



**UNIVERSIDAD TECNOLÓGICA
INDOAMÉRICA**

**FACULTAD DE INGENIERÍA Y TECNOLOGÍAS DE LA
INFORMACIÓN Y LA COMUNICACIÓN**

CARRERA DE INGENIERÍA INDUSTRIAL

TEMA:

**“IMPLEMENTACIÓN DEL SISTEMA ANDON EN EL ÁREA DE
APARADO CENTRAL PARA LA GESTIÓN AUTOMÁTICA DE
IMPREVISTOS EN UN PROCESO PRODUCTIVO DE LA EMPRESA
PLASTICAUCHO INDUSTRIAL”**

Trabajo de titulación previo a la obtención del título de Ingeniero Industrial.

Autor:

Alex Javier Noboa Castro

Tutor:

Ing. Fernando David Saá Tapia; M.Sc.

AMBATO – ECUADOR

2018

**AUTORIZACIÓN POR PARTE DEL AUTOR PARA LA CONSULTA,
REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN
ELECTRÓNICA DEL TRABAJO DE TÍTULACIÓN**

Yo, Alex Javier Noboa Castro, declaro ser autor del Trabajo de Titulación con el nombre “IMPLEMENTACIÓN DEL SISTEMA ANDON EN EL ÁREA DE APARADO CENTRAL PARA LA GESTIÓN AUTOMÁTICA DE IMPREVISTOS EN UN PROCESO PRODUCTIVO DE LA EMPRESA PLASTICAUCHO INDUSTRIAL”, como requisito para optar al grado de Ingeniero Industrial y autorizo al Sistema de Bibliotecas de la Universidad Tecnológica Indoamérica, para que con fines netamente académicos divulgue esta obra a través del Repositorio Digital Institucional (RDI-UTI).

Los usuarios del RDI-UTI podrán consultar el contenido de este trabajo en las redes de información del país y del exterior, con las cuales la Universidad tenga convenios. La Universidad Tecnológica Indoamérica no se hace responsable por el plagio o copia del contenido parcial o total de este trabajo.

Del mismo modo, acepto que los Derechos de Autor, Morales y Patrimoniales, sobre esta obra, serán compartidos entre mi persona y la Universidad Tecnológica Indoamérica, y que no tramitaré la publicación de esta obra en ningún otro medio, sin autorización expresa de la misma. En caso de que exista el potencial de generación de beneficios económicos o patentes, producto de este trabajo, acepto que se deberán firmar convenios específicos adicionales, donde se acuerden los términos de adjudicación de dichos beneficios.

Para constancia de esta autorización, en la ciudad de Ambato, a los 8 días del mes de agosto de 2018, firmo conforme:

Autor: Alex Javier Noboa Castro

Firma:

Número de Cédula: 180487371-7

Dirección: Tungurahua, Ambato, La Matriz, La Merced.

Correo Electrónico: alexnoba7@hotmail.com

Teléfono: 0984392098

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Titulación “IMPLEMENTACIÓN DEL SISTEMA ANDON EN EL ÁREA DE APARADO CENTRAL PARA LA GESTIÓN AUTOMÁTICA DE IMPREVISTOS EN UN PROCESO PRODUCTIVO DE LA EMPRESA PLASTICAUCHO INDUSTRIAL” presentado por Alex Javier Noboa Castro, para optar por el Título de Ingeniero Industrial,

CERTIFICO

Que dicho trabajo de investigación ha sido revisado en todas sus partes y considero que reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del Tribunal Examinador que se designe.

Ambato, agosto del 2018

.....
Ing. Fernando David Saá Tapia; M.Sc.

DECLARACIÓN DE AUTENTICIDAD

Quien suscribe, declaro que los contenidos y los resultados obtenidos en el presente trabajo de investigación, como requerimiento previo para la obtención del Título de Ingeniero Industrial, son absolutamente originales, auténticos y personales y de exclusiva responsabilidad legal y académica del autor

Ambato, agosto de 2018

.....

Alex Javier Noboa Castro
180487371-7

APROBACIÓN TRIBUNAL

El trabajo de Titulación, ha sido revisado, aprobado y autorizada su impresión y empastado, sobre el Tema: IMPLEMENTACIÓN DEL SISTEMA ANDON EN EL ÁREA DE APARADO CENTRAL PARA LA GESTIÓN AUTOMÁTICA DE IMPREVISTOS EN UN PROCESO PRODUCTIVO DE LA EMPRESA PLASTICAUCHO INDUSTRIAL, previo a la obtención del Título de Ingeniero Industrial, reúne los requisitos de fondo y forma para que el estudiante pueda presentarse a la sustentación del trabajo de titulación.

Ambato, septiembre de 2018

.....

Ing. Pedro Segundo Muzo Villacís M.Sc.
PRESIDENTE DEL TRIBUNAL

.....

Ing. Myriam Emperatriz Cumbajín Alferez M.Sc.
VOCAL

.....

Ing. Edwin Leonardo Sánchez Almeida M.Sc.
VOCAL

DEDICATORIA

Este trabajo lo dedico a Dios principalmente por sus bendiciones, a mis padres, abuelitos y tíos por el apoyo, cariño y amor que me han brindado, a mis hermanos y primos por ver en mi un ejemplo de superación, a ti Nena Carolina (+) porque sé que desde el cielo me observas y me cuidas, a ti hija preciosa por ser mi motor y ese empujón para concluir esta meta.

¡Los quiero mucho!

ALEX JAVIER NOBOA CASTRO

AGRADECIMIENTO

Agradecido a Dios, por guiar cada paso en mi camino y permitirme culminar esta etapa de mi vida.

A mi abuelita la Sra. Hilda Guevara, por su confianza, paciencia y sustento, a mi padre el Sr. Javier Noboa, por impartirme sus conocimientos y darme ánimos para seguir adelante, a mi madre la Sra. Alexandra Castro, por su sin igual amor y perseverancia.

A cada miembro de mi familia y amigos, por su granito de arena, su apoyo incondicional y por estar presentes en cada momento duro y relevante de mi carrera.

A la Universidad Tecnológica Indoamérica, y a sus docentes por abrirme sus puertas formándome como profesional, con una educación de calidad.

A la empresa Plasticaucho Industrial S.A., por brindarme las facilidades de poder realizar esta propuesta en una de sus áreas de producción de calzado.

Al Ing. Fernando Saá, por sus recomendaciones en el desarrollo de este trabajo.

A todos ustedes mi agradecimiento eterno.

ALEX JAVIER NOBOA CASTRO.

ÍNDICE DE CONTENIDOS

PORTADA.....	i
AUTORIZACIÓN PARA EL REPOSITORIO DIGITAL.....	ii
APROBACIÓN DEL TUTOR.....	iii
DECLARACIÓN DE AUTENTICIDAD.....	iv
APROBACIÓN TRIBUNAL	v
DEDICATORIA	vi
AGRADECIMIENTO	vii
ÍNDICE DE CONTENIDOS	viii
ÍNDICE DE TABLAS.	xi
ÍNDICE DE FIGURAS.....	xii
RESUMEN EJECUTIVO	xiv
ABSTRACT.....	xv

CAPÍTULO I

EL PROBLEMA

Introducción.	1
Antecedentes.	3
Justificación.....	5
Objetivos.	6
Objetivo General.	6
Objetivos Específicos.....	6

CAPÍTULO II

INGENIERÍA DEL PROYECTO

Diagnóstico de la situación actual de la empresa.....	7
Desperdicios del proceso en al área de aparado.....	10

Descripción General del proceso de fabricación de calzado.....	12
Descripción de las áreas del proceso de fabricación de calzado.....	13
Área de Estudio.....	16
Modelo Operativo.	18
Desarrollo del modelo operativo.....	19
Analizar el proceso productivo del area de aparato.	19
Establecer los factores que afectan la producción en el área.	19
Implementación del sistema ANDON.	20
Sistema de Control.	20
Desarrollar un registro de datos.	21

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Presentación de la propuesta.	22
Componentes de la propuesta.....	23
Componentes Lógicos.....	23
Sistemas numéricos.....	25
Redes inalámbricas.....	27
Redes XBee.....	27
Tecnologías WI-FI.	28
Componentes Físicos.	28
Arduino MEGA.....	28
Módulo WI-FI ESP8266-01.....	29
Módulo XBee PRO S1.....	30
Módulo XBee Shield.....	30
XBee Explorer.....	31
Relé 5V A 10 amperios.....	31

Luces giratorias.	32
Pulsadores.	33
Batería de litio 3.7 V / 900 mAh.	33
Implementación de la propuesta.....	34
Sistema electrónico.	35
Accionamiento de luces giratorias.	41
Visualización.....	43
Reproducción de melodías.	45
Registro de datos.	46
Comunicación serial entre ESP8266-01 y mega2560.....	48
Resultados esperados.	49
Encender luz giratoria.	49
Página web de datos.	51
Análisis de costos.....	55

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones.	57
Recomendaciones.....	58
BIBLIOGRAFÍA	59
ANEXOS	60

ÍNDICE DE TABLAS.

Tabla 1: Informe de actividades del proceso de apagado.	9
Tabla 2: Registro de averías.....	20
Tabla 3: Sistemas numéricos.	26
Tabla 4: Configuración dispositivos finales XBee.	37
Tabla 5: Configuración del coordinador.	41
Tabla 6: Análisis de costos.	55

ÍNDICE DE FIGURAS.

Figura 1: Promedio de desperdicios previo instalación.	10
Figura 2: Paro de producción por falta de gestión de calidad.....	11
Figura 3: Reporte de paros mensuales 2017 previo instalación.....	12
Figura 4: Área de acopio materia prima.	13
Figura 5: Área de troquelado.	13
Figura 6: Área de aparado.	14
Figura 7: Área de montaje.	15
Figura 8: Área de control de calidad y empaquetado.	15
Figura 9: Ubicación Satelital de la empresa Plasticaucho Industrial.....	17
Figura 10: Flujograma del modelo Operativo.....	18
Figura 11: IDE Arduino.....	23
Figura 12: Software X-CTU.	24
Figura 13: Módulos XBee activando un timbre.....	27
Figura 14: Arduino Mega.	29
Figura 15: Módulo ESP8266.	30
Figura 16: Módulos XBee S1.	30
Figura 17: Módulo XBee Shield.....	31
Figura 18: Módulo XBee Explorer.....	31
Figura 19: Módulo Relés.	32
Figura 20: Luz giratoria.....	32
Figura 21: Pulsador y sus estados.	33
Figura 22: Batería de litio.....	33
Figura 23: Esquema de funcionamiento del sistema ANDON.	34
Figura 24: Flujograma del XBee/Pulsadores.....	35
Figura 25: Acople pulsadores y XBee.	36
Figura 26: Conexión de pulsadores.	38
Figura 27: Flujograma de recepción de datos.....	40
Figura 28: Configuración de XBee como coordinador.....	40
Figura 29: Configuración básica del accionamiento de luces.....	42
Figura 30: Diagrama Acometida 220vac del relé	42
Figura 31: Diagrama instalación de luces giratorias.....	43
Figura 32: Apariencia tablero de visualización.	43
Figura 33: Dimensionamiento del tablero.	44

Figura 34: Diseño de avisos y leyendas.....	44
Figura 35: Diseño final del tablero.	45
Figura 36: Disposición de pines MP3.....	45
Figura 37: Comunicación serial con WT9601M03.	46
Figura 38: Diagrama del circuito de almacenamiento de datos.....	48
Figura 39: Comunicación serial entre esp8266-01 y mega2560.....	48
Figura 40: Visualización de datos en puerto serial.	50
Figura 41: Código para la activación del relé.	50
Figura 42: Código para la reproducción de la pista de audio para mantenimiento.....	51
Figura 43: Conexión con una red doméstica.	51
Figura 44: Encendido de luces giratorias.....	52
Figura 45: Encendido del tablero de información.	53
Figura 46: Interfaz página web.	53
Figura 47: Promedio de desperdicios post instalación.....	54
Figura 48: Reporte de paros mensuales 2017-2018 post instalación.....	55

UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA
FACULTAD DE INGENIERÍA Y TECNOLOGÍAS DE LA
INFORMACIÓN Y LA COMUNICACIÓN
CARRERA DE INGENIERÍA INDUSTRIAL

TEMA: “IMPLEMENTACIÓN DEL SISTEMA ANDON EN EL ÁREA DE APARADO CENTRAL PARA LA GESTIÓN AUTOMÁTICA DE IMPREVISTOS EN UN PROCESO PRODUCTIVO DE LA EMPRESA PLASTICAUCHO INDUSTRIAL”.

AUTOR: Alex Javier Noboa Castro.

TUTOR: Ing. Fernando David Saá Tapia; M.Sc.

RESUMEN EJECUTIVO

En el presente trabajo, el objetivo involucró implementar el sistema ANDON en el área de aparado central, para la gestión automática de imprevistos en un proceso productivo de la empresa Plasticaucho Industrial. Debido a inadecuados métodos de control se presentan los siguientes contratiempos: falla en la maquinaria, excesivo tiempo en los protocolos de edición y reinicio de las máquinas de trabajo y procesos de elaboración de calzado no estandarizados; suscitándose un promedio total de desperdicios de 3,31% el cual 2,36% comprende materia prima y el 0,95% corresponde a pérdidas de tiempo en producción. Se automatizó el proceso, utilizando como herramientas una tarjeta Mega Arduino, módulos XBee Pro S1 que incorporados a un mando de tres pulsadores envían señales mediante comunicación inalámbrica a un módulo coordinador. El mismo que se encargará de activar todo el sistema como; tablero informativo, luces giratorias, melodías y registro de datos, atrayendo la atención del personal encargado de dar solución al inconveniente obteniendo un mejor control del proceso. Los resultados obtenidos permitieron disminuir en un 0,875% los tiempos de pérdidas en producción al pasar de 15,114 horas a 13,804 horas promedio. Además, con la implementación del sistema los tiempos improductivos van tendiendo a la baja paulatinamente gracias a la pronta gestión del responsable a cargo, con maquinaria más confiable y procesos más ordenados. En lo referente al sistema, de las pruebas realizadas se determinó que el alcance de los módulos XBee, cumplen satisfactoriamente su propósito, manteniendo la señal estable, confiable y segura, al utilizar señales de radio frecuencia de libre acceso en la banda de 2.4 GHz, aparatos tales como; teléfonos celulares, redes Wi-Fi, entre otros no afecta al sistema, consiguiendo una información y gestión sin pérdida de datos.

DESCRIPTORES: Arduino (Software - Hardware), calidad, comunicación inalámbrica, mantenimiento, sistema ANDON, XBee (Software - Hardware)

UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA
FACULTAD DE INGENIERÍA Y TECNOLOGÍAS DE LA
INFORMACIÓN Y LA COMUNICACIÓN
CARRERA DE INGENIERÍA INDUSTRIAL

**THEME: “THE ANDON SYSTEM IMPLEMENTATION IN THE SEWING
CENTRAL SECTION FOR AUTOMATIC MANAGEMENT OF
UNEXPECTED EVENTS DURING A PRODUCTION PROCESS AT
PLASTICAUCHO INDUSTRIAL FACTORY”**

AUTHOR: Alex Javier Noboa Castro

TUTOR: Ing. Fernando David Saá Tapia; M.Sc.

ABSTRACT

The aim of this work involved the ANDON system implementation in the sewing central section for automatic management of unexpected events during a production process at Plasticaucho Industrial factory. Due to inadequate control protocols methods, many production setbacks occurred: machinery failure, excessive time in editions of protocols, working machine re-start and non-standardized footwear manufacturing process; which represented an average of 3,31% waste, the raw material consists of 2,36% and 0,95% corresponds to loss of production time. The process was automated through the use of a Mega Arduino card and XBee Pro S1 modules were used to be incorporated as part of a three button control system, which sends signals through wireless communication to a specific module. It will be in charge of activating the entire failure system such as: readable failure information display, visible failure color coded rotating lights, failure alert sound and a memory stored unique code will be created, all of these will catch the personnel's attention which are in charge of give solution to any inconvenience, thus improving the process control. The result of introducing the alert failure protocols, reduced mostly the loss of production time from 15.114 hours to 13.804 hours which represents 0.875% instead of 0.95% originally reported. Additionally, it was found that the loss of production time gradually keeps lowering. As far the used of the system module XBee demonstrated that it fulfills the requirements satisfactorily, since it was proven that the generated failure signal was very stable during the test, the 2.4 GHz bandwidth radio signal transmission was safe to be used under the stress and presence of devices such as: cell phones, Wi-Fi networks, among others, they do not affect the system; gathering information and operation without data loss.

KEYWORDS: ANDON system, Arduino (Software - Hardware), maintenance, quality, wireless, XBee (Software - Hardware).

CAPÍTULO I

EL PROBLEMA

Tema:

“Implementación del sistema ANDON en el área de aparato central para la gestión automática de imprevistos en un proceso productivo de la Empresa PLASTICAUCHO INDUSTRIAL”.

Introducción.

