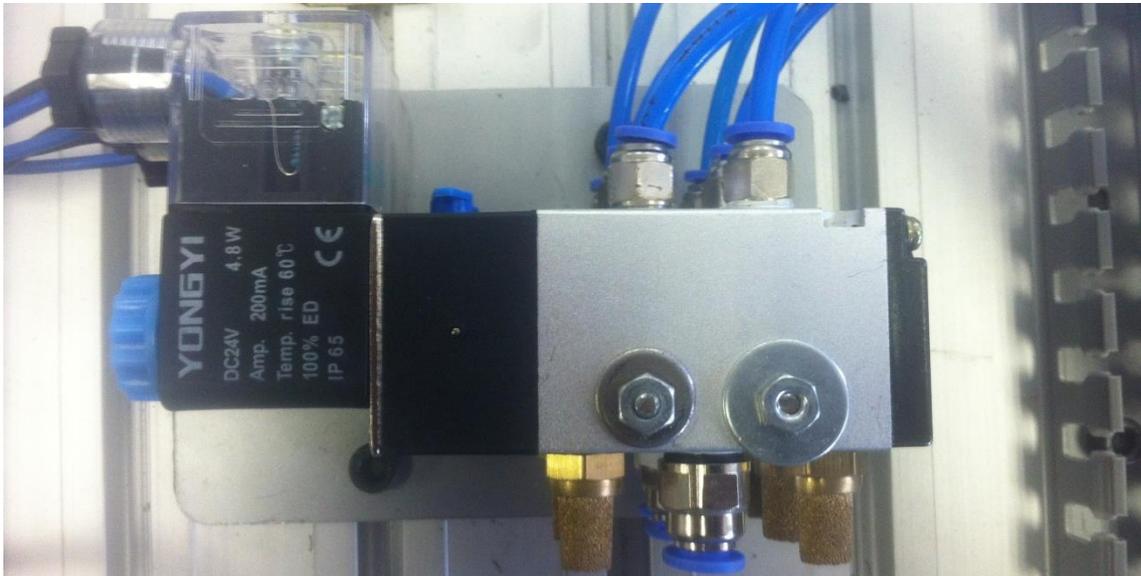


Figura 59.2. Válvulas YONGYI conexión mangueras

Realizado por: Avalos Edgar, 2015.

Se continúa con el cableado y la tarjeta CENTRONIX muy importante a la hora de realizar la comunicación entre el sistema embebido y el módulo de distribución esta tarjeta tiene comunicación directa con el PLC.

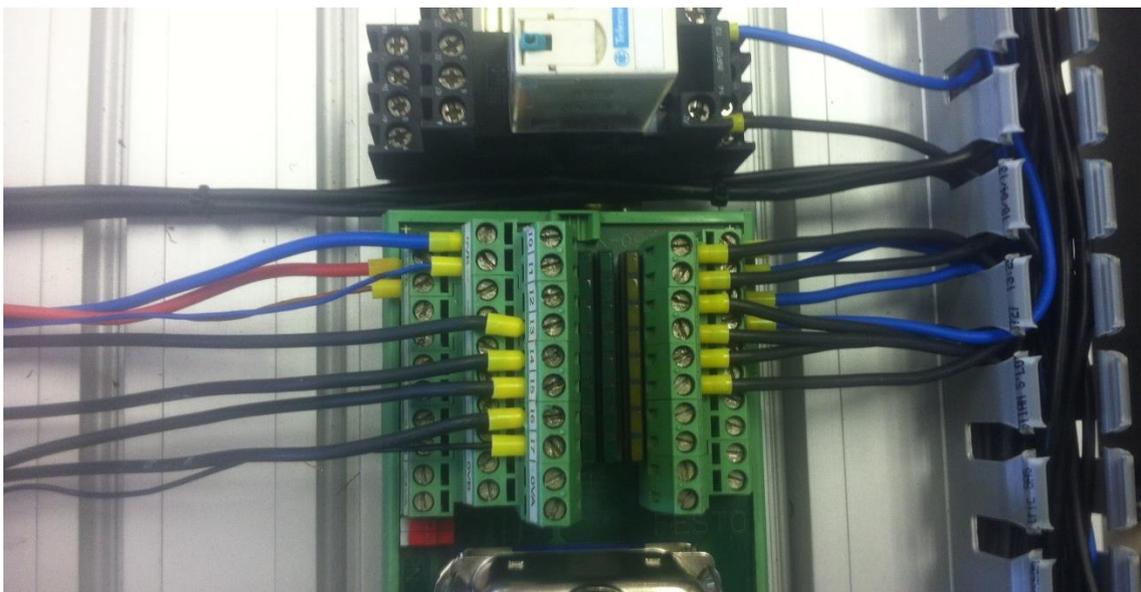


Figura 60-2. Cableado Centronix bus de datos

Realizado por: Avalos Edgar, 2015.

Para tener un control general de la estación de distribución de materiales con el sistema embebido se procede a instalar pulsadores de diferentes colores. Los cuales programaremos de acuerdo a las necesidades del módulo de distribución.



Figura 61.2. Módulo de distribución pulsadores

Realizado por: Avalos Edgar, 2015.

Ubicados todos los cilindros neumáticos en sus respectivos lugares podemos observar su posición final en el módulo de distribución, observamos que cada magazine o contenedor tiene en su parte inferior un cilindro neumático el cual mediante un cuadro rectangular inferior en cada magazine empuja las piezas de diferentes colores hacia la banda transportadora. Podemos apreciar además en la Figura 62-2 el montaje realizado ya de las mangueras azules que emiten aire comprimido, así como también el cableado necesario para que el PLC pueda funcionar mediante las órdenes de distribución que se haga en el sistema embebido.



Figura 62.2. Estación de distribución cableada

Realizado por: Avalos Edgar, 2015.

En la Figura 63-2 observamos otra toma de la estación de distribución y las conexiones de los diferentes dispositivos teniendo en claro que todos los componentes que intervienen están sujetadas a la base del módulo



Figura 63-2. Cableado PLC

Realizado por: Avalos Edgar, 2015.

Se sigue con el cableado del PLC y la tarjeta necesarios para la comunicación del sistema embebido, todo esta parte se encuentra implementada en un tablero adicional que fue creado con el fin de soporte del PLC empleado, como del cableado requerido para la comunicación del módulo de distribución de materiales.



Figura 64-2. Cableado tablero PLC

Realizado por: Avalos Edgar, 2015.

Terminado de emparejar todos los dispositivos correspondientes, la estación de distribución de materiales está a un 100% del armado total como siguiente paso realizaremos la implementación con el sistema embebido.

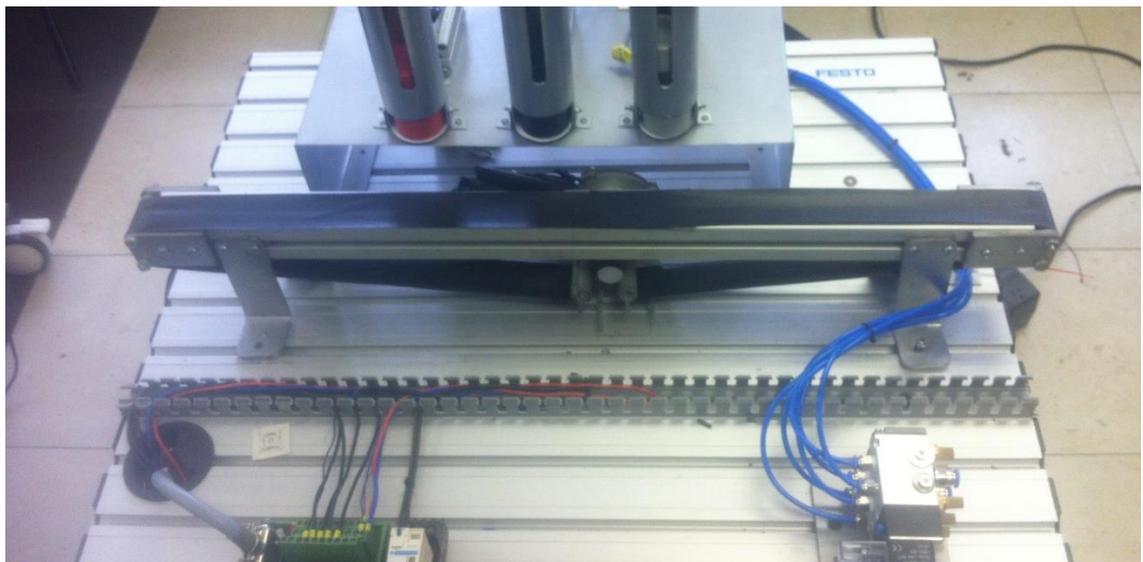


Figura 65-2. Módulo de Distribución de Materiales completo

Realizado por: Avalos Edgar, 2015.

2.2.1 Drivers de Comunicación.

Con la construcción total del módulo de distribución de materiales y la instalación del software para diseño de sistemas embebidos Labview se requiere realizar la conexión entre las dos partes mediante un PLC, el cual detallaremos su programación en Anexos.

Como primera dificultad encontrada al realizar esta conexión, encontramos que el sistema operativo Scientific Linux 6.6 utilizado para esta investigación científica no cuenta con Drivers de comunicación suficientes para realizar esta conexión debido a que las licencias necesarias solo se encuentran disponibles para sistemas operativos propietarios como Windows.

Por tanto se procede a la Instalación de Labview 2013 sobre Windows 8 asegurándonos de utilizar la versión completa.

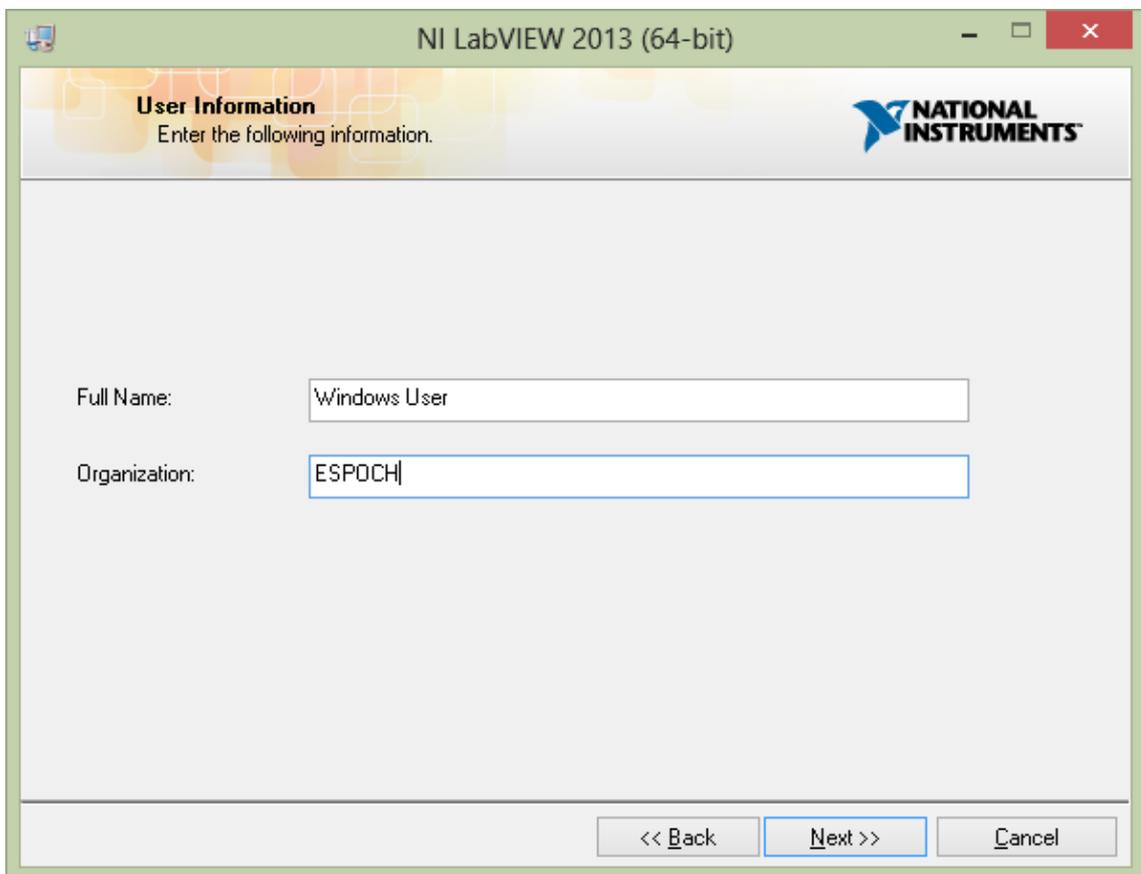


Figura 66-2. Instalación NI Labview 2013

Realizado por: Avalos Edgar, 2015.

Ejecutamos el instalador y como primera opción tenemos que ingresar la organización y el nombre de usuario. Colocamos nuestros datos y seleccionamos la opción siguiente o Next esto se detalla en la Figura 67-2.

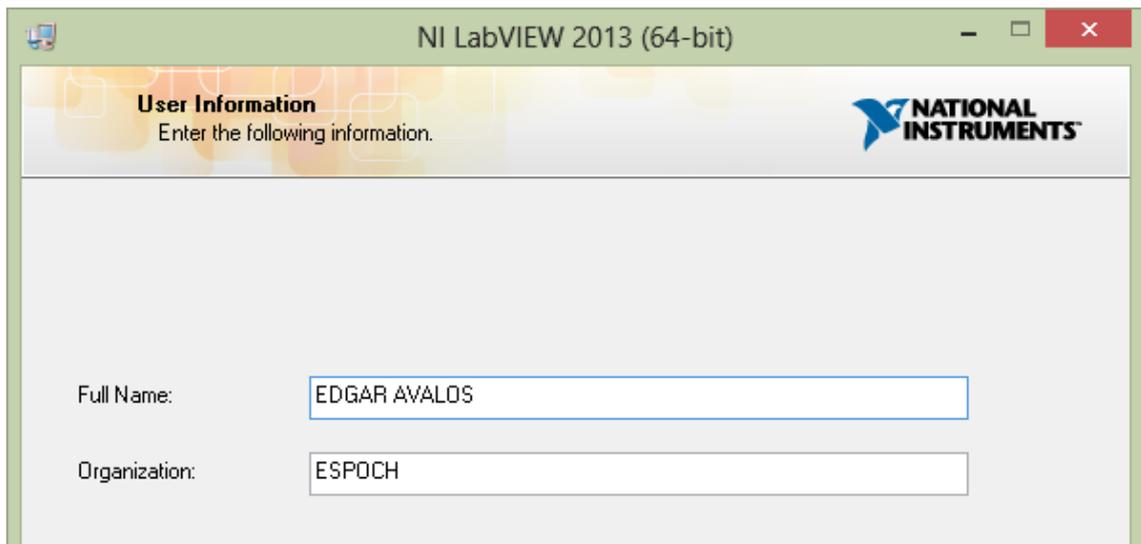


Figura 67-2. Instalación NI Labview Ingreso usuario y organización

Fuente: Labview 2013.

A continuación debemos colocar los números de seriales de las licencias las cuales debemos ingresarlas en los dos campos, como siguiente punto seleccionamos la tecla Next, para continuar con la instalación.



Figura 68-2. Instalación Labview Seriales

Fuente: Labview 2013.

Como siguiente punto de la instalación seleccionamos los directorios o carpetas de destino donde se almacenara el software, luego presionamos el botón Next para continuar con la instalación.

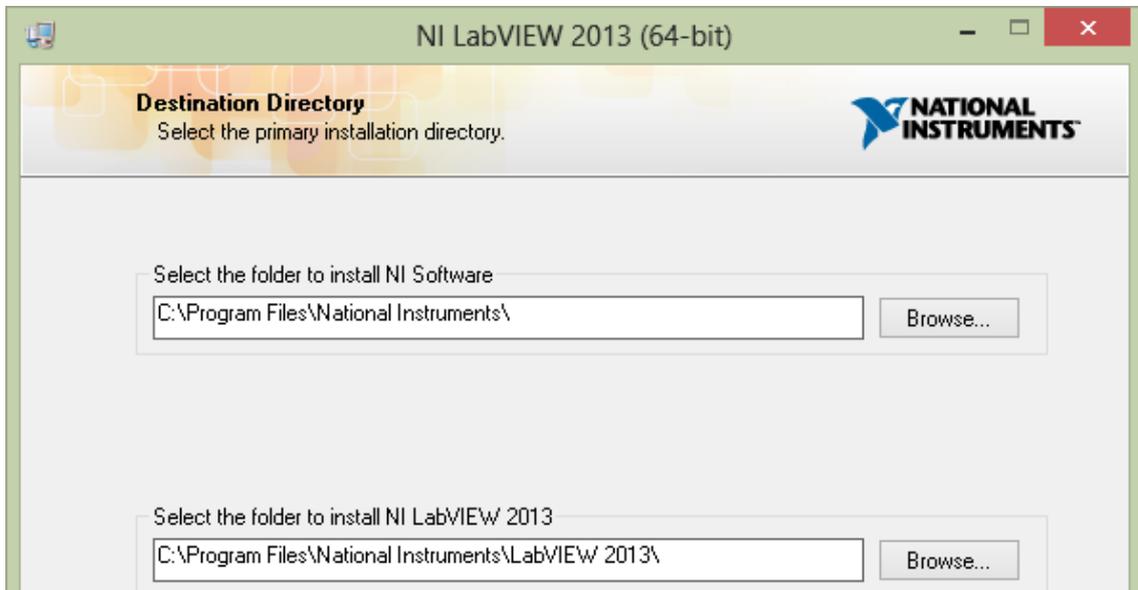


Figura 69-2. Instalación Ni Labview directorio de destino

Fuente: Labview 2013.

El siguiente paso en la instalación tenemos como opción seleccionar o no, si deseamos obtener nuevas actualización del software para diseño de sistemas embebidos Labview 2013 luego de esto presionamos el botón siguiente o Next. Y continuamos con la instalación.

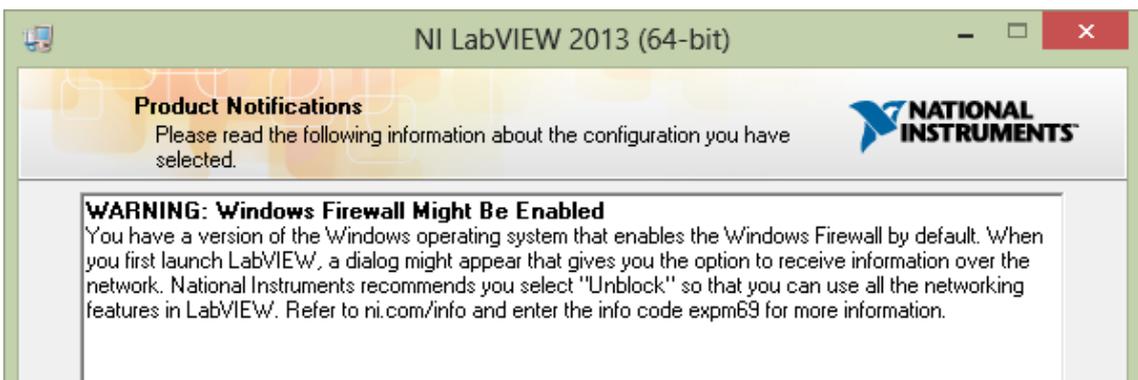


Figura 70-2. Instalación NI Labview Actualizaciones

Fuente: Labview 2013.

En la Figura 71-2 se pide seleccionar los términos de instalación y si aceptamos los mismos, luego de esto presionamos la opción Next como siguiente punto en la instalación.