El desarrollo de los procesos de manufactura va acorde a diferentes sistemas que garantizan no solo la obtención del producto final, sino la eficacia y eficiencia en el desarrollo de los mismos. Toyota, vio que el método de trabajo de la producción en masa no les convenía por diversas situaciones del país; como resultado, sus ingenieros Eiji Toyoda y Taiichi Ohno, iniciaron lo que llamarían el Sistema de Producción Toyota, y que más tarde sería nombrado Lean Manufacturing (LM). Esta filosofía de trabajo ha sido divulgada en todo el mundo y puesta en práctica por diferentes sectores productivos tanto de servicios como de manufactura.

Lean Manufacturing es un sistema integrado socio-tecnológico de mejoramiento de procesos, cuyo objetivo principal es eliminar desperdicios o actividades que no agregan valor (Tejeda, 2011).

El sistema Lean Manufacturing agrupa un conjunto de medidas prácticas de comunicación utilizadas con el propósito de plasmar, de forma evidente y sencilla, el estado de algún sistema productivo. De los ramales aplicativos destaca ANDON, expresión de origen japonés que significa "lámpara" y que se relaciona con el control visual considerado como un elemento de la filosofía del sistema Lean (Salazar, 2016).

En Tungurahua; PLASTICAUCHO INDUSTRIAL desea implementar el sistema ANDON en la planta de calzado escolar en el área de aparado central, con el objetivo de conseguir mayor eficiencia en sus procesos, reducir desperdicios, eliminar tiempos muertos, etc.

De acuerdo al conocimiento específico del proceso productivo en el área de aparado, resulta la necesidad de identificar los inconvenientes que generan desperdicios y contratiempos en la producción.

Para identificar los defectos que ocurren durante el proceso productivo, se realiza una observación directa en el área de trabajo y se detecta que el operador no realiza eficazmente las tareas designadas, pierde el tiempo en actividades que no son de su competencia ya sea tratando de reparar una máquina, abandona su lugar de trabajo en busca de abastecimiento de materiales, o localizar a su supervisor para alguna consulta.

Antecedentes.

PLASTICAUCHO INDUSTRIAL, una empresa Ambateña ubicada en el Parque Industrial, inició su funcionamiento en el año 1931, dirigida por su fundador Don José Filometor Cuesta Tapia, dedicada a la producción de calzado en general bajo la marca VENUS, registrada ante el estado ecuatoriano en el año 1938 (PLASTICAUCHO INDUSTRIAL, 2016).

En 1965, PLASTICAUCHO INDUSTRIAL, se constituye como personería jurídica, incorporando nueva tecnología y maquinaria moderna. (PLASTICAUCHO INDUSTRIAL, 2016).

La incorporación de sistemas de control en los procesos productivos cambia los roles de empleadores y trabajadores, destinando nuevas funciones e incorporando metodologías que involucran cambios profundos del trabajo (El nuevo trabajo industrial: una mirada critica desde la calidad del empleo, 2014)

Algunos autores aseguran que la reconfiguración productiva impacta positivamente en la economía en su conjunto a partir de, entre otros factores a nuevas formas de cooperación entre empresarios y trabajadores que permiten recomponer los niveles de productividad (El nuevo trabajo industrial: una mirada critica desde la calidad del empleo, 2014)

Esta nueva relación se sustentaría en la participación activa de los trabajadores y la alineación de objetivos orientados a incrementar el compromiso con la actividad laboral.

La tecnología moderna necesita apoyarse en métodos y procesos que avalen la productividad y la optimización de tiempos.

ANDON es un sistema de señales para reportar fallas en el proceso productivo, ocasionalmente un tablero con luz eléctrica visible para todas las personas que se encuentran en la fábrica, así, cualquier trabajador de la empresa puede alertar acerca de máquinas dañadas, defectos de calidad y otros hechos que requieran atención inmediata para continuar con el proceso (Administración de la producción-UNALMED, 2013).

La aplicación del sistema ANDON a los procesos industriales, tiene como objetivo la mejora continua de la sistematización. El Ecuador no es ajeno a estos sistemas de instrumentación industrial, pero lastimosamente no es utilizado de forma general quedando solo en contadas empresas, siendo una de ellas:

MUNDY HOME Cia.Ltda., empresa ecuatoriana dedicada a la fabricación y distribución de arneses eléctricos (conjunto de cables y conectores), para la industria automotriz, implementa en su planta un sistema ANDON desde el año 2012, desarrollado por estudiantes de la escuela de Ingeniería Eléctrica y Electrónica de la Politécnica Nacional, para el mejoramiento de su producción (Rivadeneira, y otros, 2012).

La detección de averías involucra un proceso de corrección que por lo general se lo hace mediante reportes verbales o escritos. En la manufactura del calzado la empresa cuenta con tres áreas que se encargan de sub procesos como son: troquelado, aparado central, montaje y terminado.

El área de aparado central es una de las más críticas referente a la producción del calzado, ya que en este proceso se va a definir las características principales del producto, tales como: un corte (capellada) sin defectos para su posterior montaje, colocación de hebillas, apliques, punteras, en síntesis, el aparado o costura del calzado.

Justificación.

La implementación del sistema ANDON que es un pilar significativo dentro de la estructura y filosofía del Lean Manufacturing; con el fin de la mejora continua, se alinea a la política de la empresa PLASTICAUCHO INDUSTRIAL por lo cual existe la apertura de la misma para su implementación.

Se genera un **alto impacto** en los métodos de trabajo al reducir al máximo los tiempos muertos por la gestión oportuna durante un imprevisto, al implementar un tablero de control provisto de luminarias, que distinga las áreas en las que se necesita apoyo durante el proceso de aparado.

La presente propuesta es **importante** debido a la necesidad de la empresa PLASTICAUCHO INDUSTRIAL, en mejorar sus procesos y con ellos su productividad, ya que durante los últimos años se ha venido incrementando la demanda, y también por la puesta en marcha del proyecto de manufactura de calzado casual, por lo cual se ha optado por implementar este sistema.

La propuesta posee una **utilidad** práctica muy significativa ya que, mediante la implementación del sistema, se pretende gestionar y solucionar imprevistos en el proceso productivo de forma inmediata y sobre todo confiable, que permita reducir al máximo los desperdicios y reprocesos.

Los resultados obtenidos aplicando esta propuesta **beneficiará** directamente a la empresa PLASTICAUCHO INDUSTRIAL, puntualmente al área de aparado central y a sus trabajadores, optimizando el proceso productivo, generando confianza del trabajo realizado por el personal.

El presente proyecto posee **factibilidad** para su realización, ya que se cuenta con los conocimientos adquiridos para hacer una reingeniería por parte del investigador, así como también el apoyo y supervisión del tutor del proyecto, además de los recursos necesarios para la inversión, el levantamiento de información y la apertura por parte de la empresa.

Objetivos.

Objetivo General.

Implementar el sistema ANDON en el área de aparado central para la gestión automática de imprevistos en un proceso productivo de la Empresa “PLASTICAUCHO INDUSTRIAL”.

Objetivos Específicos.

- Analizar el proceso productivo del área de aparado y la aplicabilidad del sistema ANDON, para la detección y solución de contratiempos.
- Establecer los factores que afectan la producción en el área de aparado de la empresa PLASTICAUCHO INDUSTRIAL.
- Implementar un sistema electrónico para la gestión automática de imprevistos, en el proceso productivo.
- Desarrollar un registro de datos, para obtener información en tiempo real del sistema.

CAPÍTULO II

INGENIERÍA DEL PROYECTO

Diagnóstico de la situación actual de la empresa.

El Ecuador ingresa materia prima para calzado en forma de pieles en bruto (cuero) de ovino pero sin curtir en un volumen de 21,84 toneladas (año 2010) y adicionalmente cueros de res pero sin curtir 147,81 toneladas (año 2010), de esta materia prima según la Asociación de Fabricantes de Calzado del Ecuador se estima que la media nacional de consumo es aproximadamente 2,3 zapatos por habitante anualmente la demanda sería 27 600 000 pares, 45% calzado de cuero, 25% calzado inyectado, 15% calzado deportivo, 15% calzado plástico, textiles y otros.

Las provincias de mayor ocupación en lo relacionado a las actividades artesanales de curtiembre, productos de cuero y calzado son Tungurahua, Imbabura, Azuay y Cotopaxi.

La provincia de Tungurahua representaría 75,6% de trabajo artesanal total del sector, sus principales talleres se localizan en los cantones de Ambato, Quisapincha, Baños y Cevallos. Esta actividad en la Zona 3 genera alrededor de 50000 plazas de trabajo formal e informal (SENPLADES, 2013 - 2017).

PLASTICAUCHO INDUSTRIAL, actualmente con su producción de 1'500.000 pares por mes en todas sus líneas, cubre las necesidades del sector; sin embargo, se pretende optimizar el proceso productivo con la implementación de un sistema que permita la detección temprana de imprevistos, tales como los desperdicios generados por la mala utilización de los tiempos por parte de MOD, fallos de maquinaria crítica, falta de abastecimiento de materiales, entre otros, que se presentan como los principales problemas dentro del área de aparato en la empresa.

En la sección de aparado de calzado escolar en la actualidad tiene la capacidad de planificación de 8 células por turno un total de 24 células día, conformadas entre 6 y 7 personas cada una, las cuales están capacitadas al 100% para todos los modelos de la línea si hablamos de recurso humano y de disponibilidad de máquinas.

Entre las actividades del proceso de aparado se destacan las siguientes ANEXO H:

- Recepción de cortes de Troquelado.
- Cosido en máquinas automátatas.
- Colocado de punteras.
- Servicios.
- Distribución de planes a células.
- Aparado de cortes.
- Moldeado de talón.
- Revisión de calidad.
- Entrega de cortes a bodega.
- Incumplimientos a responsabilidades.

La Tabla 1, indica el análisis donde se identifica las actividades que no agregan valor en el proceso productivo en el área de aparado, el presente informe es realizado por el departamento de tiempos y movimientos cuyos resultados establecen un tiempo total de 4 horas 50 minutos para realizar la manufactura de los modelos Tito y Braga de diferentes tallas, detectando 3 actividades que no agregan valor sumando un tiempo de 39 minutos.

Tabla 1: Informe de actividades del proceso de aparado.

ACTIVIDAD	COMIENZA/Tiempo de una actividad	TERMINA/Número de veces que se repite una	DURACIÓN DE CADA ACTIVIDAD	OBSERVACIONES
Bar	9:35:00	10:00:00	0:25:00	
Tiempo de costura	RESUMEN		2:44:00	
Cambio de bobina	0:00:43	7	0:05:01	El tiempo que se demora en cambiar la bobina es de 0,43 seg, esto lo
Tiempo de edición	0:02:50	6	0:17:00	No agregan valor
Se cuelga la máquina. Se debe apagar la máquina, encender, buscar al líder para que ingrese la clave al sistema, realizar el home, cargar nuevamente	0:04:00	3	0:12:00	No agregan valor
Acomodar los cortes en la mesa para un nuevo lote	0:02:00	10	0:20:00	
Cambiar de pallets para otro modelo, deja en las perchas los que no va a utilizar, trae los pallets que va a utilizar para el nuevo modelo. Coloca los pallets en el bastidor, cambia de	RESUMEN		0:15:00	
Ir al baño, beber agua	11:24:00	11:29:00	0:05:00	
Ingresan y realizan el sorteo para saber que modelo se va a coser	14:00:00	14:10:00	0:10:00	No agregan valor
Halar cortes hasta la maquina donde van a coser. Traen la bobina, cambian de hilos, buscan los pallets, ordenan las javas, colocan las piezas en la mesa para coser, acomodan pallets en los bastidores, cargan el programa con	14:10:00	14:27:00	0:17:00	
TOTAL			4:50:00	
CONCLUSIONES Se empezó a tomar el tiempo desde las 9:35 hasta las 14:27, en la máquina OW-01, como operador el Sr. Gustavo Calucho. Se cosieron los modelos deportivos Braga y Tito de diferentes tallas. El tiempo total es de 4 horas con 50 minutos. Mediante este informe se detectaron 3 actividades que no agregan valor, las cuales suman un tiempo de 39 minutos y están identificadas en las observaciones.				

Fuente: Plasticaucho Industrial.

Realizado por: Noboa, 2017.

Desperdicios del proceso en al área de aparado.

En el área de aparado se genera un promedio total de 3.31% de desperdicios, donde los porcentajes se dividen en 0,95% y 2,36% respectivamente según la Figura 1. En la materia prima se presenta el mayor porcentaje de desperdicios donde se refleja que el cuero esta con defectos de fábrica como lacras, venas, lluros y tajos, lo cual dicha problemática no corresponde al área de estudio.

El 0,95% corresponde a desperdicios de producción como, mal armado de talón, mal armado de puntas, malcarados entre otros lo que se pretende reducir aplicando el sistema ANDON con lo cual se garantiza la optimización progresiva del proceso, oportuna asistencia a imprevistos con lo cual se aprovecha de mejor manera los tiempos de trabajo.

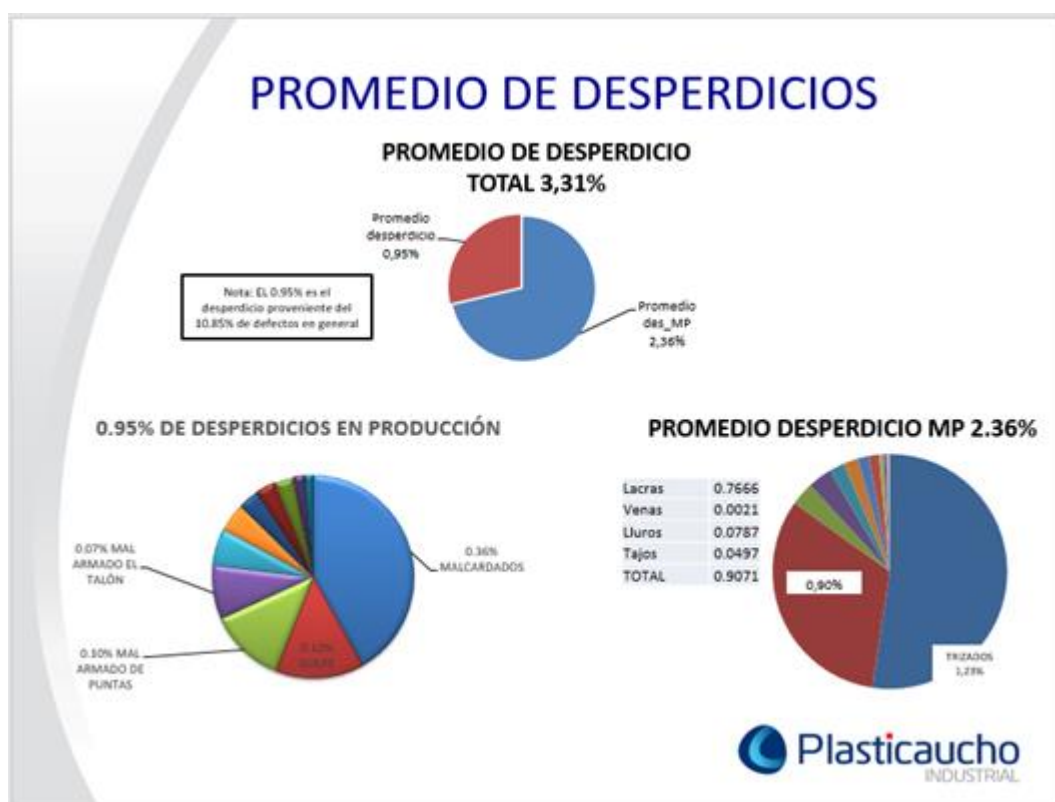


Figura 1: Promedio de desperdicios previo instalación.

Fuente: PLASTICAUCHO INDUSTRIAL.

Para caracterizar la problemática anterior se describe el siguiente ejemplo tomado como referencia en la denominada célula “lean”, y la información generada desde el sistema SAP que maneja la empresa.

La Figura 2, indica un problema en la gestión tardía del departamento de calidad, en la Untadora-ES09; al presentarse una anomalía de funcionamiento, provoca que el aditivo manche la suela, originándose un paro de producción hasta poder corregir el defecto. Se observa también el tiempo de paro de una hora, el inicio y fin del tratamiento de la avería como muestra el sistema.

Equipo	1000690			UNTADORA - ES09	
Datos avería					
Inicio avería	20.05.2016	12:00:00	<input checked="" type="checkbox"/> Parada		
Fin de avería	20.05.2016	13:00:00	Duración parada	60.00	MIN
Circunstancias					
Codificación	PNOPLANE	0001	FALTA DE MATERIALES		
Descripción	PARO DE PRODUCCION				
PARO DE PRODUCCION POR DEVOLUCION DE SUELAS MANCHADAS SUSY II A BODEGA					
Responsabilidades					
Grupo planif.	107	/	PI10	TEC-MTTO-MONTAJE	
<input checked="" type="checkbox"/> Mensaje está concluido, por esta razón sólo es posible visualizar					

Figura 2: Paro de producción por falta de gestión de calidad.

Fuente: PLASTICAUCHO INDUSTRIAL.

Para el presente estudio de los datos provistos por la empresa se compara los tiempos de paros del año 2017 desde el mes de abril hasta agosto, que es antes de la instalación del sistema ANDON, como indica la Figura 3. El contraste se lo hace con los meses octubre, noviembre, diciembre año 2017 y marzo 2018 como indica la Figura 48.