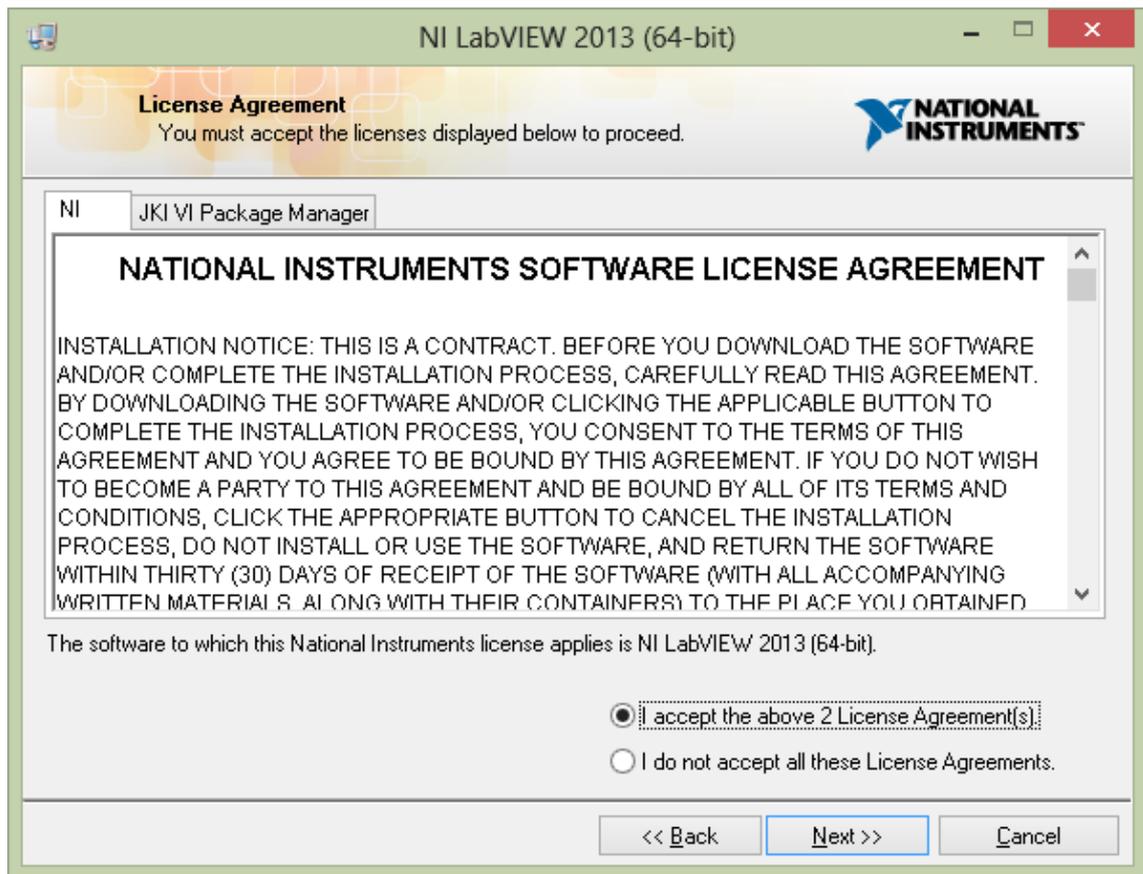


Figura 71-2. Instalación Ni Labview Licencias

Fuente: Labview 2013.

En el siguiente grafico se detallan todos los componentes que se van a instalar, para que este paso tenga efecto se selecciona la opción Next o siguiente.

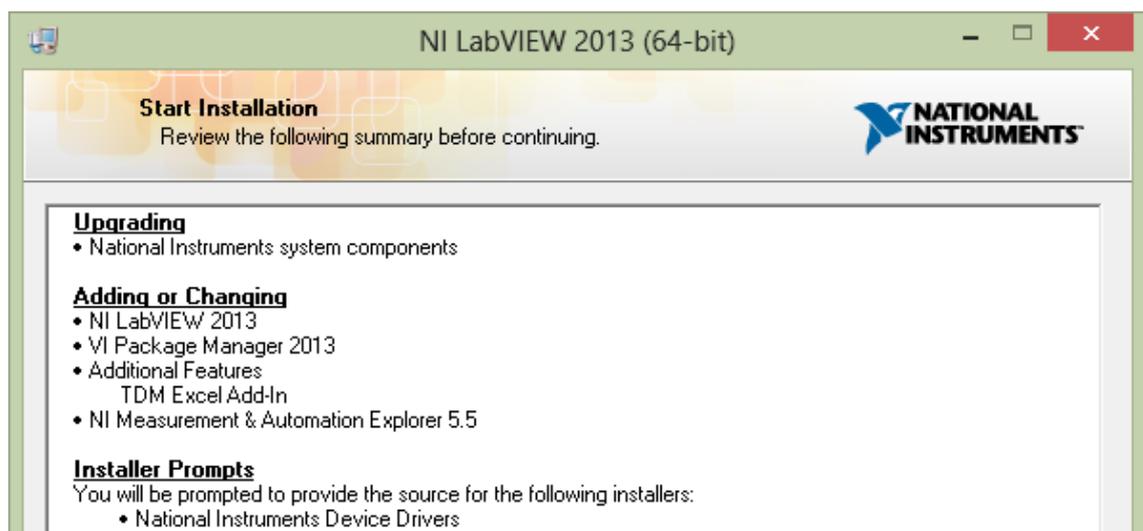


Figura 72-2. Instalación Ni Labview archivos

Fuente: Labview 2013.

Empieza la instalación en figura 73-2 con barras que indica el avance de la misma, hasta este punto de la instalación no se registra ningún inconveniente por compatibilidad del Software.

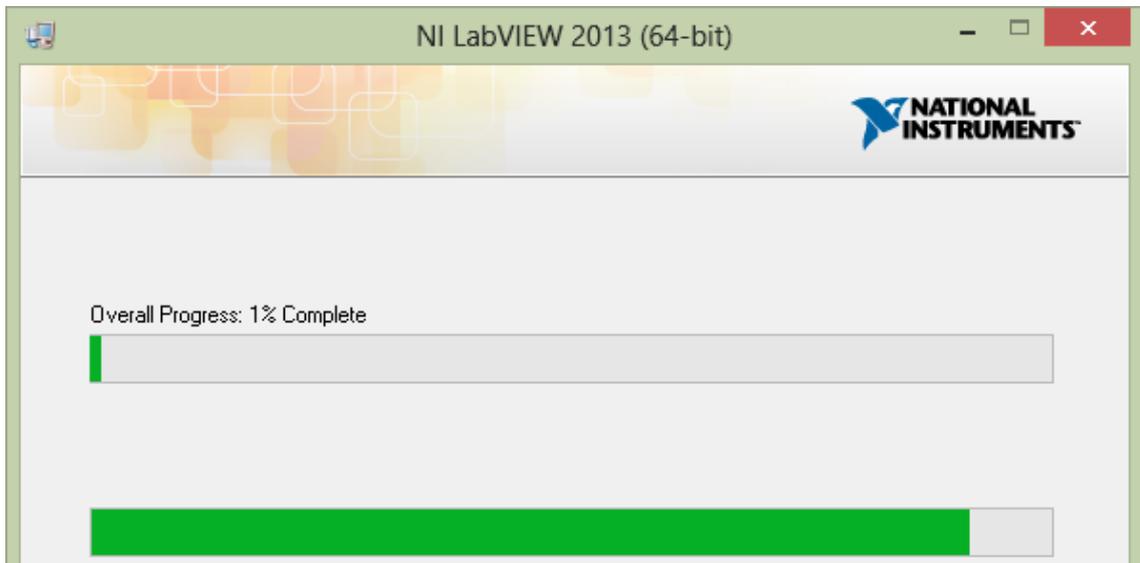


Figura 73-2. Instalación Ni Labview pantalla

Fuente: Labview 2013.

Se completa la instalación del Software para diseño de sistemas embebidos Labview Con el siguiente mensaje en la figura 74-2.



Figura 74-2. Instalación Labview Final

Fuente: Labview 2013.

2.2.2 Sistema embebido.

Se procede a detallar el sistema embebido necesario para controlar el módulo de distribución de materiales.

Procedemos a crear un nuevo proyecto usando Labview, seleccionamos archivo y nuevo proyecto, se despliegan dos pantallas el Panel Frontal Figura 75-2, 76-2 y el Diagrama de Bloques.

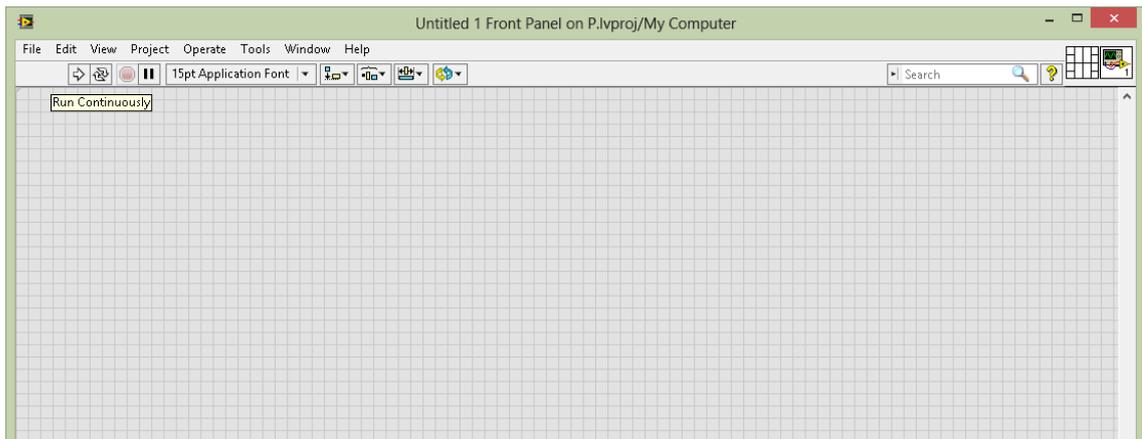


Figura 75-2. . Labview Nuevo Proyecto

Fuente: Labview 2013.

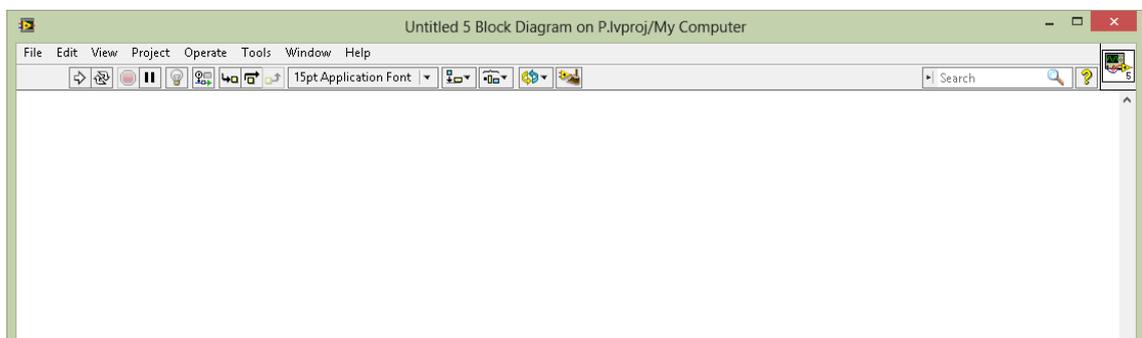


Figura 76-2. Labview Diagrama de Bloques

Fuente: Labview 2013.

Creamos controles de la estación original física y lo plasmamos en el panel frontal existen botones como:

Inicio: Ejecuta el Sistema Embebido se enciende la lámpara verde

Paro: Detiene el Sistema Embebido se enciende la lámpara roja

Accionamos el botón Cilindro o Contenedor A: para empujar una pieza Roja

Accionamos el botón Cilindro o Contenedor B: para empujar una pieza Negra

Accionamos el botón Cilindro o Contenedor C: para empujar una pieza Ploma o Plateada

Botón Motor: Controla la banda transportadora

Botón Manual: Utilizamos los botones físicos de la Estación de distribución, para ejecutar todo

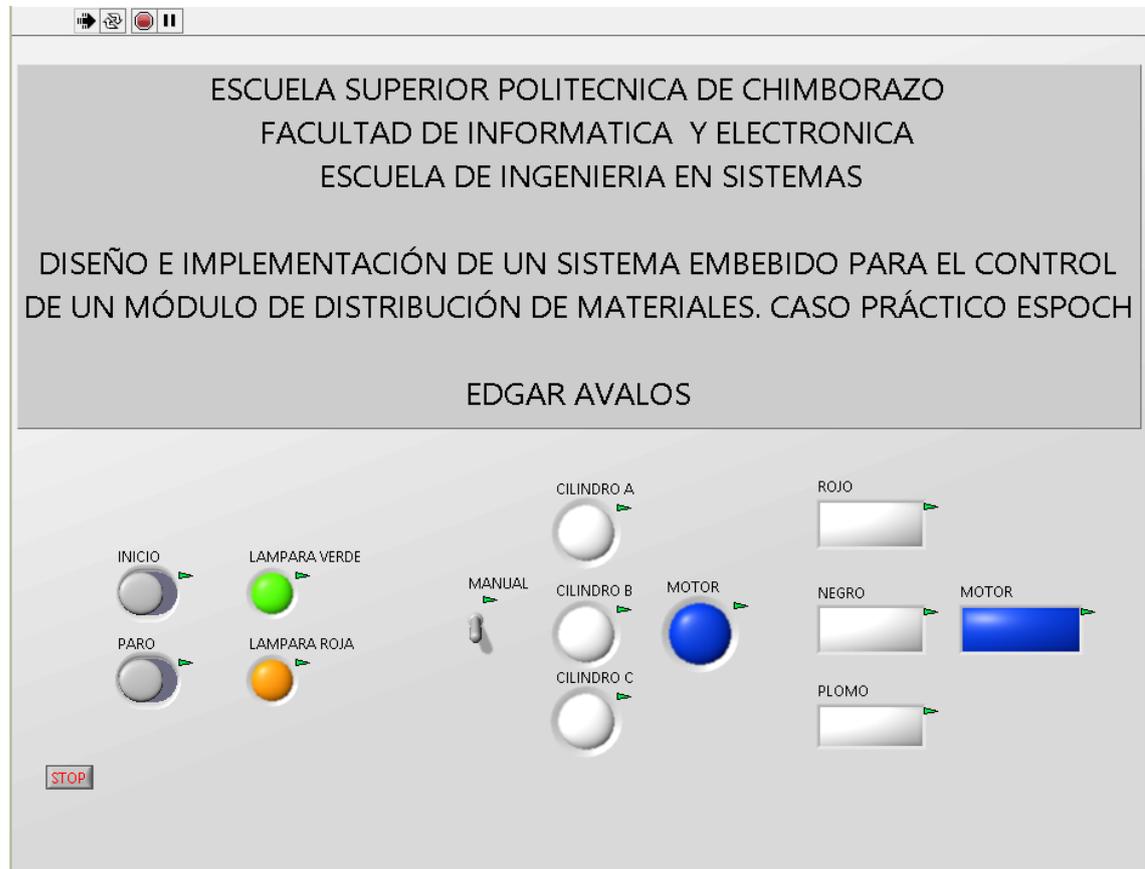
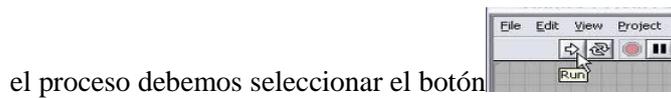


Figura 77-2. Sistema Embebido Panel Frontal

Realizado por: Avalos Edgar, 2015.

El diagrama de bloques tiene relación directa con el panel frontal que es la interfaz donde arrastramos todos los componentes necesarios para construir el sistema embebido, en cambio en el diagrama de bloques detallaremos las funciones específicas de cada componente.

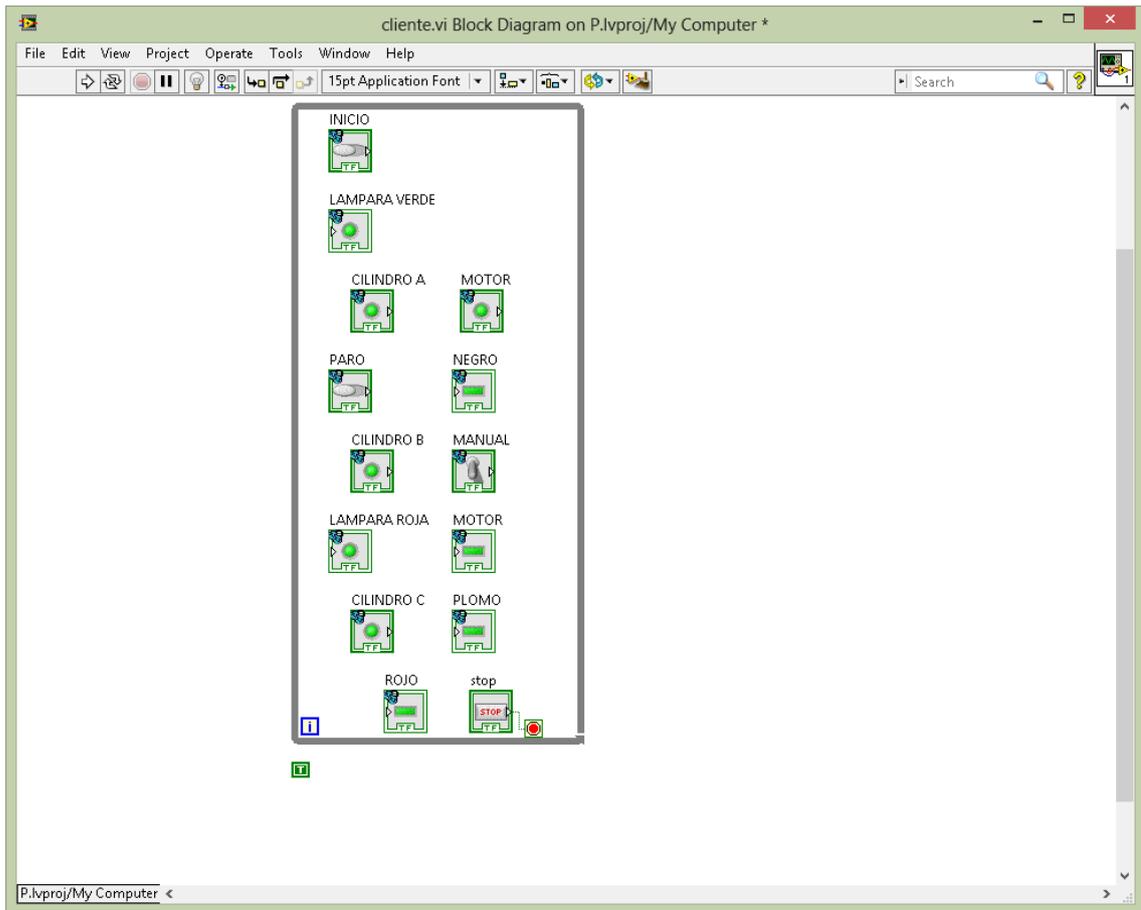


Figura 78.2. Diagrama de Bloques Sistema Embebido

Realizado por: Avalos Edgar, 2015.

Creado el sistema embebido, necesitamos conectarlo a la estación de distribución de materiales este proceso se lo realiza programando el PLC (Siemens SIMATIC S7-1200 los bloques de programación se encuentran en Anexos), utilizado en esta investigación científica.

Para este paso necesitamos el software compatible con Labview de nombre **NI OPC Server** el cual puede comunicarse con cualquier controlador lógico programable o **PLC**, OPC proviene de (OLE for Process Control) de donde las siglas en ingles OLE significan Object Linking and Embedding, de donde la traducción al español más cercana sería Objeto Enlazado y Embebido para el control de procesos.

Esto define el estándar para comunicar datos en tiempo real entre los dispositivos de control del sistema embebido y el módulo de distribución de materiales. Los servidores OPC están disponibles virtualmente para todos los PLC del Mercado en la siguiente figura se detalla su funcionamiento y la forma de conexión entre un PLC y el software desarrollado en Labview.