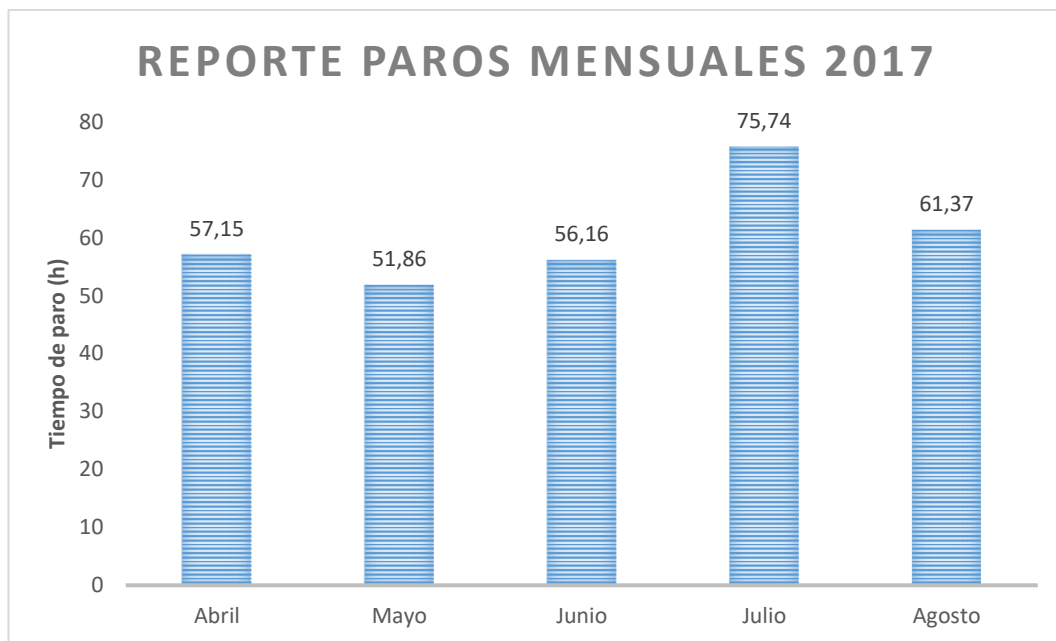


Figura 3: Reporte de paros mensuales 2017 previo instalación.
Fuente: PLASTICAUCHO INDUSTRIAL.

Descripción General del proceso de fabricación de calzado.

En los últimos años PLASTICAUCHO INDUSTRIAL ingresa a una nueva línea de negocio como es calzado escolar, elaborado en base de la materia prima que es la piel de res (cuero), el cual ingresa a una selección minuciosa tomando en consideración que se trata de un material noble, natural y transpirable que ofrece infinidad de posibilidades para su procesamiento.

Después de su selección, el proceso de fabricación inicia con el diseño, dotando al calzado de estética y funcionalidad, para aquello suele recurrirse a patrones de corte, en la actualidad se lo realiza con herramientas informáticas.

El diseño es enviado al área de troquelado en donde se cortan las piezas que conforman al zapato, tales como la capellada, lengüetas, talones, etc., las mismas que, por consiguiente, pasa al área de aparado donde se cosen las piezas mencionadas anteriormente, se ponen punteras, forros, se queman hilos y finalmente al área de montaje, terminado y empaque.

Descripción de las áreas del proceso de fabricación de calzado.

Área de acopio de materia prima.

La Figura 4, indica el área donde ingresan los paquetes de cuero provenientes de la tenería, los cuales serán sometidos a un control de calidad.



Figura 4: Área de acopio materia prima.
Realizado por: Noboa, 2017.

Área de troquelado.

La Figura 5, indica la maquinaria que se usa en este proceso, tales como, las troqueladoras manuales y troqueladoras automatizadas, su función es cortar los modelos de calzado y plantillas.



Figura 5: Área de troquelado.
Realizado por: Noboa, 2017.

Área de aparado.

La Figura 6, indica el área donde se realiza nuestra propuesta, se puede observar las máquinas de costura, en donde, una vez confeccionadas las piezas que conforman la parte superior del zapato (empeine y forro), se procede a su unión mediante cosido.



Figura 6: Área de aparado.
Realizado por: Noboa, 2017.

Área de montaje.

La Figura 7, indica el proceso de montaje, el cual, ya obtenido el corte cosido se lo monta en la horma y se lo pega a la planta, se arma punteras, talones y posteriormente se lo une a la suela mediante adhesivos e incluso costura.



Figura 7: Área de montaje.

Realizado por: Noboa, 2017.

Área de control de calidad y empaque.

La Figura 8, indica el proceso de control de calidad, los acabados mejoran el aspecto externo del calzado, eliminando pequeñas arrugas, limpiando defectos, dándole el brillo necesario y colocándole la plantilla correspondiente.

Para concluir, cada zapato se hermana con su pareja, se lo protege con papel de seda y se coloca en su caja. Las cajas se remiten al departamento de logística quienes se encargan de su almacenamiento para su posterior distribución a varios puntos de venta del país y del exterior.



Figura 8: Área de control de calidad y empaquetado.

Realizado por: Noboa, 2017.

Área de Estudio.

Delimitación del objeto de investigación.

Dominio:	Tecnología
Línea de Investigación:	Empresarial y Productividad
Campo:	Ingeniería Industrial
Área:	Control de Procesos
Aspecto:	Productividad
Objeto de estudio:	Reingeniería de Procesos y Productividad
Periodo de análisis:	Febrero 2017 – Septiembre 2018

Ubicación.

La empresa se ubica en la panamericana norte Km 10, en el Parque Industrial Ambato (P.I.S.A.).



Figura 9: Ubicación Satelital de la empresa Plasticaucho Industrial.

Fuente: Google Maps.

Modelo Operativo.

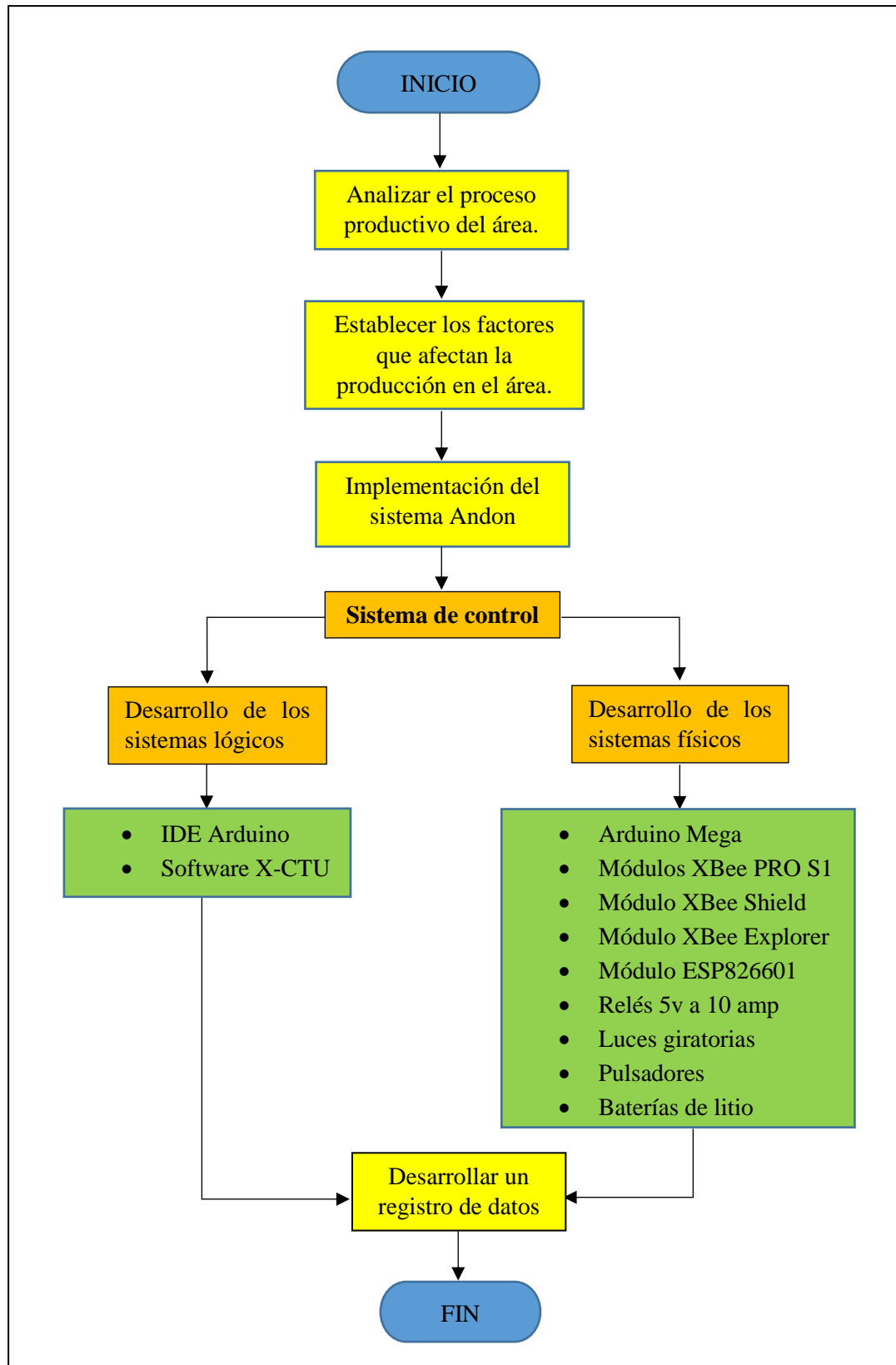


Figura 10: Flujograma del modelo Operativo.

Realizado por: Noboa, 2017.

Desarrollo del modelo operativo

Para llevar a cabo el modelo operativo descrito en el flujograma que se presenta en la Figura 10 se cumple con algunos pasos, los mismos que se detallan a continuación:

Analizar el proceso productivo del área de aparado.

Consiste en la comprobación del proceso que desarrolla cada trabajador u operario en función del área de trabajo, caracterizada tanto por las labores manuales como la de manejo de equipos o maquinaria.

Se solicita al departamento de tiempos y movimientos un registro de actividades del proceso con sus respectivos períodos de trabajo, respecto al documento solicitado se empezó a tomar el tiempo desde las 9:35 hasta las 14:27, en la máquina OW-01, donde se cosieron los modelos deportivos Braga y Tito de diferentes tallas.

El tiempo total en este proceso de aparado es de 4 horas con 50 minutos detectando 3 actividades que no agregan valor, las cuales suman un tiempo de 39 minutos y están identificadas en las observaciones. tal y como se evidencia en la Tabla 1 anteriormente citada.

Establecer los factores que afectan la producción en el área.

Para establecer los factores que afectan en la producción, se realiza una observación directa al área de aparado y se comprueba que el operador no realiza eficazmente sus tareas asignadas, por prestar atención a otras actividades que no le corresponden tales como, arreglar la maquina a su cargo, buscar suministros o localizar a su supervisor para alguna consulta, actividades que no agregan valor ocasionando tiempos improductivos en el proceso.

Otros factores que afectan en la producción del área de aparado son las fallas en la maquinaria, para su constatación se involucra en el funcionamiento de las mismas observando que se generan averías, como indica el informe de la Tabla 2, ocasionando costuras defectuosas por la mala calibración de estas, errores en edición y mecanizado, consumo innecesario de energía y perdida de materiales por destrucción, esto conlleva a desperdicios en la producción como se menciona

anteriormente en el diagrama de la Figura 1.

Tabla 2: Registro de averías.

Aviso	Orden	Denominación	Causas avería	Texto causa	Duración	Unidad
2053838	10070956	APARADO COLUMNA 2 AGUJAS(674)-0N13	0037	ROTO SEGURO DE REGULADOR DE PUNTADA	0.5	h
2053866	10071061	APARADO COLUMNA 2 AGUJAS(674)-0N14	0037	PIN DE REGULACIÓN DE PUNTADA ROTO	0.5	h
2053940	10071491	APARADO COLUMNA 1 AGUJA(674)-0L16	0031	PIN DE REGULACIÓN DE PUNTADA ROTO	0.67	h
2053935	10071460	APARADO COLUMNA 1 AGUJA(674)-0L16	0031	GARFIO ATRASADO	0.33	h
2053924	10071391	APARADO COLUMNA 1 AGUJA(674)-0L16	0031	DESINCRONIZACIÓN GARFIO	0.5	h
2053922	10071404	APARADO ZIGZADORA – PJ04	0031	GARFIO LASTIMADO	0.5	h
2053896	10071069	APARADO ZIGZADORA – PJ04	0031	GARFIO ADELANTADO	0.33	h

Fuente: Plasticaucho Industrial.

Realizado por: Noboa, 2017.

Implementación del sistema ANDON.

Para la implementación se desarrolla un sistema de control, basado en sistemas electrónicos, además de un registro de datos que proporcione información a los involucrados en el proceso.

Sistema de Control.

Realizado el estudio del proceso, se establece que lo adecuado es realizar un sistema de control electrónico, ver ANEXO E, mediante sistemas físicos y lógicos, con el fin de obtener un mejor control del proceso por parte del personal técnico, de calidad y de supervisión y reducir de esta manera el tiempo de reacción ante un imprevisto.

Desarrollo de los sistemas físicos.

Se realiza la implementación de nuevas tecnologías, basadas en sistemas micro controlados, como una tarjeta Arduino y módulos XBee, los mismos que permitirán una comunicación inalámbrica para la optimización y gestión de los procesos de automatización, ver ANEXO F.

Desarrollo de los sistemas lógicos.

Se utiliza software de programación, como el IDE Arduino (Figura 11) y el X-CTU (Figura 12) que sirve para la programación de los parámetros de los módulos XBee y así generar los códigos de comunicación.

Tablero de visualización.

Elaborado en madera MDF e iluminación LED se construye un tablero que permite visualizar la necesidad de intervención del personal a cargo ver Figura 35. Las células y el servicio requerido serán iluminadas por una serie de leds de distinto color cuando se requiera atención ver Figura 45.

Desarrollar un registro de datos.

Se desarrolla un sistema de monitoreo vía Wifi mediante el módulo ESP8266 y en tiempo real para obtener información a través de una página Web desarrollada en lenguaje HTML, ver Figura 46, de esta manera comprobar cuánto ha mejorado el proceso reformado respecto del original.

CAPÍTULO III

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

Presentación de la propuesta.

PLASTICAUCHO INDUSTRIAL dentro de su plan estratégico para los años posteriores implementa en sus plantas de producción el Lean Manufacturing.

Entendiendo que, al eliminar desperdicios, la calidad debe aumentar, mientras que los tiempos y costos de producción deberán disminuir en corto tiempo.

El sistema Andon como pilar del Lean Manufacturing se instala en el área de apartado central, para el apoyo en la gestión de imprevistos basado en una red inalámbrica de pulsadores que activan ciertos sonidos, además de luces giratorias que indican en donde se produce tal anomalía.

Tomando en cuenta que la mayor parte de la información que captan las personas proviene de señales visuales y los sonidos, se construye además un tablero que contiene las leyendas del servicio (calidad, mantenimiento, supervisión), así como de los puestos de trabajo (células).

En el presente capítulo se desarrolla el contenido siguiente:

- Introducción a los componentes lógicos y físicos de la propuesta.
- Selección de componentes lógicos y físicos, realizando una descripción de cada uno de ellos.
- Finalmente, la implementación del sistema ANDON.

Componentes de la propuesta.

Componentes Lógicos.

Los componentes lógicos incluyen, entre muchos otros, aplicaciones informáticas, tales como el procesamiento de texto, que permite al usuario realizar todas las tareas concernientes a la edición de algoritmos; el llamado software, tal como el sistema operativo de un PC, que básicamente permite al resto de los programas funcionar adecuadamente, facilita la interacción entre los componentes físicos y el resto de las aplicaciones, proporcionando una interfaz al usuario.

IDE Arduino.

Un IDE (entorno de desarrollo integrado), es un entorno de programación que consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI).

Además, en el caso de Arduino incorpora las herramientas para cargar el programa ya compilado en la memoria flash del hardware.

Los programas de arduino están compuestos por un solo fichero con extensión “ino”, aunque es posible organizarlo en varios ficheros.

El fichero principal siempre debe estar en una carpeta con el mismo nombre que el fichero.

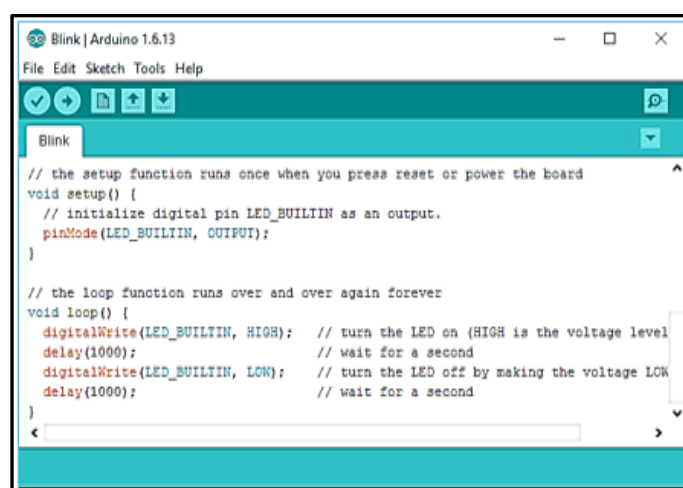


Figura 11: IDE Arduino.

Fuente: Software Arduino 2018.

La Figura 11, indica la interfaz del software IDE Arduino, donde se puede visualizar los códigos de programación para la comunicación con módulos externos

X-CTU.

Es una aplicación multiplataforma diseñada para permitir a los desarrolladores interactuar con módulos Digi RF a través de una interfaz gráfica fácil de usar.

Incluye todas las herramientas que un desarrollador necesita para comenzar a trabajar rápidamente con módulos XBee.

Funciones únicas como la vista de red gráfica, junto con la intensidad de la señal de cada conexión, y el generador de tramas XBee API.

Puede administrar y configurar varios dispositivos de RF (radio frecuencia), incluso dispositivos conectados remotamente (por aire).

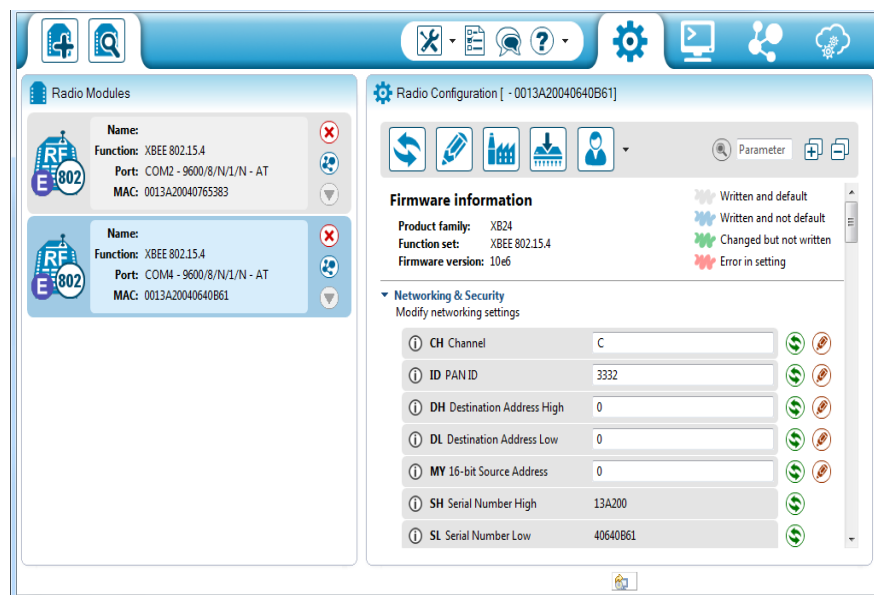


Figura 12: Software X-CTU.

Fuente: Digi.

La Figura 12, indica la interfaz principal del software X-CTU, para la configuración de los módulos XBee.

Sistemas numéricos.

Es importante tener una idea clara de estos sistemas, ya que para manipular los módulos XBee se necesita saber, qué tipo de información se está enviando y que se está recibiendo.

Cuando se trabaja con circuitos lógicos se debe hacer referencia a los datos que se ingresan en la memoria del microcontrolador, que en ocasiones son de tipo numéricos; generalmente decimal, binario o hexadecimal.

Un Sistema numérico es un conjunto de símbolos y reglas que se utilizan para representar datos numéricos o cantidades.

Sistema decimal.

Este es el sistema que se maneja cotidianamente, está formado por diez símbolos {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} por lo tanto la base del sistema es diez (10).

Sistema binario.

Es el sistema que utiliza internamente el hardware de las computadoras actuales, se basa en la representación de cantidades utilizando los dígitos 1 y 0. Por tanto su base es 2 (número de dígitos del sistema). Cada dígito de un número en este sistema se denomina bit (contracción de binary digit).

Se puede utilizar con nombre propio determinados conjuntos de dígitos en binario.

Cuatro bits se denominan cuaterno (ejemplo: 1001), ocho bits octeto o byte (ejemplo: 10010110), al conjunto de 1024 bytes se le llama Kilobyte o simplemente K, 1024 Kilobytes forman un megabyte y 1024 megabytes se denominan Gigabytes.

Sistema octal.

El sistema numérico octal utiliza ocho símbolos o dígitos para representar cantidades y cifras numéricas. Los dígitos son: {0, 1, 2, 3, 4, 5, 6, 7}; la base de éste es ocho (8) y es un sistema que se puede convertir directamente en binario como se verá más adelante.

Sistema hexadecimal.

El sistema numérico hexadecimal utiliza dieciséis dígitos y letras para representar cantidades y cifras numéricas. Los símbolos son: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}; la base del sistema es dieciséis (16).

En la Tabla 3, se muestran los primeros veinte y un número decimal con su respectiva equivalencia binaria, octal y hexadecimal.

Tabla 3: Sistemas numéricos.

DECIMAL	BINARIO	OCTAL	HEXADECIMAL
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Fuente: Digi.

Realizado por: Noboa, 2017.

Redes inalámbricas.

Es una clase de comunicación que se realiza entre un transmisor y un receptor, sin necesidad de un medio físico, sino a través de ondas electromagnéticas.

La red inalámbrica permite la comunicación de dispositivos que estén cerca o distantes, facilitando la comunicación sin necesidad de infraestructuras costosas, además en cuestión de mantenimiento no es muy relevante, lo que supone comodidad y reducción de costos.

Redes XBee.

Una red XBee la forman básicamente 3 tipos de elementos. Un único dispositivo Coordinador, dispositivos Routers y dispositivos finales (end points). Los módulos XBee son versátiles a la hora de establecer diversas topologías de redes.

El Coordinador: Es el responsable de establecer el canal de comunicaciones y del PAN ID (identificador de red) para toda la red. Una vez establecidos estos parámetros, el coordinador puede formar una red, permitiendo unirse a él a dispositivos finales o end device.

Formada la red, el coordinador hace las funciones de Router, esto es, participar en el enrutado de paquetes y ser origen y/o destinatario de información.

End Device: Los dispositivos finales no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un coordinador o un Router, es decir, no puede enviar información directamente a otro end device. Normalmente estos equipos van alimentados a baterías. El consumo es menor al no tener que realizar funciones de enrutamiento.

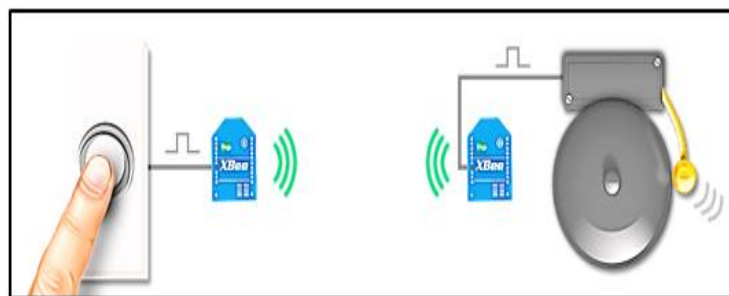


Figura 13: Módulos XBee activando un timbre.
Fuente: XBee.

La Figura 13, indica la activación de un timbre utilizando módulos XBee, sin necesidad de cableado.

Modo de operación API.

Cuando la configuración API está activada, cada paquete RF que se envía o recibe, se encapsula en un frame de datos.

La parte más importante del modo API es que brinda la posibilidad de conocer el estado del módulo, enviar y recibir paquetes de datos, recibir lecturas de las entradas digitales y analógicas, etc.

Tecnologías WI-FI.

Tecnología de comunicación inalámbrica WI-FI, utilizan ondas electromagnéticas que se transmiten a través de bandas de frecuencias libres, sus longitudes de ondas son cortas, utilizan potencia de transmisión muy baja, se le denomina WLAN o estándar IEEE 802.11.

Comúnmente WI-FI opera en la frecuencia 2.4Ghz, los equipos que transmiten a esta frecuencia son utilizados para distancias cortas, debido a su corto rango de cobertura.

Componentes Físicos.

Arduino MEGA.

Un microcontrolador es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria.

Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica.

Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y puertos de entrada/salida.

La Mega 2560 es una placa electrónica basada en el Atmega2560. Cuenta con 54 pines digitales de entrada / salida (de los cuales 15 se pueden utilizar como salidas

PWM), 16 entradas analógicas, 4 UARTs (puertos serie de hardware), un oscilador de 16MHz, una conexión USB, un conector de alimentación, un conector ICSP, y un botón de reset.

El ATmega2560 posee una memoria flash de 256KB para almacenar el código, 8KB son utilizados en el arranque.

Los pines digitales pueden proporcionar o recibir hasta 20mA, además poseen una resistencia pull-up entre 20-50 kohms.

Los pines digitales para la utilización de los puertos Serie son: 0 (RX) y 1 (TX); Serial 1: 19 (RX) y 18 (TX); Serial 2: 17 (RX) y 16 (TX); Serial 3: 15 (RX) y 14 (TX).

Se utilizan para recibir (RX) y transmitir (TX) datos en serie. Consta de 16 entradas analógicas, con una resolución de 10 bits, es decir 1024 valores distintos, que van desde 0 a 5V.



Figura 14: Arduino Mega.
Fuente: IndiaMart.

Módulo WI-FI ESP8266-01.

Este módulo contiene un SOC (System on Chip) integrado, compatible con el protocolo TCP/IP. Su objetivo principal es dar acceso a cualquier microcontrolador a una red Wifi, trabaja en la frecuencia 2.4GHz, cada módulo viene pre-programado con firmware de comandos AT (conjunto de comandos estándar), para conectarlo al microcontrolador.

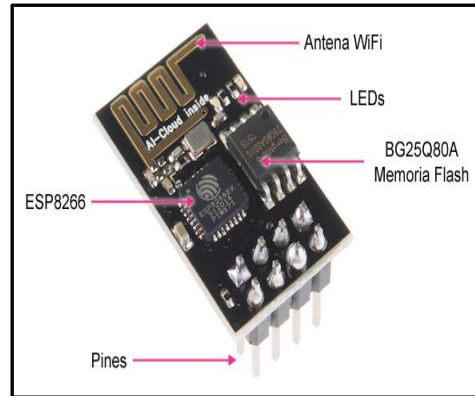


Figura 15: Módulo ESP8266.
Fuente: Sánchez Rubén, 2016.

Módulo XBee PRO S1.

Los módulos XBee de digi son pequeños módulos RF (radio frecuencia) que transmiten y reciben datos sobre el aire usando señales de radio. La capacidad inalámbrica es esencial cuando se quieren instalar sensores en lugares donde no hay cables. Son altamente configurables y soportan múltiples protocolos para permitir usarlo tanto en un enlace punto a punto o con más dispositivos en una red.

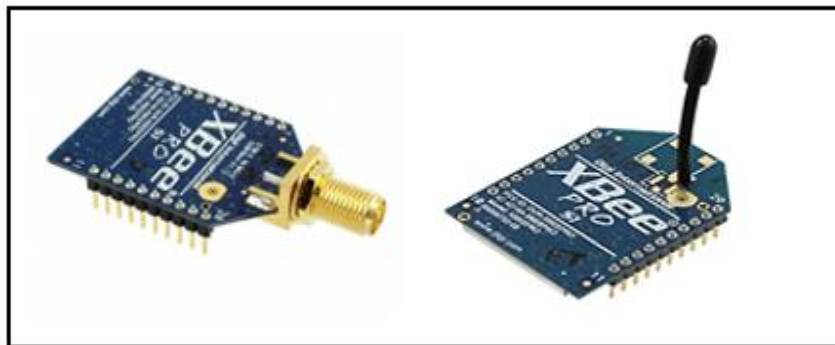


Figura 16: Módulos XBee S1.
Fuente: Digi.

Módulo XBee Shield.

El XBee Shield simplifica la tarea de conectar un módulo XBee con Arduino. Esta tarjeta puede ser utilizada directamente proporcionándole de comunicación inalámbrica utilizando el módulo de comunicaciones XBee. Esta unidad funciona con los módulos XBee de la serie 1 y 2.5 en sus versiones estándar y pro.

Los pines seriales (DIN y DOUT) del XBee están conectados a unos jumpers, con lo que puede conectarse a la UART (D0, D1) o a los pines digitales 2 y 3 del

Arduino. La alimentación de 5V la toma directamente, debido a que posee un regulador de voltaje de 3.3VDC para poder alimentar el XBee (Thayer Ojeda).

Es importante remover el módulo XBee de Arduino mientras se lo programa para establecer la comunicación serial. Para evitar conflictos al momento de cargar el programa en el módulo, se debe retirar uno de los jumper y volverlo a su sitio después de haberlo cargado.

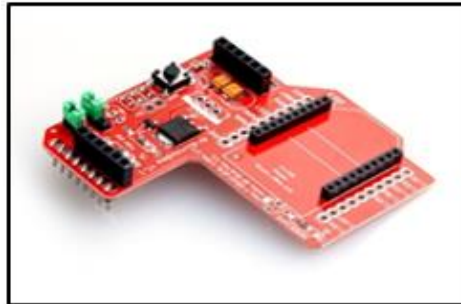


Figura 17: Módulo XBee Shield.
Fuente: Digi

XBee Explorer.

Se la utiliza para la programación y configuración de los módulos mediante un cable mini USB conectado al computador, ya que provee del voltaje óptimo de 3.3V en corriente continua para su manejo.



Figura 18: Módulo XBee Explorer.
Fuente: Digi.

Relé 5V A 10 amperios.

Elemento electrónico que permite la conmutación de un estado a otro, para cerrar o abrir el circuito de ser el caso, con voltajes AC de 110 hasta 220, internamente consta de una bobina la cual se activa aplicando voltajes a partir de 5v.

Este módulo es indispensable para el proyecto, porque mediante sus terminales se conectarán los cables que van hacia las luces giratorias.

Además, contiene un optoaislador, dispositivo que funciona como interruptor activado mediante luz, con la finalidad de aislar el circuito electrónico con la parte de potencia.



Figura 19: Módulo Relés.
Realizado por: Noboa, 2017.

Luces giratorias.

Aparatos que utilizan la última tecnología LED para emitir una señal visual. Configuradas en secuencia para crear un efecto de luz giratoria (rotación simulada).

Baliza compacta con larga vida útil que elimina la necesidad de cambiar la lámpara constantemente.

Todas las unidades cuentan con lentes de policarbonato UV estable.

Están distribuidas en medio de dos células de apagado para la visualización de algún imprevisto o requerimiento de servicio.



Figura 20: Luz giratoria.
Fuente: GAMSCO.

Pulsadores.

Por lo general son activados al ser pulsados con un dedo, permitiendo el flujo de corriente mientras son accionados.

Cuando ya no se presiona sobre él vuelve a su posición de reposo, en su interior tiene dos contactos que, al ser pulsado, si es un dispositivo NA (normalmente abierto) se cierra, si es un dispositivo NC (normalmente cerrado) será abierto.

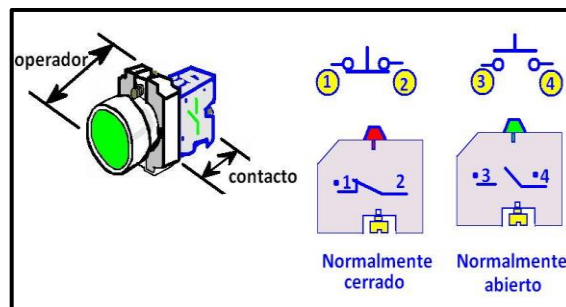


Figura 21: Pulsador y sus estados.

Fuente: coparoman. Blogspot

Batería de litio 3.7 V / 900 mAh.

Para proporcionar libertad de movimiento a los mandos inalámbricos, se optó por alimentarlos con baterías de litio.

La batería es una pila recargable con dos o más celdas donde están separados los iones de litio, cuando funciona en modo de descarga los iones de una y otra celda se combinan químicamente para formar el elemento estable, esta combinación se produce de forma exo-térmica, es decir, produce energía que es la que se aprovecha cuando se ha agotado la batería.



Figura 22: Batería de litio.

Fuente: HOWELL.

Implementación de la propuesta.

Al sistema se lo define en las siguientes partes:

- Sistema electrónico.
- Sistema de visualización y sonido.
- Sistema de almacenamiento y presentación de datos.

El esquema de la Figura 23, indica el funcionamiento del sistema ANDON.