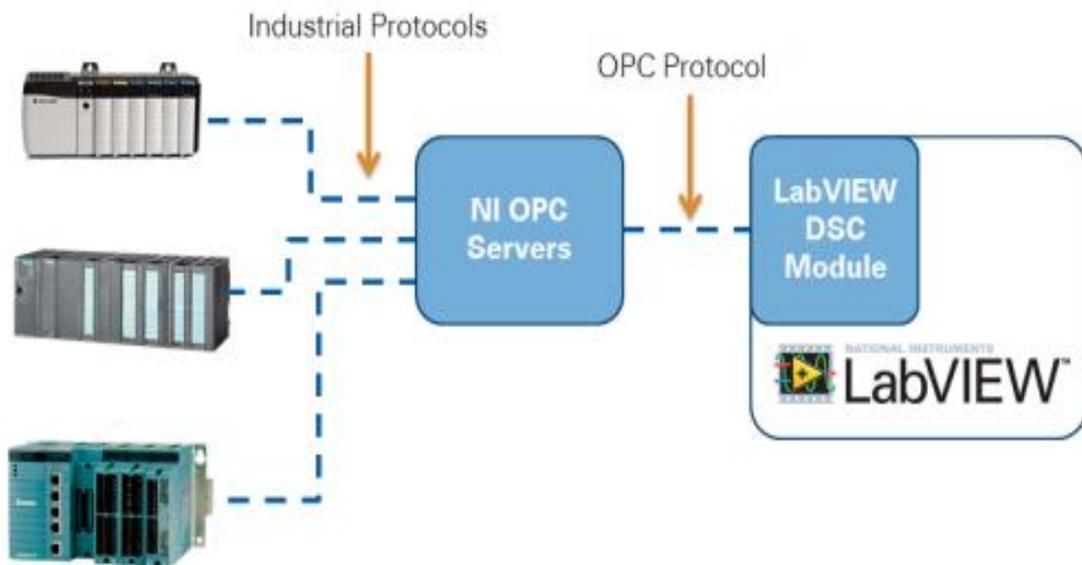


Figura 79.2. NI OPC

Fuente: Labview.

Para la conexión entre el sistema embebido y el PLC creamos un archivo en NI OPC Server declarando las siguientes variables necesarias para la comunicación, utilizamos los siguientes pasos:

1. Ingresamos a NI OPC Servers seleccionando Start » Programs » National Instruments » NI OPC Servers » NI OPC Servers. Con NI OPC Servers a continuación creamos, configuramos etiquetas que se asocian al PLC.

2. NI OPC Servers debe abrirse con la simulación del sistema embebido anteriormente creado y configurado en NI OPC Servers. Luego de crear todas las etiquetas necesarias las visualizaremos de la siguiente forma.

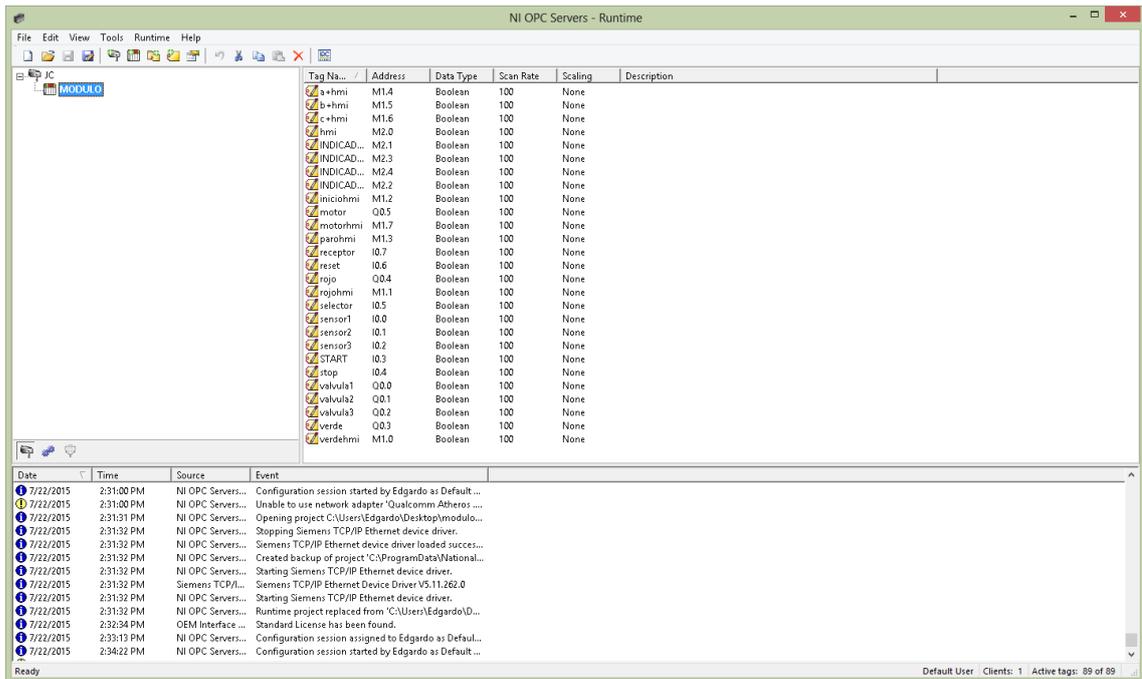


Figura 80-2. NI OPC Runtime

Realizado por: Avalos Edgar, 2015.

A continuación en. En NI OPC Servers, seleccionamos **Tools » Launch OPC Quick Client** para ingresar al OPC Quick Client, el cual se usa para ver los datos de las etiquetas del OPC.

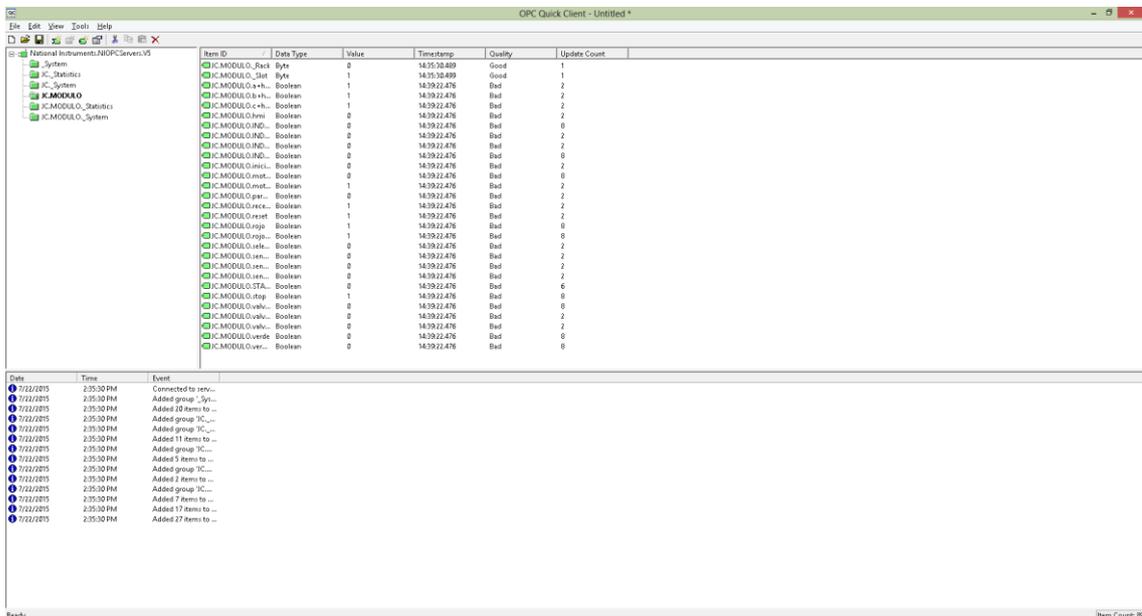


Figura 81-2. OPC Quick Client

Realizado por: Avalos Edgar, 2015.

Realizados todas las conexiones anteriores se procede a la etapa de pruebas en la cuales cumplen todos los requerimientos planteados en esta investigación científica. Tomando el módulo de distribución de materiales de 1 contenedor y el módulo de 3 contenedores.

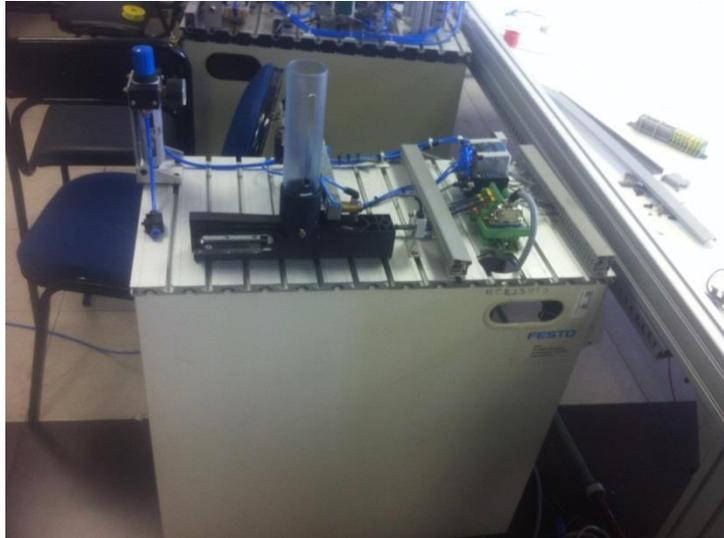


Figura 82-2. Módulo de Distribución 1 contenedor

Realizado por: Avalos Edgar, 2015.

Los resultados obtenidos en esta etapa de pruebas se los detalla en el capítulo III señalando las principales diferencias entre cada módulo de distribución.

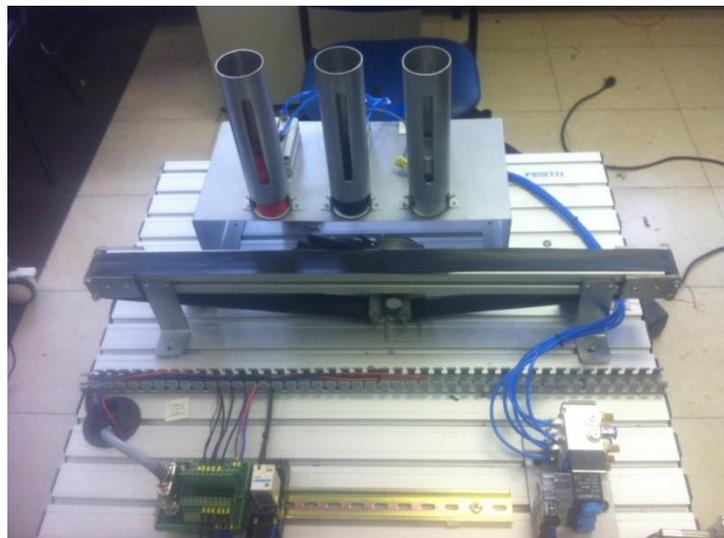


Figura 83-2. Módulo de Distribución 3 contenedores

Realizado por: Avalos Edgar, 2015.