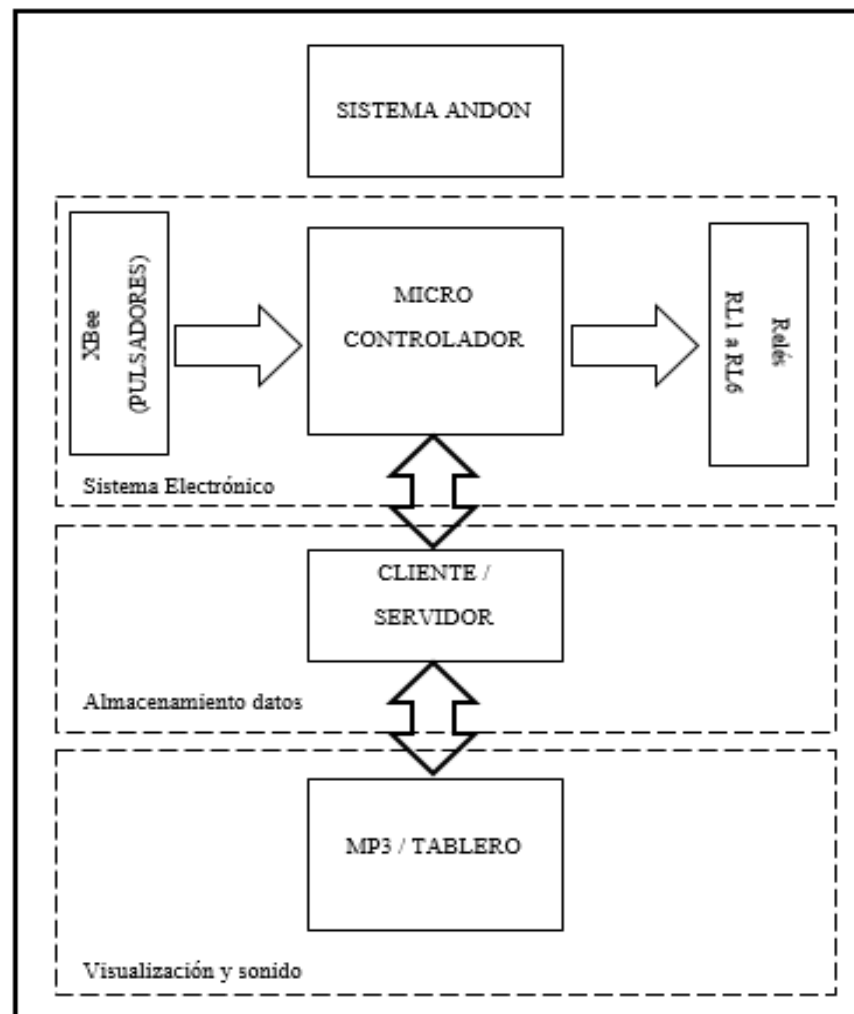


Figura 23: Esquema de funcionamiento del sistema ANDON.
Realizado por: Noboa, 2017.

Sistema electrónico.

Es un conjunto de dispositivos que se ubican dentro del campo de la ingeniería y la física, los cuales pueden generar (XBee), recibir, transmitir o almacenar información (arduino).

Los sistemas electrónicos ofrecen diferentes funciones para procesar dicha información: amplificación de señales débiles para que pueda utilizarse correctamente, generación de ondas de radio, extracción de información, operaciones lógicas como los procesos electrónicos que se desarrollan en los ordenadores, etc.

Generación de datos.

Pulsadores XBee.

Los dispositivos finales (pulsadores) no tienen capacidad de enrutar paquetes. Deben interactuar siempre a través de su nodo padre, ya sea este un Coordinador, es decir, no puede enviar información directamente a otro dispositivo final. Normalmente estos equipos van alimentados a baterías. El consumo de energía es menor al no tener que realizar funciones de enrutamiento.

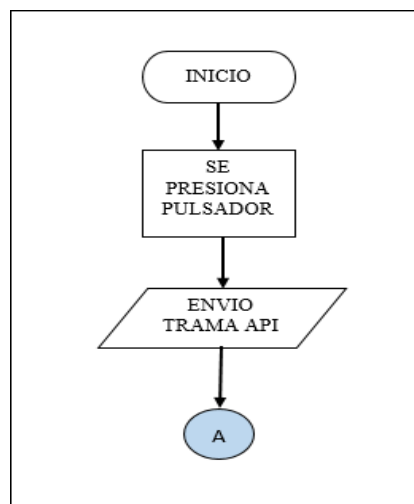


Figura 24: Flujograma del XBee/Pulsadores.
Realizado por: Noboa, 2017.

La Figura 24, indica el flujograma del proceso del XBee conjuntamente con los pulsadores, para el envío de información al arduino, el cual inicia apretando un pulsador, el mismo que genera una trama API o código para que dé inicio a la siguiente actividad (A) que se detalla en el flujograma de la Figura 27.

En un cajetín se acondiciona tres pulsadores a los que mediante un láser se inscribe las siglas CAL (calidad), MAN (mantenimiento), SUP (supervisión) respectivamente.

Luego los terminales se empatan con el módulo XBee pro S1, quien será el encargado de generar las señales mediante radio frecuencia de los servicios y estación de trabajo (célula), que requiera atención.

La Figura 25, indica la disposición de los pulsadores con el XBee.



Figura 25: Acople pulsadores y XBee.

Realizado por: Noboa, 2017.

Para que el módulo transmita información se lo configura mediante el software X-CTU, el mismo que le transfiere los parámetros para su operación.

La Tabla 4, indica la configuración de cada uno de los dispositivos.

Tabla 4: Configuración dispositivos finales XBee.

PARAMETRO	C1 – 2	C3 – 4	C5 – 6	C7 – 8	C9 - 10	AU
CH	C	C	C	C	C	C
ID	3332	3332	3332	3332	3332	3332
DH	0	0	0	0	0	0
DL	1	1	1	1	1	1
MY	C1	C2	C3	C4	C5	C5
IR	1	1	1	1	1	1
CE	0	0	0	0	0	0

Fuente: Xctu.

Realizado por: Noboa, 2017.

Algunos de los parámetros más importantes, y que deben ser configurados en todos los módulos XBee que forman parte de la red, son:

CH (*Chanel*):

Es el canal de comunicación, el mismo que debe ser igual para todos los módulos de la red, para la propuesta canal C.

ID (*Identity*):

Determina el identificador de la red (*PAN ID*) a la que pertenece el nodo.

La red del proyecto tiene como identificador el valor 3332.

NI (*Node Identity*):

Cadena de caracteres usada como identificador del nodo.

El microcontrolador determina dinámicamente la identidad del nodo y el origen de los mensajes a partir de este valor.

Cada nodo de la red está identificado por un carácter alfanumérico: 0 para el coordinador, del 1 al 9 para los routers y una letra ASCII para los terminales.

En la propuesta se designa a los dispositivos finales como: Celula1, Celula2, Celula3, Celula4, Celula5 y Autómata.

DH (*Destination High*):

Valor que representa los 32 bits más significativos de la dirección de 64 bits del nodo destino de los mensajes transmitidos.

Siempre debe ser 0x00000000.

DL (*Destination Low*):

Son los 32 bits menos significativos del nodo destino de los mensajes transmitidos. Para el coordinador este valor debe ser la dirección *broadcast* (0x0000FFFF), y para todos los demás nodos debe ser la dirección por defecto del coordinador (0x00000000).

MY:

Dirección de 16 bits del nodo, asignada dinámicamente al unirse a la red.

En la propuesta se los asigna como C1, C2, C3, C4, C5 y AU.

Cuando el pulsador es presionado ingresa una señal baja (LOW) a los pines elegidos para generar la señal de radio frecuencia, la misma que es transmitida en modo API al módulo coordinador.

En la Figura 26, se indica el diseño electrónico de la conexión de los pulsadores a los pines 20 AD0/DI00 (calidad), 19 AD1/DI01 (mantenimiento), y 18 AD2/DI02 (supervisión).

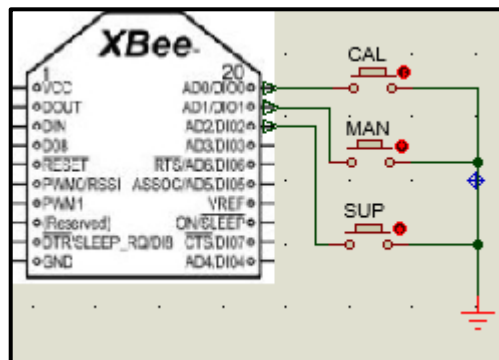


Figura 26: Conexión de pulsadores.
Realizado por: Noboa, 2017.

Recepción y procesamiento de datos.

Arduino Mega/XBee Shield.

El coordinador constituido por la tarjeta arduino mega y el XBee Shield, es el encargado de gestionar los datos recibidos desde los dispositivos finales (pulsadores) que conforman la red.

El puerto de comunicación serial, pines 0 (TX) y 1 (RX) gestionan la comunicación entre el XBee Shield y el Arduino que monitorean la red inalámbrica basada en RF.

Los puertos digitales del 2 al 7 son los encargados de gestionar los relés que a su vez comandan las luces giratorias.

Los puertos digitales del 22 al 39 activan los avisos luminosos del tablero.

Los sonidos son gestionados por una tarjeta MP3 conectada al puerto Serie 1, pines 18 (TX1) y 19 (RX1).

El servidor Wifi ESP 8266-01 donde se gestionan los datos del sistema estarán comandados por el puerto Serie 2, pines 16 (TX2) y 17 (RX2).

El ANEXO A, muestra en su totalidad el programa para el funcionamiento del sistema ANDON.

Diseño XBee coordinador.

Es el responsable de establecer el canal de comunicaciones, con los dispositivos finales. Una vez formada la red, el Coordinador participa en el enrutado de paquetes siendo el destinatario de la información.

La Figura 27, indica el flujograma de la recepción de datos enviados desde los dispositivos finales (pulsadores), proceso que consiste en establecer una relación de pertenencia de un dispositivo final dentro de la red, que después de haber enviado una trama API o código, esta será receptada por el coordinador quien decide que sonido y servicio reproduce, ya sea de calidad, mantenimiento o supervisión.

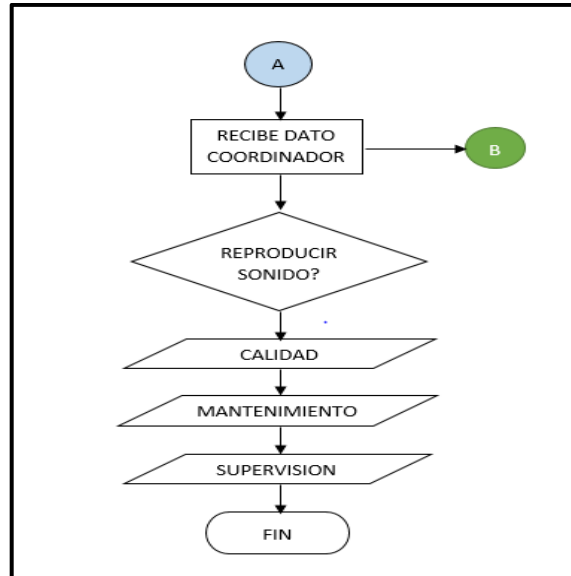


Figura 27: Flujograma de recepción de datos.
Realizado por: Noboa, 2017.

En esta etapa el dispositivo coordinador (XBee Shield / Arduino), permite el reconocimiento y comunicación con los demás dispositivos de la red, al recibir los datos; los decodifica y los envía a las salidas digitales pines 2 al 7.

En el ANEXO D, se indica un ejemplo de la trama recibida por el coordinador desde el dispositivo final C1 (célula 1).

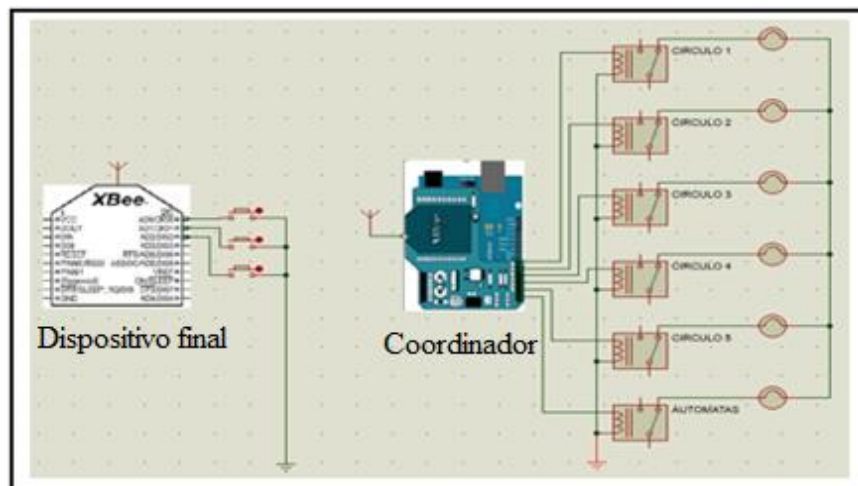


Figura 28: Configuración de XBee como coordinador.
Realizado por: Noboa, 2017.

En la figura 28, se indica la conexión del coordinador a los relés que accionan las luces giratorias.

La Tabla 5, indica los parámetros de configuración del XBee COORDINADOR.

Tabla 5: Configuración del coordinador.

PARAMETRO	COORDINADOR
CH	C
ID	3332
DH	0
DL	FFFF
MY	1
IR	0
CE	1

Fuente: Xctu.

Realizado por: Noboa, 2017.

Accionamiento de luces giratorias.

El encendido o apagado de las luces giratorias es controlado por los puertos de salida a los que se conectan los elementos actuadores (Relé RL1 a RL7), basado en el principio de funcionamiento de conexiones básicas eléctricas de accionamiento de una carga marcha y paro.

Cuando un dato es recibido desde una trama, el algoritmo la interpreta y según sea el caso, envía a sus salidas un nivel bajo (LOW) para habilitar el respectivo relé.

Hay que mencionar que el voltaje al cual trabajaran las luces giratorias es de 220 voltios en corriente alterna, por lo cual se optó usar relés para su control.

Se debe poner cuidado en la manipulación debido al alto voltaje de operación.

La Figura 29, indica el control de arduino al relé que activa una luz giratoria.

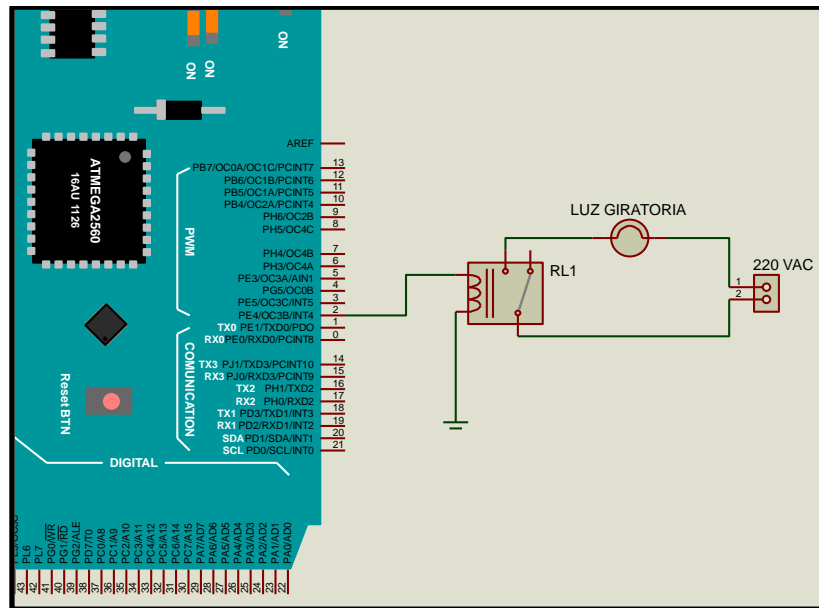


Figura 29: Configuración básica del accionamiento de luces.
Realizado por: Noboa, 2017.

La Figura 30, indica el diagrama eléctrico de la acometida de voltaje y el acondicionamiento de los módulos relé.

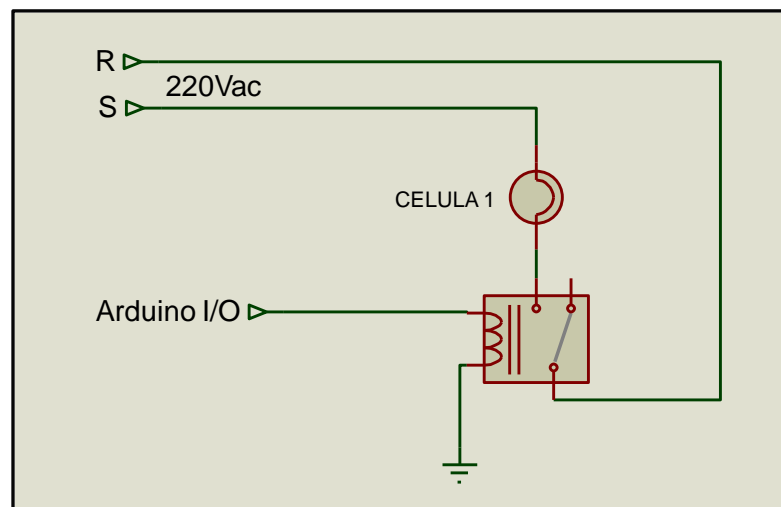


Figura 30: Diagrama Acometida 220vac del relé .
Realizado por: Noboa, 2017.

La Figura 31, indica el diagrama eléctrico para la instalación de las luces giratorias.