CAPÍTULO III

3. MARCO DE RESULTADOS, DISCUSIÓN Y ANÁLISIS

3.1 Resultados y Analisis

En este capítulo se detalla los resultados obtenidos a partir de la etapa de pruebas entre los dos módulos de distribución de materiales.

3.1.1 Definición de parámetros

Para evaluar el rendimiento entre el módulo de distribución original y el implementado en esta investigación científica, utilizamos los siguientes parámetros.

Tabla 2-3. Definición de Variables

INDICADOR	DESCRIPCIÓN
Distribución de materiales	Este indicador nos permitirá medir en porcentajes la distribución de materiales
Velocidad Piezas	Se medirá la velocidad de las piezas al atravesar la banda transportadora.
Color Piezas	Ayudará a medir en porcentajes la distribución de materiales por color

Realizado por: Avalos Edgar, 2015.

Dentro de cada uno de los parámetros antes mencionados hemos visto la necesidad de tener índices, para obtener mejores resultados los cuales se enumeran a continuación.

Tabla 3-3. Índice de los parámetros

PARÁMETRO	ÍNDICE
Distribución de materiales	Banda transportadora %
Velocidad Piezas	En metros por segundo m/s
Color Piezas	Banda transportadora %

Realizado por: Avalos Edgar, 2015.

3.1.2 Resultados individuales de los parámetros.

Para determinar la estación de distribución de mejor rendimiento se necesita herramientas que apoyan a la verificación, con la finalidad de realización de esta tarea se seleccionó la herramienta para diseño de sistemas embebidos Labview 2013 la cual mediante su interfaz de monitoreo controla todo el proceso.

Tabla 4-3. Definición de valorización

Rangos	Valoración Cualitativo	Valoración Cuantitativo	Valor representativo
>89 %	Excelente	4	
>59% y <90%	Bueno	3	
>29% y <60%	Malo	2	

Realizado por: Avalos Edgar, 2015.

Distribución de Materiales

Luego de recopilar la información en la etapa de pruebas de la distribución de materiales por número piezas, 12, 8, 4, 1 y la tabla número III se muestran los resultados obtenidos.

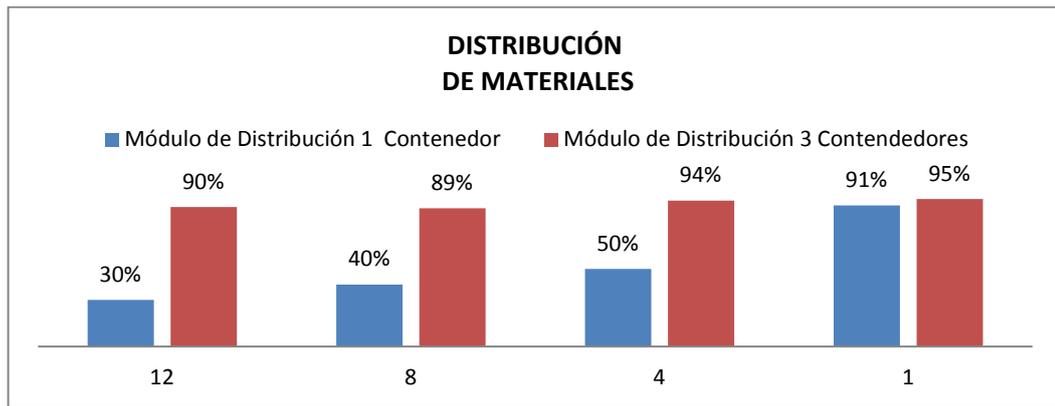
Tabla 5-3. Conglomerado de datos

	DISTRIBUCION DE MATERIALES							
	Módulo de Distribución 1 Contenedor				Módulo de Distribución 3 Contenedores			
	12	8	4	1	12	8	4	1
Parámetros								
Banda transportadora%	2	2	2	4	4	3	4	4

Realizado por: Avalos Edgar, 2015.

En la siguiente ilustración se puede visualizar la diferencia entre el módulo de distribución de materiales con 1 contenedor y el módulo de distribución de materiales de 3 contenedores en la distribución de 12,8,4,1 piezas.

Tabla 6-3. Estadístico del Indicador Distribución de Materiales



Realizado por: Avalos Edgar, 2015.

Velocidad piezas

Luego de recopilar la información en la etapa de pruebas de la velocidad de las piezas en la banda transportadora realizada por número piezas, 12, 8, 4, 1 y la tabla número III se muestran los resultados obtenidos.

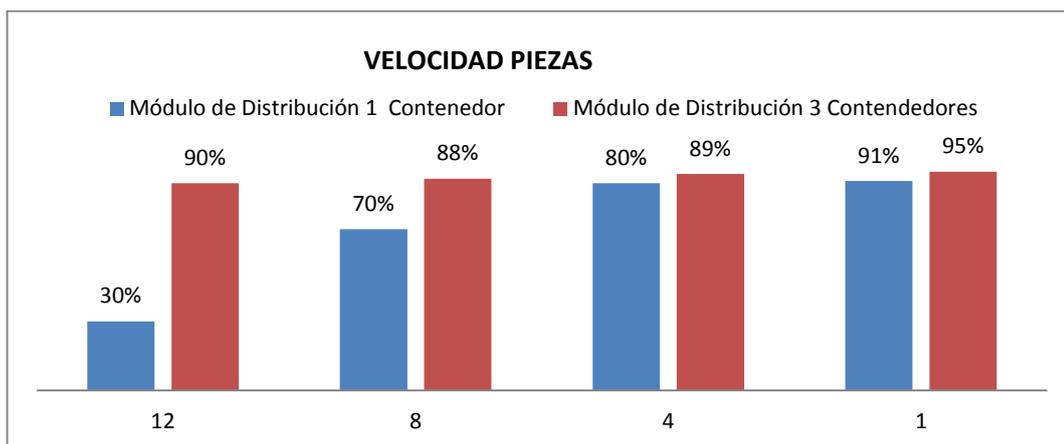
Tabla 7-3. Conglomerado de datos

Parámetros	VELOCIDAD PIEZAS							
	Módulo de Distribución 1 Contenedor				Módulo de Distribución 3 Contenedores			
	12	8	4	1	12	8	4	1
Velocidad en m/s	2	3	4	4	4	3	3	4

Realizado por: Avalos Edgar, 2015.

En la siguiente ilustración se puede visualizar la diferencia entre el módulo de distribución de materiales de 1 contenedor y el módulo de distribución de materiales de 3 contenedores y la velocidad al distribuir 12,8,4,1 piezas por la banda transportadora.

Tabla 8-3. Estadístico del Indicador Velocidad Piezas



Realizado por: Avalos Edgar, 2015.

Color Piezas

Luego de recopilar la información del Número de peticiones ejecutadas y el Tiempo de respuesta al pedir piezas por determinado color, en las pruebas de 12, 8, 4,1 piezas y la tabla número III muestra los resultados obtenidos.

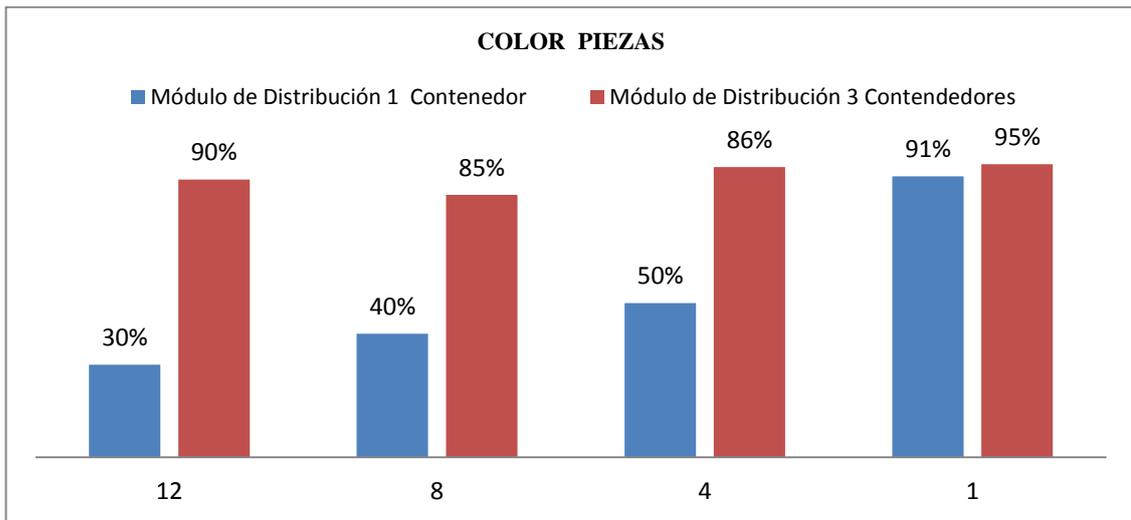
Tabla 9-3. Conglomerado de datos Color Piezas

	COLOR PIEZAS							
	Módulo de Distribución 1 Contenedor				Módulo de Distribución 3 Contenedores			
Parámetros	12	8	4	1	12	8	4	1
Color Piezas	2	2	2	4	4	3	3	4

Realizado por: Avalos Edgar, 2015.

En la siguiente ilustración se puede visualizar la diferencia entre el módulo de distribución de materiales de 1 contenedor y el módulo de distribución de materiales de 3 contenedores y la velocidad al distribuir piezas por color 12,8,4,1 piezas por la banda transportadora.