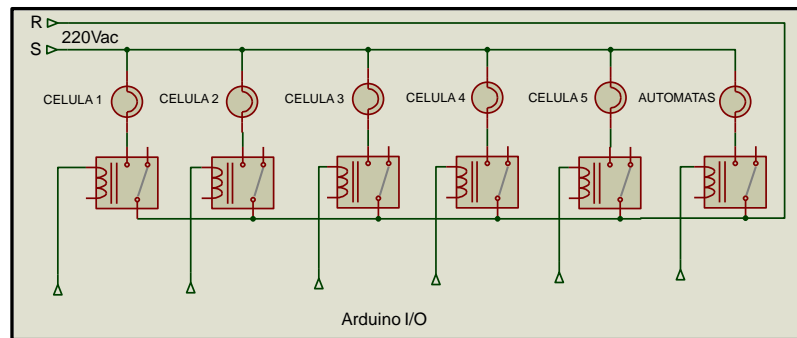


Figura 31: Diagrama instalación de luces giratorias.
Realizado por: Noboa, 2017.

Visualización.

Por la parte de visualización se construye un tablero que contiene la simbología y leyendas, las mismas que son iluminadas por un arreglo de led de alto brillo.

La Figura 32, indica un bosquejo del tablero y en el ANEXO G se encuentra el diagrama eléctrico de la conexión de los leds.

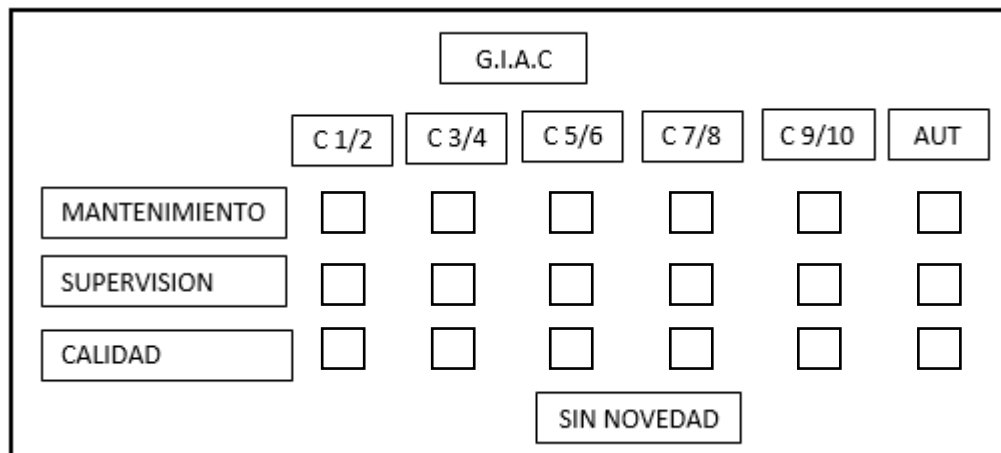


Figura 32: Apariencia tablero de visualización.
Realizado por: Noboa, 2017.

La Figura 33, indica la construcción del tablero de visualización, y su dimensionamiento el cual se elabora en madera triplex y es acabado con una capa de laca negra como fondo.

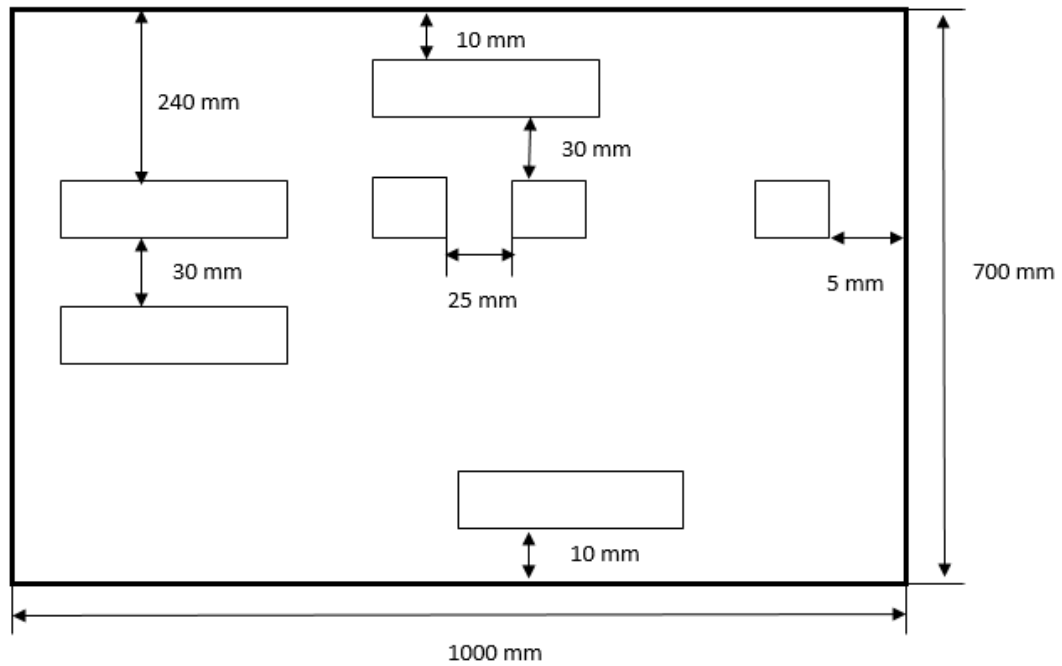


Figura 33: Dimensionamiento del tablero.

Realizado por: Noboa, 2017.

En la Figura 34, se indica los diseños en acrílico, y las dimensiones de todas las figuras y leyendas que contiene el tablero.

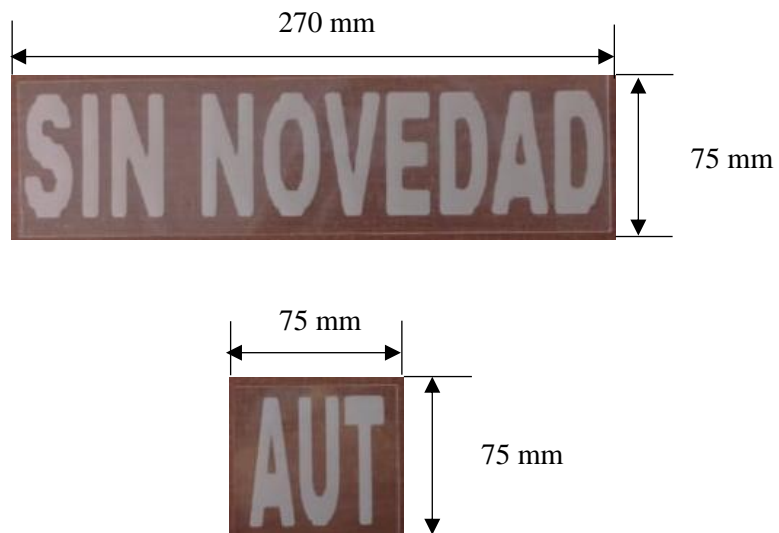


Figura 34: Diseño de avisos y leyendas.

Realizado por: Noboa, 2017.

Posteriormente se colocan todos los elementos en la posición elegida de acuerdo al diseño previo (Figura 32), para luego soldar y colocar los leds que iluminan los

diferentes módulos, de igual manera cada línea de entrada que recibe los datos desde el coordinador son conectados a terminales DB25 y DB9.

La Figura 35, indica el diseño final del tablero.



Figura 35: Diseño final del tablero.
Realizado por: Noboa, 2017.

Reproducción de melodías.

El sonido es gestionado por una tarjeta MP3 de la serie WT9501M03, soporta hasta 320 kbps de archivos de audio, los que son almacenados en una tarjeta de memoria micro SD.

El control se lo puede realizar ya sea por teclado o comunicación serial mediante cualquier micro controlador, en nuestro caso Arduino.

La Figura 36, indica el aspecto físico y la disposición de pines de la tarjeta MP3.

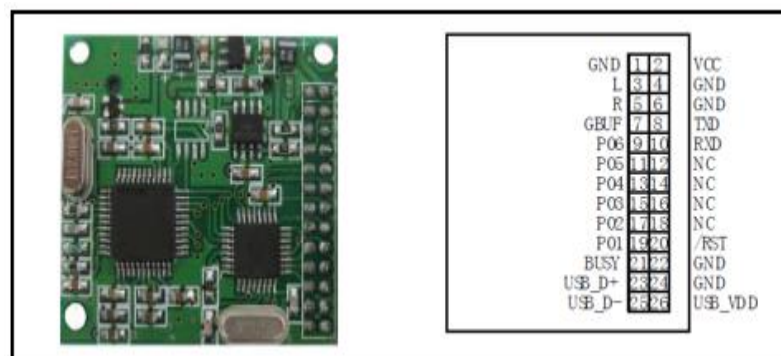


Figura 36: Disposición de pines MP3.
Realizado por: Noboa, 2017.

La Figura 37, indica la conexión básica de un microcontrolador por puerto serial con la tarjeta MP3 (WT9601M03) y la salida al amplificador.

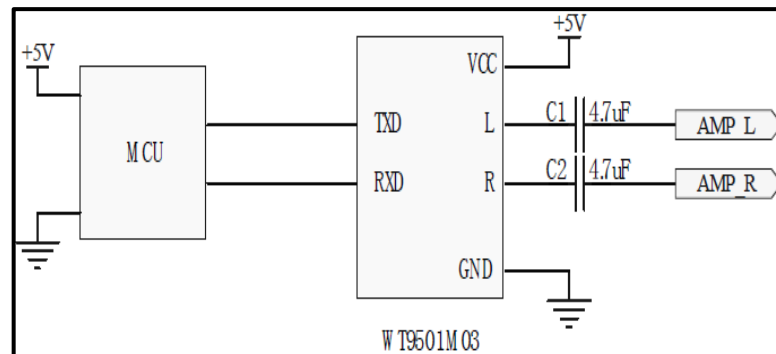


Figura 37: Comunicación serial con WT9601M03.
Realizado por: Noboa, 2017

El dispositivo se conecta a los pines 18 (TX1) y 19 (RX1) del puerto serie de Arduino.

Los comandos para reproducir una melodía tienen la siguiente trama que viene sugerida por el fabricante:

7E	07	A0	30	30	30	30	31	7E
----	----	----	----	----	----	----	----	----

En el ANEXO C, se indica los comandos para la programación del dispositivo.

Registro de datos.

En la propuesta se diseña una página web utilizando lenguaje de programación HTML (Lenguaje de Marcado para Hipertextos o Híper Text Markup Language) en la que se observa los datos enviados desde Arduino al accionar un pulsador, dando la posibilidad de que toda la planta pueda tener información de lo ocurrido vía Wifi en tiempo real.

Dicha página web es programada en el módulo Wifi ESP8266-01, el mismo que se lo ha configurado como AP o Access points o punto de acceso a la red, a la que se puede acceder al ingresar en el navegador la dirección 192.168.4.1.

Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia o presentaciones de una página web.

También se hace uso de JavaScript que es un robusto lenguaje de programación que puede ser aplicado a un documento HTML y usado para crear interactividad dinámica en los sitios web.

Algunos de los códigos usados para nuestra página HTML, se describen a continuación.

<HTML>: define el inicio del documento HTML, le indica al navegador que lo que viene a continuación debe ser interpretado como código HTML.

<script>: incrusta un script en una web.

<head>: define la cabecera del documento HTML.

<title>: define el título de la página. Por lo general, el título aparece en la barra de título encima de la ventana.

<body>: define el contenido principal o cuerpo del documento. Esta es la parte del documento HTML que se muestra en el navegador; dentro de esta etiqueta pueden definirse propiedades comunes a toda la página, como color de fondo y márgenes.

<table>: define una tabla.

La mayoría de etiquetas deben cerrarse como se abren, pero con una barra (</>).

El ANEXO B, contiene la página HTML y el programa creado para el registro de datos.

Comunicación serial entre ESP8266-01 y mega2560.

Cuando un pulsador es activado el dato recibido es enviado al puerto Serial2 del arduino el mismo que se encuentra en los pines TX2 (PIN 16) y el RX2 (PIN 17), y se conecta con el único puerto serie del ESP8266-01 a los pines RX y TX respectivamente.

El puerto serial estará configurado a una velocidad de 115200 baudios, para agilizar los tiempos de respuestas en el intercambio de información Arduino – ESP8266.

La Figura 38, indica el diagrama de la conexión serial ESP8266 – Arduino.

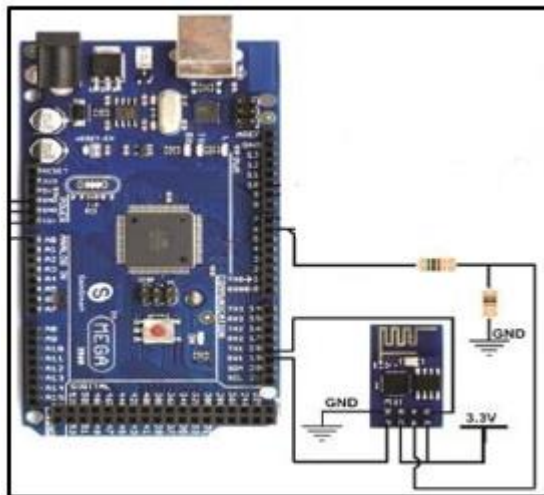


Figura 38: Diagrama del circuito de almacenamiento de datos.
Realizado por: Noboa, 2017.

La Figura 39, indica como es la comunicación, transmisión (TX) y recepción (RX), entre el módulo esp8266 y la tarjeta arduino mega 25660.

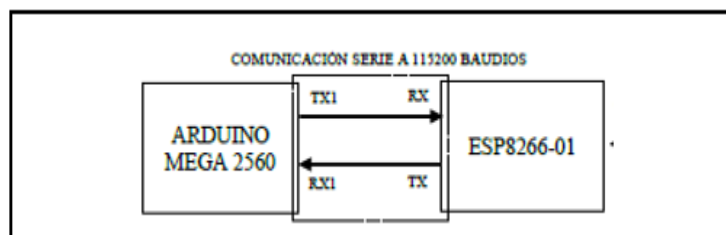


Figura 39: Comunicación serial entre esp8266-01 y mega2560.
Realizado por: Noboa, 2017.

Gracias al desarrollo de una librería para el módulo ESP es posible la configuración del mismo, utilizando el IDE de arduino.

El ANEXO E, indica el diagrama general de conexiones del sistema electrónico del sistema ANDON.

Resultados esperados.

Cuando se arranca el sistema electrónico entra en un bucle infinito, el cual adquiere información del estado de los pulsadores o dispositivos finales, los que envían información al Arduino para su tratamiento, es decir; ejecuta el programa interno y dependiendo del estado encenderán las luces giratorias y se escuchará una melodía.

En ese instante los datos generados se enviarán al servidor y mediante una dirección IP asignada previamente se puede acceder al registro de los mismos alojados en la página web.

El sistema siempre está a la espera de recibir una orden remota desde el centro de control, para ejecutar acciones en las salidas conectadas a los actuadores (relé RL1 a RL6), a la tarjeta MP3 y al servidor web.

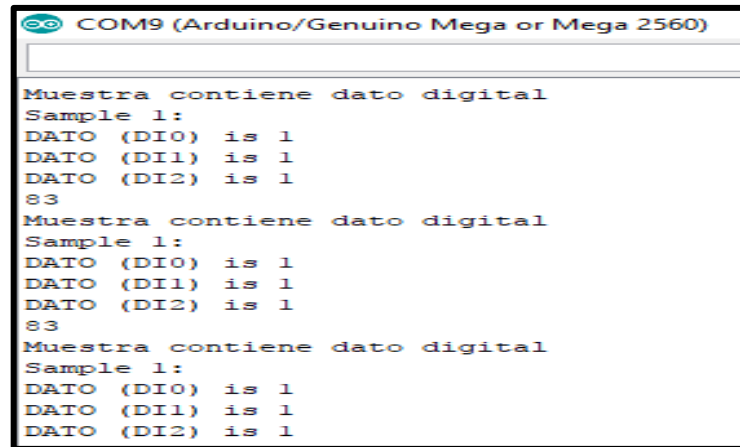
A continuación, se presentan ensayos realizados durante el proyecto que intervienen en la ejecución del sistema a fin de obtener resultados.

Encender luz giratoria.

Objetivo: Recibir una trama de datos y procesarla para saber qué célula está requiriendo un servicio.

Ensayo: Se conecta la salida 3 de Arduino hasta el módulo de relés como se indicó en la Figura 29, se aprieta un pulsador y por medio del monitor serial se observa que los datos o tramas son capturados por Arduino.

La Figura 40, indica los datos recibidos por puerto serial en el Coordinador.



```
COM9 (Arduino/Genuino Mega or Mega 2560)

Muestra contiene dato digital
Sample 1:
DATO (DI0) is 1
DATO (DI1) is 1
DATO (DI2) is 1
83
Muestra contiene dato digital
Sample 1:
DATO (DI0) is 1
DATO (DI1) is 1
DATO (DI2) is 1
83
Muestra contiene dato digital
Sample 1:
DATO (DI0) is 1
DATO (DI1) is 1
DATO (DI2) is 1
83
```

Figura 40: Visualización de datos en puerto serial.
Fuente: Software Arduino 2018.