Tabla 10-3. Estadístico del Indicador Color Piezas



Realizado por: Avalos Edgar, 2015.

Hipótesis

La implementación de un sistema embebido para el control de un módulo de distribución de materiales, para el sistema de producción modular optimizara el proceso de manejo de materiales del sistema de montaje.

Determinación de la Hipótesis

Después de haber realizado el análisis de cada variable con su respectivo indicador se llega a la conclusión que al implementar un sistema embebido para el control del módulo de distribución de materiales con 3 contenedores de esta investigación científica es superior al módulo de distribución de materiales original de un solo contenedor, a continuación tenemos cada uno de los valores en forma general de cada uno de los parámetros, la siguiente tabla visualizara cada parámetro con sus respectivos valores en números y en porcentajes.

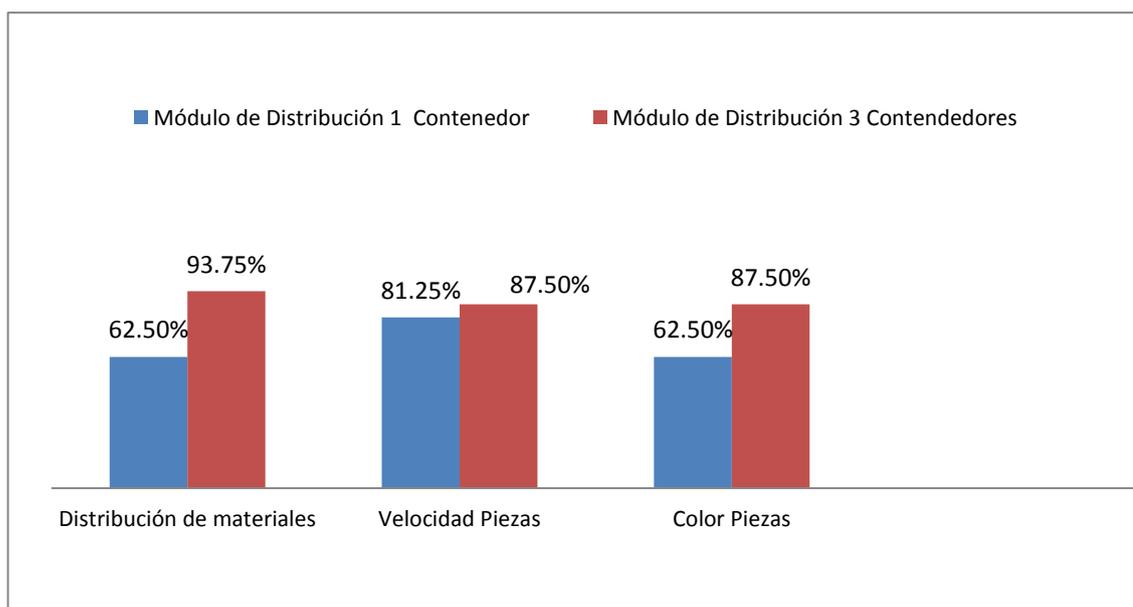
Tabla 11-3. Conglomerado de datos total

Parámetro e Indicadores		Módulo de Distribución 1 Contenedor	Módulo de Distribución 3 Contenedores	Calificación Máxima	% Módulo de Distribución 1 Contenedor	% Módulo de Distribución 3 Contenedores
Distribución de materiales	Banda transportadora%	10	15	16	62.5%	93.75%
Velocidad Piezas	En metros por segundo m/s	13	14	16	81.25%	87.50%
Color Piezas	Banda transportadora %	10	14	16	62.50%	87.50%

Realizado por: Avalos Edgar, 2015.

En la siguiente ilustración se puede visualizar la diferencia entre el módulo de distribución de materiales de 1 contenedor y el módulo de distribución de materiales de 3 contenedores

Tabla 12-3. Resultados Globales



Realizado por: Avalos Edgar, 2015.

En la Tabla 12-3, los resultados globales obtenidos durante las diferentes pruebas realizadas demuestran que el módulo de distribución de materiales de 3 contenedores es más eficiente que el módulo de distribución de materiales de 1 contenedor superándolo con un promedio de un 31.25% en la Distribución de materiales, superándolo con un 6.25% en la velocidad de piezas por distribución, superándolo además con un 25% en la distribución por piezas de color concluyendo con un calificativo de excelente para el módulo de distribución de materiales de 3 contenedores y de bueno para el módulo de distribución de 1 contenedor.

CONCLUSIONES

- La implementación de un sistema embebido para el control de un módulo de distribución de materiales permitió determinar las características, ventajas y beneficios que tiene el módulo de distribución de 3 contenedores, el cual obtuvo un porcentaje del 89,58% sobre el módulo de distribución de un solo contenedor, que obtuvo un porcentaje del 68,75%; dados los resultados se dice que el módulo de distribución de materiales que mejor responde a las pruebas realizadas es el de tres contenedores, denominándole como el más eficiente en rendimiento.
- Los indicadores determinados para medir el rendimiento entre los dos módulos de distribución de materiales, contribuyeron a la verificación clara y concisa de cada uno de ellos, obteniendo los siguientes resultados: para la distribución de materiales con el módulo de tres contenedores 93,75% y el módulo de 1 contenedor 62,50%, para la velocidad de las piezas en la distribución del módulo de 3 contenedores 87.50% y el módulo de 1 contenedor 81,25%, para la distribución por color con el módulo de 3 contenedores 87.50% y el módulo de 1 contenedor 62,50%, obteniendo como resultado final que el módulo de tres contenedores es mejor para realizar la distribución de materiales.
- El sistema embebido implementado en esta investigación científica fue desarrollado y enfocado específicamente para que la distribución de materiales sea más rápida y eficiente.
- El módulo de distribución de 3 contenedores es el mejor para la distribución de materiales el cual fue implementado en el Laboratorio de procesos de la FIE

RECOMENDACIONES

- Se recomienda al director del Laboratorio de procesos de la Facultad de Informática y Electrónica utilizar sistemas operativos propietarios para la implementación de sistemas embebidos, ya que los sistemas operativos libres utilizados en esta investigación científica presentan problemas en drivers de comunicación.
- Estimular a los estudiantes en el estudio de la programación de dispositivos como PLC'S ya que estos son de mucha ayuda en la automatización industrial, automatizar procesos y facilitar la producción y distribución de materiales
- Investigar diferentes tipos de comunicación entre el módulo de distribución de materiales y el sistema embebido.
- Utilizar software libre para desarrollar investigaciones científicas similares, para evitar problemas en el uso de licencias y el pago de las mismas.
- La implementación de un sistema embebido para el control de un módulo de distribución de materiales, es una puerta abierta hacia la fabricación y procesamiento de productos en una sociedad, se recomienda a los estudiantes tomar la posta y seguir investigando el cómo mejorar estos procedimientos para algún día optimizar la matriz productiva de nuestro país.

BIBLIOGRAFÍA

- **Bneum.** *Actuadores Neumáticos* [blog]. [Consulta: 02-05-2015] Disponible en: <http://bneum.blogspot.com/2012/09/22-calculo-de-actuadores-neumaticos.html>
- **MARTINEZ, I.** *Célula De Fabricación Flexible* [en línea]. [Consulta: 08-07-2014] Disponible en: http://www.lhusurbil.eus/irjlmartinez/MANUALES/E3-PROCESO/E3-Proceso_Omron.pdf
- ***Diseño De Sistemas Embebidos*** [en línea]. [Consulta: 13-11-2013] Disponible en : <https://www.ni.com/academic/esa/embedded.htm>
- ***Estación Cinta Transportadora*** [en línea]. [Consulta: 03-05-2013] Disponible en: <http://www.festo-didactic.com/mx-es/learning-systems/tecnica-para-escuelas-de-ensenanza-general/complete-package/estacion-cinta-transportadora.htm?fbid=bXguZXMuNTY0LjE0LjE4LjExMTkuNTQxMA>
- ***Estación de distribución.*** [en línea]. [Consulta: 22-06-2014] Disponible en: <http://www.festo-didactic.com/mx-es/learning-systems/mps-sistema-de-produccion-modular/estaciones/estacion-de-distribucion-mps-para-empezar.htm?fbid=bXguZXMuNTY0LjE0LjE4LjYwNi4zOTQz>
- ***Herramientas de diseño para sistemas embebidos.*** [en línea]. [Consulta: 17-11-2013] Disponible en: <http://es.slideshare.net/luisftube/herramientas-de-diseo-para-sistemas-embebidos>
- ***Implementación de sistemas embebidos.*** [en línea]. [Consulta: 12-11-2013] Disponible en: http://www.sase.com.ar/2012/files/2011/11/case2012_memorias_pp109-234.pdf
- **Jiménez, N.** *Diseño e implementación de un sistema embebido para la adquisición y transmisión de señales biomédicas a través de la red celular.*, Tesis Pregrado., Universidad Nacional de Colombia., Facultad de Ingeniería., Departamento de Ingeniería de Sistemas e Industrial., Bogotá, Colombia., 2011., Pp. 36-53.[Consulta: 18-05-2015]. Disponible en: http://www.bdigital.unal.edu.co/4234/1/299696.2011_pte_1.pdf