La Figura 41, indica un segmento de código que ejecuta el encendido de la luz giratoria mediante el accionamiento del relé respectivo.



```
ANDON_XBee Arduino 1.8.5
Archivo Editar Programa Herramientas Ayuda

ANDON_XBee $
//Cual CELULA requiere servicio MANTENIMIENTO
int celula = ioSample.getRemoteAddress16();

switch (celula){//Lectura parametro MY

case 193://C1 CELULA 1
c1();digitalWrite(2,HIGH);
break;

case 194://C2 CELULA 2
c2();digitalWrite(3,HIGH);
break;

case 195://C3 CELULA 3
c3();digitalWrite(4,HIGH);
break;

case 196://C4 CELULA 4
c4();digitalWrite(5,HIGH);
break;

case 197://C5 CELULA 5
c5();digitalWrite(6,HIGH);
break;

case 161://A1 AUTOMATAS
a1();digitalWrite(7,HIGH);
break;
}
}
```

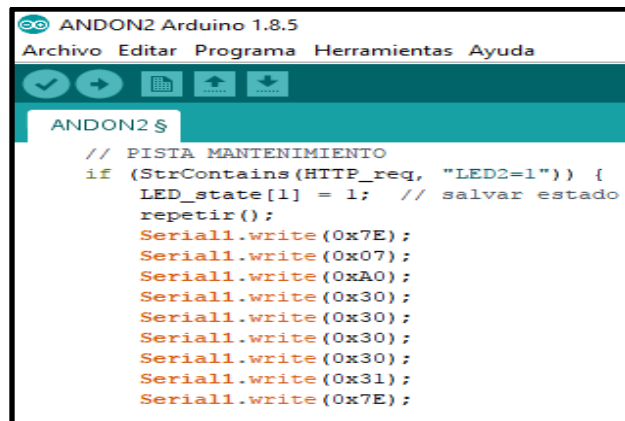
Figura 41: Código para la activación del relé.
Fuente: Software Arduino 2018.

Reproducir una pista de audio.

Objetivo: Recibir una trama de datos y procesarla para saber qué servicio es requerido mediante una pista de audio específica.

Ensayo: Se conecta los pines 18 y 19 de Arduino hasta el módulo MP3, se aprieta un pulsador y se observa que las tramas son capturadas por Arduino y procesadas.

La reproducción de la pista se efectúa de acuerdo al siguiente segmento de código como en la Figura 42.

A screenshot of the Arduino IDE interface. The title bar reads 'ANDON2 Arduino 1.8.5'. The menu bar includes 'Archivo', 'Editar', 'Programa', 'Herramientas', and 'Ayuda'. Below the menu bar is a toolbar with icons for opening, saving, and running. The main text area shows a code snippet for a maintenance track. The code is as follows:

```
ANDON2 $  
  
// PISTA MANTENIMIENTO  
if (StrContains(HTTP_req, "LED2=1")) {  
  LED_state[1] = 1; // salvar estado  
  repetir();  
  Serial1.write(0x7E);  
  Serial1.write(0x07);  
  Serial1.write(0xA0);  
  Serial1.write(0x30);  
  Serial1.write(0x30);  
  Serial1.write(0x30);  
  Serial1.write(0x30);  
  Serial1.write(0x31);  
  Serial1.write(0x7E);  
}
```

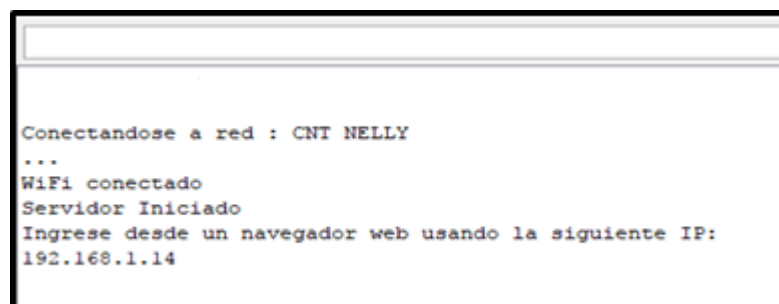
Figura 42: Código para la reproducción de la pista de audio para mantenimiento.

Fuente: Software Arduino 2018.

Página web de datos.

Objetivo: Implementar una página web dentro del módulo Wifi ESP8266-01

Ensayo: Se conecta el módulo servidor (ESP8266) al puerto serie2 pines 16 y 17 del Arduino, luego abriendo el navegador ingresa la dirección 192.168.4.1., obteniéndose por monitor serial lo indicado en la Figura 43.

A screenshot of a serial monitor window. It displays the following text:

```
Conectandose a red : CNT NELLY  
...  
WiFi conectado  
Servidor Iniciado  
Ingrese desde un navegador web usando la siguiente IP:  
192.168.1.14
```

Figura 43: Conexión con una red doméstica.

Fuente: IDE Arduino 2018.

Criterio de Éxito.

Como se indica en la Figura 44, cuando un pulsador es activado el XBee coordinador envía un dato a los relés a través del Arduino encendiendo las luces giratorias.



Figura 44: Encendido de luces giratorias.

Realizado por: Noboa, 2017.

Como se indica en la Figura 45, el tablero de información, muestra el aviso del servicio requerido, y la célula que lo solicita cuyo dato es enviado desde el Arduino a través del cable con los conectores DB9 y DB25.

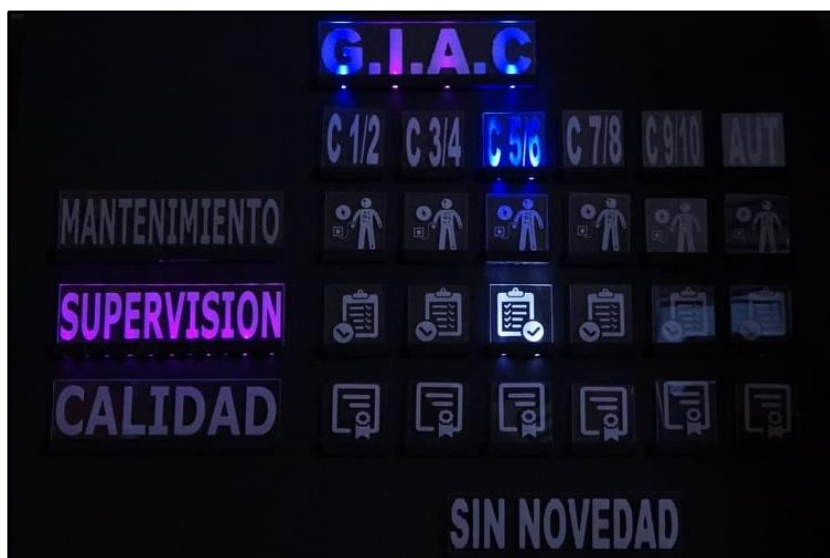


Figura 45: Encendido del tablero de información.
Realizado por: Noboa, 2017

Cuando se pulsa un botón del mando inalámbrico, un dato se envía al servidor (ESP8266), activando un contador interno que monitorea el número de paros, al mismo instante un temporizador entra en marcha.

La página web al ser dinámica puede presentar mediante un símbolo que simula un led encendido, el servicio y la célula que requiere atención. El registro sigue en funcionamiento mientras el sistema no sea apagado, en cuyo caso tanto el contador de paros como el temporizador serán reseteados.

El registro de la página web, se ingresa manualmente al sistema SAP de la empresa para su archivo. La Figura 46, indica una captura de la página web.



Figura 46: Interfaz página web.
Realizado por: Noboa, 2017.

Con la implementación del sistema propuesto se observa que el porcentaje del problema de desperdicios de producción tiende a la baja de 0,95% a 0,87%, reduciendo así el promedio total de desperdicios a un 3,23% como se indica en la Figura 47.

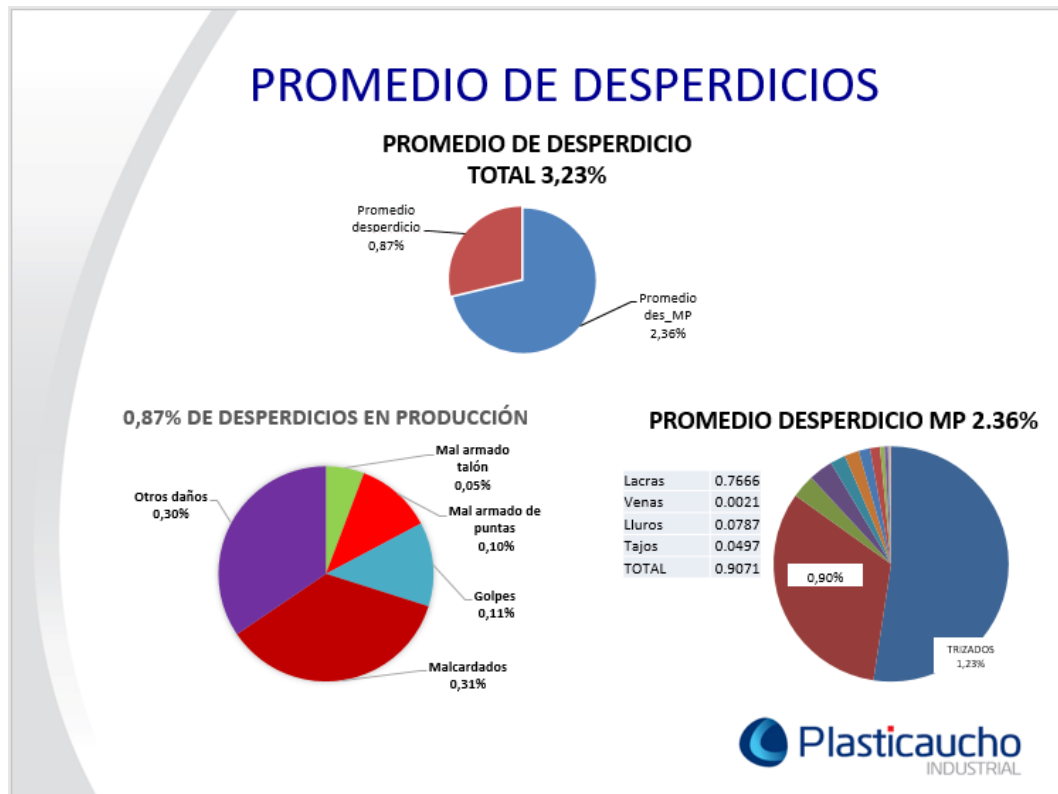


Figura 47: Promedio de desperdicios post instalación.
Realizado por: Noboa, 2017.

Por otro lado, los tiempos de paros del año 2017 desde el mes de octubre hasta diciembre y marzo del 2018, refleja una disminución de los tiempos de pérdidas en producción de 2 horas al pasar de 15,114 horas a 13,804 horas promedio, como lo indica la Figura 48, evidenciándose el objetivo del proyecto con la implementación del sistema ANDON.

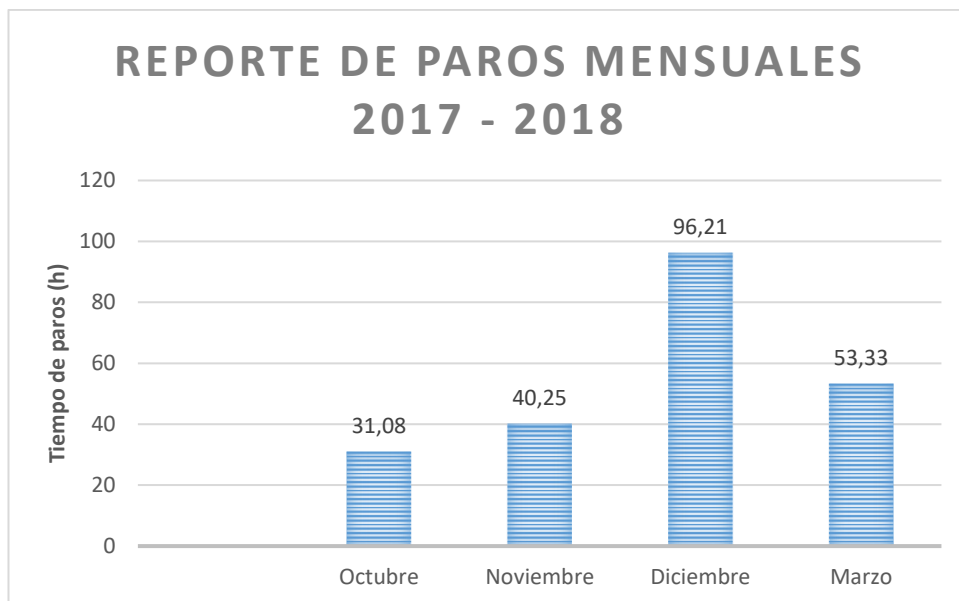


Figura 48: Reporte de paros mensuales 2017-2018 post instalación.

Realizado por: Noboa, 2017.

Análisis de costos.

El presupuesto del sistema electrónico para la adquisición y el envío de información, el estado de los pulsadores, el control de las luces y tablero, son detallados en la Tabla 6.

Tabla 6: Análisis de costos.

| SISTEMA ANDON | | | |
|------------------------|----------|------------|---------------|
| ELEMENTO | CANTIDAD | V.UNITARIO | V.TOTAL |
| Arduino Mega | 1 | 22 | 22 |
| ESP8266-01 | 1 | 8 | 8 |
| Módulo relé 4 canales | 2 | 12 | 24 |
| XBee S1 | 7 | 57 | 399 |
| Fuente de poder triple | 1 | 67 | 67 |
| XBee Shield | 1 | 15 | 15 |
| XBee PRO | 1 | 67 | 67 |
| Fuente 3.3 V | 7 | 7 | 49 |
| Luces giratorias | 7 | 20 | 140 |
| OTROS | | | 200 |
| TOTAL | | | \$ 991 |

Realizado por: Noboa, 2017.

Los valores de cada elemento son internos del país al por mayor, pero en mercados internacionales el precio reduciría radicalmente a precio de mayorista.

El rubro OTROS del presupuesto, son todos los elementos restantes como cables eléctricos de conexión entre sistema electrónico y tablero ANDON. También representa a los cables de conexiones interna del sistema electrónico, tornillos, luces led, caja plástica, pulsadores, etc.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

Conclusiones.

- El estudio del proceso de la producción del área de aparado en función de los datos suministrados por la empresa Plasticaucho, demuestran paros por diversos motivos entre los que se destacan, los fallos en las máquinas, el reinicio de las mismas como también el incumplimiento en el proceso de trabajo.
- De acuerdo al análisis de los datos, se establece que los factores como: fallos en la maquinaria ocasionados por mano de obra, la gestión tardía durante el proceso por parte de los departamentos de mantenimiento, supervisión y calidad; han ido mejorando, evidenciando la ayuda que brinda el sistema en el proceso.
- Con la implementación del sistema ANDON que basa su accionar en una alerta temprana de los imprevistos en el proceso de aparado central luego de la identificación de los pormenores del mismo se verifica una optimización en un 0,87% de los tiempos improductivos con tendencia a la baja paulatina conforme los diferentes operarios se vayan adaptando a esta tecnología aplicada.
- De las pruebas realizadas se determinó que el alcance de los módulos XBee, cumplen satisfactoriamente su propósito, manteniendo su señal estable, confiable y segura.

- Se establece una comunicación en modo Access Point (AP) con el módulo Wi-fi ESP8266-01 para el registro de datos del sistema.

Al utilizar señales de RF (radio frecuencia) de libre acceso en la banda de 2.4 GHz, aparatos tales como; teléfonos celulares, redes wifi, entre otros no afecta al sistema, consiguiendo una información y gestión sin pérdida de datos.

Recomendaciones.

- Se puede efectuar una actualización del sistema utilizando módulos XBee S2 por las prestaciones avanzadas que traen estas series.
- Ubicar los dispositivos a la mayor línea de vista posible para lograr una comunicación eficiente.
- Se debe proteger al sistema colocándolo en lugares secos y de incidencia baja de efectos electromagnéticos que pueden perturbar la comunicación.
- Aumentar la capacidad de las baterías de litio para una larga operación de los módulos pulsadores.
- Tener presente que la operación del XBee al igual que el ESP8266 es a 3.3. voltios en corriente continua, a mayor voltaje se ocasionará la avería de los mismos.
- Para tener un mejor control del registro se puede enlazar con una base de datos como por ejemplo MySQL entre otras.
- Profundizar el estudio de los nuevos modelos de producción como la Manufactura esbelta y su incidencia en los procesos de las industrias, de igual manera conceptos como el TPM o mantenimiento productivo total.