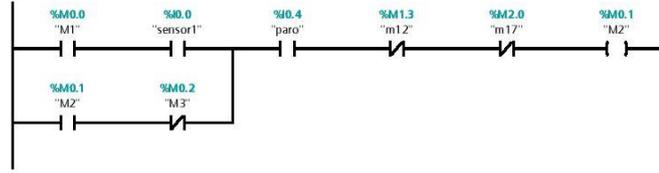
- **Kharsansky, A.** *Diseño e implementación de un sistema embebido de control de actitud para aeronaves no tripuladas.*, Tesis Pregrado, Universidad de Buenos Aires., Facultad de Ingeniería., Buenos Aires, Argentina., 2013., Pp. 2.[Consulta: 13-05-2015].
Disponible en:
<http://laboratorios.fi.uba.ar/lse/tesis/LSE-FIUBA-Tesis-Grado-Alan-Kharsansky-2013.pdf>
- **Labview Run-Time Engine 2013 - (Linux) - Linux, Redhat, Suse, Scientific Linux.** [en línea]. [Consulta: 02-03-2015] Disponible en:
<http://www.ni.com/download/labview-run-time-engine-2013/4062/en/>
- **Labview 2013.** [en línea]. [Consulta: 22-02-2014] Disponible en:
<http://www.ni.com/newsroom/release/labview-2013-all-systems-go/es/>
- **Linux y el sistema Gnu.** [en línea]. [Consulta: 20-12-2013] Disponible en:
<http://www.gnu.org/gnu/linux-and-gnu.es.html>
- **Módulos de Derivación Frm/Bloques De Distribución Frz, Serie D.** [en línea]. [Consulta: 10-09-2014] Disponible en:
https://www.festo.com/cat/en-gb_gb/data/doc_ES/PDF/ES/FRM-D_ES.PDF
- **Motores y Controladores** [en línea]. [Consulta: 19-12-2014] Disponible en:
http://www.festo.com/net/SupportPortal/Files/17216/Motors_and_controllers_ES.pdf
- **Oracle Vm Virtualbox.** [en línea]. [Consulta: 04-06-2014] Disponible en:
<http://www.oracle.com/technetwork/es/server-storage/virtualbox/downloads/index.html>
- **Programación de Controladores Lógicos.** [en línea]. [Consulta: 25-05-2015]
Disponible en:
[http://upload.wikimedia.org/wikipedia/commons/6/65/Programacion_de_controladores_logicos_\(PLC\).pdf](http://upload.wikimedia.org/wikipedia/commons/6/65/Programacion_de_controladores_logicos_(PLC).pdf)
- **Scientific Linux.** [en línea]. [Consulta: 10-01-2014] Disponible en:
<https://www.scientificlinux.org/>

- ***Sistemas embebidos basados en fpgas para instrumentación.*** [en línea]. [Consulta: 15-11-2013] Disponible en:
<http://ocw.uc3m.es/tecnologia-electronica/sistemas-embebidos-basados-en-fpgas-para-instrumentacion/material-de-clase-1/introduccion-a-sistemas-de-instrumentacion-embebidos>

ANEXOS

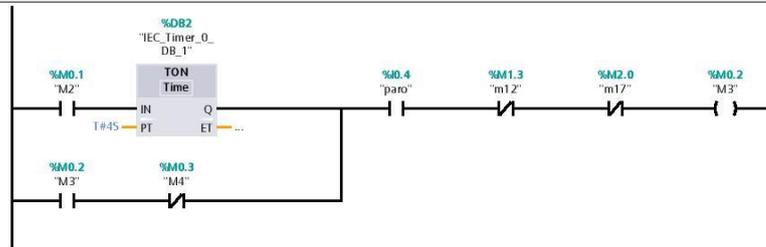
Anexo A. Programación del PLC

Totally Integrated Automation Portal			
PROGRAMA PLC / PLC_1 [CPU 1212C AC/DC/Rly] / Bloques de programa			
Main [OB1]			
Main Propiedades			
General			
Nombre	Main	Número	1
Idioma	KOP	Tipo	OB.ProgramCycle
Información			
Título	"Main Program Sweep (Cycle)"	Autor	
Familia		Versión	0.1
		ID personalizada	
Nombre			
Temp		Tipo de datos	
Segmento 1:			
Símbolo			
"marcha"	%I0.3	Tipo	Bool
"paro"	%I0.4	Tipo	Bool
"llave"	%I0.5	Tipo	Bool
"M1"	%M0.0	Tipo	Bool
"M2"	%M0.1	Tipo	Bool
"M6"	%M0.5	Tipo	Bool
"IEC_Timer_0_DB"	%DB1	Tipo	IEC_Timer
T#65	T#65	Tipo	Time
"m11"	%M1.2	Tipo	Bool
"m12"	%M1.3	Tipo	Bool
"m17"	%M2.0	Tipo	Bool
Segmento 2:			



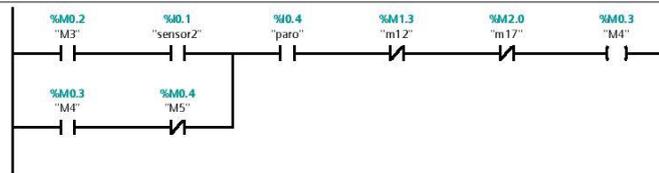
Símbolo	Dirección	Tipo	Comentario
"paro"	%I0.4	Bool	
"M1"	%M0.0	Bool	
"M2"	%M0.1	Bool	
"sensor1"	%I0.0	Bool	
"M3"	%M0.2	Bool	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	

Segmento 3:



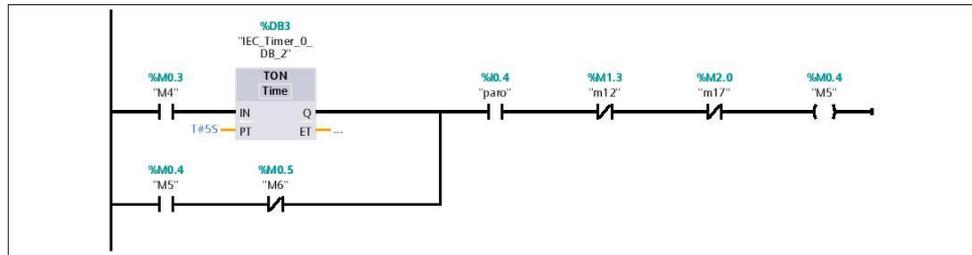
Símbolo	Dirección	Tipo	Comentario
"paro"	%I0.4	Bool	
"M2"	%M0.1	Bool	
"M3"	%M0.2	Bool	
T#4S	T#4S	Time	
"M4"	%M0.3	Bool	
"IEC_Timer_0_DB_1"	%DB2	IEC_Timer	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	

Segmento 4:



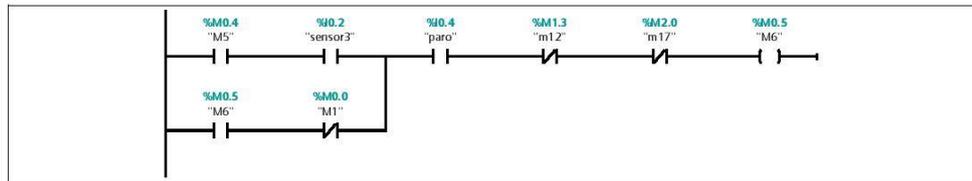
Símbolo	Dirección	Tipo	Comentario
"paro"	%I0.4	Bool	
"M3"	%M0.2	Bool	
"M4"	%M0.3	Bool	
"sensor2"	%I0.1	Bool	
"M5"	%M0.4	Bool	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	

Segmento 5:



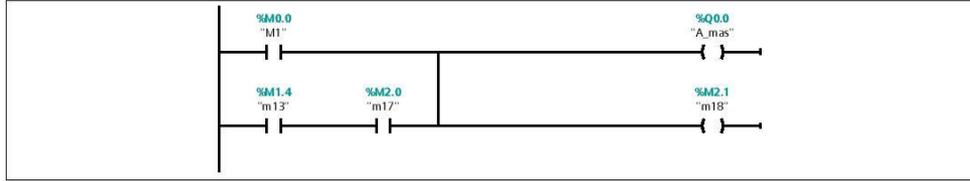
Símbolo	Dirección	Tipo	Comentario
"paro"	%I0.4	Bool	
"M6"	%M0.5	Bool	
"M4"	%M0.3	Bool	
"M5"	%M0.4	Bool	
"IEC_Timer_0_DB_2"	%DB3	IEC_Timer	
T#5S	T#5S	Time	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	

Segmento 6:



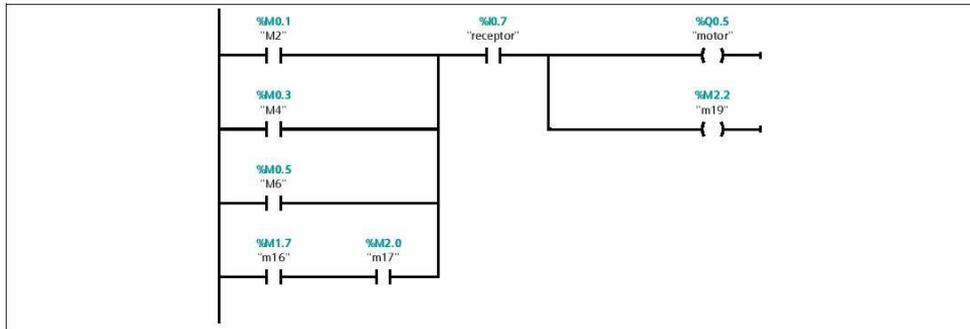
Símbolo	Dirección	Tipo	Comentario
"paro"	%I0.4	Bool	
"M1"	%M0.0	Bool	
"M6"	%M0.5	Bool	
"M5"	%M0.4	Bool	
"sensor3"	%I0.2	Bool	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	

Segmento 7:



Símbolo	Dirección	Tipo	Comentario
"M1"	%M0.0	Bool	
"A_mas"	%Q0.0	Bool	
"m13"	%M1.4	Bool	
"m17"	%M2.0	Bool	
"m18"	%M2.1	Bool	

Segmento 8:



Símbolo	Dirección	Tipo	Comentario
"M2"	%M0.1	Bool	
"M6"	%M0.5	Bool	
"M4"	%M0.3	Bool	
"motor"	%Q0.5	Bool	
"receptor"	%I0.7	Bool	
"m16"	%M1.7	Bool	
"m17"	%M2.0	Bool	
"m19"	%M2.2	Bool	

Segmento 9:

--	--	--	--

Totally Integrated Automation Portal			
Símbolo	Dirección	Tipo	Comentario
"marcha"	%I0.3	Bool	
"paro"	%I0.4	Bool	
"verde"	%Q0.3	Bool	
"rojo"	%Q0.4	Bool	
"m9"	%M1.0	Bool	
"m10"	%M1.1	Bool	
"m11"	%M1.2	Bool	
"m12"	%M1.3	Bool	
"m17"	%M2.0	Bool	