BIBLIOGRAFÍA

- **Administración de la producción-UNALMED. 2013.** Blogger. [En línea] 20 de Noviembre de 2013. http://andon2013.blogspot.com/2013/11/que-es-andon_20.html.
- **Blogger. 2013.** Blogspot. [En línea] 20 de Noviembre de 2013. <http://andon2013.blogspot.com/>.
- **Crespo, Enrique. 2014.** Aprendiendo Arduino. [En línea] 2014. <https://aprendiendoarduino.wordpress.com/category/ide/>.
- *El nuevo trabajo industrial: una mirada critica desde la calidad del empleo.* **Miglio, Martin, y otros. 2014.** 2014, SciELO, pág. 143.
- **Hernandez, Omar.** monografias.com. [En línea] <https://www.monografias.com/trabajos78/programacion-html/programacion-html.shtml>.
- **Llamas, Luis. 2017.** Ingenieria, infirmatica y diseño. [En línea] 27 de Mayo de 2017. <https://www.luisllamas.es/arduino-wifi-esp8266-esp01/>.
- *Mejoras de Lean Manufacturing en los sistemas productivos.* **Tejeda, Anne Sophie. 2011.** 2011, Redalyc, págs. 281-282.
- **PLASTICAUCHO INDUSTRIAL. 2016.** [En línea] 2016. http://www.plasticaucho.com.ec/nwp/resena_historica.php.
- **Rivadeneira, Christian y Cristian, Ligna. 2012.** *DISEÑO E IMPLEMENTACION DE UN SISTEMA AUTOMATIZADO DE AYUDA INMEDIATA EN EL PROCESO DE MANUFACTURA EN LA PLANTA DE ARNESES DE MUNDY HOME CIA.LTDA.* Quito, Pichincha, Ecuador : s.n., Noviembre de 2012.
- **Salazar, Bryan. 2016.** Herramientas para el Ingeniero Industrial. [En línea] 2016. <https://www.ingenieriaindustrialonline.com/herramientas-para-el-ingeniero-industrial/lean-manufacturing/andon-control-visual/>.
- **SENPLADES. 2013 - 2017.** [En línea] 2013 - 2017. <http://www.planificacion.gob.ec/wp-content/uploads/downloads/2015/11/Agenda-zona-3.pdf>.
- **Tejeda, Anne Sophie. 2011.** Redalyc. [En línea] 2 de Febrero de 2011. <http://www.redalyc.org/html/870/87019757005/>.
- **Thayer Ojeda, Luis.** ARDUINO.cl. [En línea] <http://arduino.cl/xbee-shield/>.

ANEXOS

ANEXO A: PROGRAMA ARDUINO MEGA (COORDINADOR XBee / CLIENTE WEB).

// PROGRAMA SISTEMA ANDON //

// Autor: Alex Noboa Castro 2018 //

// Inclusion de librerias para control del Xbee

#include <XBee.h>

XBee xbee = XBee();

Rx16IoSampleResponse ioSample = Rx16IoSampleResponse();

// Variables globales que cambian

int estadoLed = LOW; // estado actual del pin de salida CALIDAD

int estadoPulsador = LOW; // lee el estado actual del pin de entrada

int ultimoEstadoPuls = LOW; // lee el estado anterior del pulsador

int estadoLed1 = LOW; // estado actual del pin de salida MANTENIMIENTO

int estadoPulsador1 = LOW; // lee el estado actual del pin de entrada

int ultimoEstadoPuls1 = LOW; // lee el estado anterior del pulsador

int estadoLed2 = LOW; // estado actual del pin de salida SUPERVISION

int estadoPulsador2 = LOW; // lee el estado actual del pin de entrada

int ultimoEstadoPuls2 = LOW; // lee el estado anterior del pulsador

// Las siguientes variables globales son de tipo long.

// Sirve para evitar rebotes al recibir un dato.

long ultimoTiempoRebote = 0; // la última vez que el pin de salida CALIDAD cambia de estado.

long retardoRebote = 100; // El tiempo de rebote; aumentar si los parpadeos de salida son superiores.

long ultimoTiempoRebote1 = 0; // la última vez que el pin de salida MANTENIMIENTO cambia de estado.

long retardoRebote1 = 100; // El tiempo de rebote; aumentar si los parpadeos de salida son superiores.

long ultimoTiempoRebote2 = 0; // la última vez que el pin de salida SUPERVISION cambia de estado.

long retardoRebote2 = 100; // El tiempo de rebote; aumentar si los parpadeos de salida son superiores.

// Inicializacion de entradas y salidas de arduino

void setup() {

// Configuracion de puertos y velocidad de transmision

Serial.begin(9600);

xbee.setSerial(Serial); // Comunicacion Xbee puerto serial 0

Serial1.begin(9600); // Comunicacion MP3 puerto serial 1

Serial2.begin(115200); // Comunicacion ESP8266 puerto serial 2

// Entradas para lectura de pulsadores de los dispositivos finales

```

pinMode(8,INPUT);           // Pulsador calidad
pinMode(9,INPUT);           // Pulsador mantenimiento
pinMode(10,INPUT);          // Pulsador supervision

// Salidas para control de reles luces giratorias
pinMode(2,OUTPUT);digitalWrite(2,HIGH); //Celula 1/2
pinMode(3,OUTPUT);digitalWrite(3,HIGH); //Celula 3/4
pinMode(4,OUTPUT);digitalWrite(4,HIGH); //Celula 5/6
pinMode(5,OUTPUT);digitalWrite(5,HIGH); //Celula 7/8
pinMode(6,OUTPUT);digitalWrite(6,HIGH); //Celula 9/10
pinMode(7,OUTPUT);digitalWrite(7,HIGH); //Automatas A1

// Salidas para control de tablero
pinMode(22,OUTPUT);         //figura C1/C2
pinMode(24,OUTPUT);         //figura C3/C4
pinMode(26,OUTPUT);         //figura C5/C6
pinMode(28,OUTPUT);         //figura C7/C8
pinMode(30,OUTPUT);         //figura C9/C10
pinMode(32,OUTPUT);         //figura AUT

pinMode(40,OUTPUT);digitalWrite(40,HIGH); //leyenda GIAC activada siempre
pinMode(42,OUTPUT);digitalWrite(42,HIGH); //leyenda SIN NOVEDAD activada al inicio

pinMode(44,OUTPUT);         //leyenda CALIDAD
pinMode(46,OUTPUT);         //leyenda MANTENIMIENTO
pinMode(48,OUTPUT);         //leyenda SUPERVISION

// MP3 apagado al inicio
Serial1.write(0x7E);
Serial1.write(0x02);
Serial1.write(0xA3);
Serial1.write(0x7E);

}

void loop() {
  // Lectura del dato enviado desde un dispositivo final Xbee (pulsadores).
  xbee.readPacket();
  if (xbee.getResponse().isAvailable()) {
    if (xbee.getResponse().getApiId() == RX_16_IO_RESPONSE) {
      xbee.getResponse().getRx16IoSampleResponse(ioSample);

      // Visualizacion del dato recibido en puerto serie del arduino.
      Serial.print("Recibiendo desde: ");
      Serial.println(ioSample.getRemoteAddress16(), DEC);

      // Segun sea el caso de recibir peticion de una celula
      switch (ioSample.getRemoteAddress16()){
        case (193): // Dato decimal que representa a C1 en hexadecimal
          cel1();   // Si es C1 vaya a subrutina CELULA 1.
          break;

        case (194): // Dato decimal que representa a C2 en hexadecimal
          cel2();   // Si es C2 vaya a subrutina CELULA 2.
          break;

        case (195): // Dato decimal que representa a C3 en hexadecimal
          cel3();   // Si es C3 vaya a subrutina CELULA 3.

```

```

break;

case (196): // Dato decimal que representa a C4 en hexadecimal
cel4();    // Si es C4 vaya a subrutina CELULA 4.
break;

case (197): // Dato decimal que representa a C5 en hexadecimal
cel5();    // Si es C5 vaya a subrutina CELULA 5.
break;

case (161): // Dato decimal que representa a A1 en hexadecimal
A1();      // Si es A1 vaya a subrutina AUTOMATA A1.
break;
}
}
}
}

//***** CELULA 1 *****
void cel1()
{
    // Realiza un barrido de los pulsadores
    c1C();    // Llama servicio calidad
    c1M();    // Llama servicio mantenimiento
    c1S();    // Llama servicio supervision
}

// Servicio calidad
void c1C(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso=0;
    while (pulso==0){
        int leer = digitalRead(8);    // Lee puerto dato CALIDAD
        if (leer != ultimoEstadoPuls) {
            ultimoTiempoRebote = millis();
        }
        if ((millis() - ultimoTiempoRebote) > retardoRebote) {

            // Si el estado del pulsador ha cambiado
            if (leer != estadoPulsador) {
                estadoPulsador = leer;
                if (estadoPulsador == HIGH) {
                    estadoLed = !estadoLed;
                }
            }
        }
        digitalWrite(2, estadoLed);    // Enciende luz giratoria
        digitalWrite(22,!estadoLed);  // Enciende figura C1/C2
        digitalWrite(42,estadoLed);   // Apaga leyenda SIN NOVEDAD
        digitalWrite(44,!estadoLed);   // Enciende leyenda CALIDAD
        ultimoEstadoPuls = leer;        // Guarda el ultimo estado del pulsador

    // Si esta activa leyenda calidad
    if (digitalRead(44) == HIGH) {

        //Activa dato calidad del registro ESP8266
        Serial2.write('1');
    }
}

```

```

    // Activa MP3 melodía CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x31);
    Serial1.write(0x7E);

// Si no está activa leyenda
}else {

    // Desactiva dato calidad del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso=1;break;
}
}
}

// Servicio mantenimiento
void c1M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){
        int leer1 = digitalRead(9);           // Lee puerto dato MANTENIMIENTO
        if (leer1 != ultimoEstadoPuls1) {
            ultimoTiempoRebote1 = millis();
        }
        if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

            // Si el estado del pulsador ha cambiado
            if (leer1 != estadoPulsador1) {
                estadoPulsador1 = leer1;
                if (estadoPulsador1 == HIGH) {
                    estadoLed1 = !estadoLed1;
                }
            }
        }
        digitalWrite(2, estadoLed1);           // Enciende luz giratoria
        digitalWrite(22,!estadoLed1);         // Enciende figura C1/C2
        digitalWrite(42,estadoLed1);          // Apaga leyenda SIN NOVEDAD
        digitalWrite(46,!estadoLed1);         // Enciende leyenda MANTENIMIENTO
        ultimoEstadoPuls1 = leer1;            // Guarda el último estado del pulsador

// Si está activa leyenda MANTENIMIENTO
if (digitalRead(46) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266

```

```

Serial2.write('2');

// Activa MP3 melodía MANTENIMIENTO
Serial1.write(0x7E);
Serial1.write(0x07);
Serial1.write(0xA0);
Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x32);
Serial1.write(0x7E);

// Si no está activa leyenda
}else {

    // Desactiva dato mantenimiento del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso1=1;break;
}
}
}

// Servicio supervision
void c1S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso2=0;
    while (pulso2==0) {
        int leer2 = digitalRead(10);           // Lee puerto dato SUPERVISION
        if (leer2 != ultimoEstadoPuls2) {
            ultimoTiempoRebote2 = millis();
        }
        if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

            // Si el estado del pulsador ha cambiado:
            if (leer2 != estadoPulsador2) {
                estadoPulsador2 = leer2;
                if (estadoPulsador2 == HIGH) {
                    estadoLed2 = !estadoLed2;
                }
            }
        }
        digitalWrite(2, estadoLed2);           // Enciende luz giratoria
        digitalWrite(22,!estadoLed2);         // Enciende figura C1/C2
        digitalWrite(42,estadoLed2);          // Apaga leyenda SIN NOVEDAD
        digitalWrite(48,!estadoLed2);         // Enciende leyenda SUPERVISION
        ultimoEstadoPuls2 = leer2;             // Guarda el último estado del pulsador

    }
}

// Si está activa leyenda SUPERVISION

```



```

if (digitalRead(48) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
}else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso2=1;break;
}
}
}
//*****

//***** CELULA 2 *****

void cel2()
{
    // Realiza un barrido de los pulsadores
    c2C();           // Llama servicio calidad
    c2M();           // Llama servicio mantenimiento
    c2S();           // Llama servicio supervision
}

// Servicio calidad
void c2C(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso=0;
    while (pulso==0){
        int leer = digitalRead(8);           // Lee puerto dato CALIDAD
        if (leer != ultimoEstadoPuls) {
            ultimoTiempoRebote = millis();
        }
        if ((millis() - ultimoTiempoRebote) > retardoRebote) {

            // Si el estado del pulsador ha cambiado
            if (leer != estadoPulsador) {
                estadoPulsador = leer;
            }
        }
    }
}

```

```

    if (estadoPulsador == HIGH) {
        estadoLed = !estadoLed;
    }
}
digitalWrite(3, estadoLed);      // Enciende luz giratoria
digitalWrite(24, !estadoLed);    // Enciende figura C3/C4
digitalWrite(42, estadoLed);     // Apaga leyenda SIN NOVEDAD
digitalWrite(44, !estadoLed);    // Enciende leyenda CALIDAD
ultimoEstadoPuls = leer;        // Guarda el ultimo estado del pulsador

// Si esta activa leyenda calidad
if (digitalRead(44) == HIGH) {

    //Activa dato calidad del registro ESP8266
    Serial2.write('1');

    // Activa MP3 melodia CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x31);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
}else {

    // Desactiva dato calidad del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodia CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso=1;break;
}
}

// Servicio mantenimiento
void c2M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){
        int leer1 = digitalRead(9);      // Lee puerto dato MANTENIMIENTO
        if (leer1 != ultimoEstadoPuls1) {
            ultimoTiempoRebote1 = millis();
        }
        if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

            // Si el estado del pulsador ha cambiado

```

```

    if (leer1 != estadoPulsador1) {
        estadoPulsador1 = leer1;
        if (estadoPulsador1 == HIGH) {
            estadoLed1 = !estadoLed1;
        }
    }
}
digitalWrite(3, estadoLed1); // Enciende luz giratoria
digitalWrite(24, !estadoLed1); // Enciende figura C3/C4
digitalWrite(42, estadoLed1); // Apaga leyenda SIN NOVEDAD
digitalWrite(46, !estadoLed1); // Enciende leyenda MANTENIMIENTO
ultimoEstadoPuls1 = leer1; // Guarda el ultimo estado del pulsador

// Si esta activa leyenda MANTENIMIENTO
if (digitalRead(46) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x32);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato mantenimiento del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso1=1;break;
}
}
}

// Servicio supervision
void c2S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso2=0;
    while (pulso2==0) {
        int leer2 = digitalRead(10); // Lee puerto dato SUPERVISION
        if (leer2 != ultimoEstadoPuls2) {
            ultimoTiempoRebote2 = millis();
        }
        if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

```

```

// Si el estado del pulsador ha cambiado:
if (leer2 != estadoPulsador2) {
    estadoPulsador2 = leer2;
    if (estadoPulsador2 == HIGH) {
        estadoLed2 = !estadoLed2;
    }
}
}
digitalWrite(3, estadoLed2); // Enciende luz giratoria
digitalWrite(24, !estadoLed2); // Enciende figura C1/C2
digitalWrite(42, estadoLed2); // Apaga leyenda SIN NOVEDAD
digitalWrite(48, !estadoLed2); // Enciende leyenda SUPERVISION
ultimoEstadoPuls2 = leer2; // Guarda el ultimo estado del pulsador

// Si esta activa leyenda SUPERVISION
if (digitalRead(48) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso2=1; break;
}
}
}
//*****

//***** CELULA 3 *****

void cel3()
{
    // Realiza un barrido de los pulsadores
    c3C(); // Llama servicio calidad
    c3M(); // Llama servicio mantenimiento
}

```

```

c3S();           // Llama servicio supervision
}

// Servicio calidad
void c3C(){
  // Ciclo infinito hasta cuando int pulso cambie de estado.
  int pulso=0;
  while (pulso==0){
    int leer = digitalRead(8);           // Lee puerto dato CALIDAD
    if (leer != ultimoEstadoPuls) {
      ultimoTiempoRebote = millis();
    }
    if ((millis() - ultimoTiempoRebote) > retardoRebote) {

      // Si el estado del pulsador ha cambiado
      if (leer != estadoPulsador) {
        estadoPulsador = leer;
        if (estadoPulsador == HIGH) {
          estadoLed = !estadoLed;
        }
      }
    }
    digitalWrite(4, estadoLed);           // Enciende luz giratoria
    digitalWrite(26, estadoLed);          // Enciende figura C5/C6
    digitalWrite(42, estadoLed);          // Apaga leyenda SIN NOVEDAD
    digitalWrite(44, estadoLed);          // Enciende leyenda CALIDAD
    ultimoEstadoPuls = leer;              // Guarda el ultimo estado del pulsador

    // Si esta activa leyenda calidad
    if (digitalRead(44) == HIGH) {

      //Activa dato calidad del registro ESP8266
      Serial2.write('1');

      // Activa MP3 melodia CALIDAD
      Serial1.write(0x7E);
      Serial1.write(0x07);
      Serial1.write(0xA0);
      Serial1.write(0x30);
      Serial1.write(0x30);
      Serial1.write(0x30);
      Serial1.write(0x30);
      Serial1.write(0x30);
      Serial1.write(0x31);
      Serial1.write(0x7E);

      // Si no esta activa leyenda
    }else {

      // Desactiva dato calidad del registro ESP8266
      Serial2.write('0');

      // Desactiva MP3 melodia CALIDAD
      Serial1.write(0x7E);
      Serial1.write(0x02);
      Serial1.write(0xA3);
      Serial1.write(0x7E);

      // Sale del ciclo infinito
    }
  }
}

```

```

        pulso=1;break;
    }
}

// Servicio mantenimiento
void c3M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){
        int leer1 = digitalRead(9);           // Lee puerto dato MANTENIMIENTO
        if (leer1 != ultimoEstadoPuls1) {
            ultimoTiempoRebote1 = millis();
        }
        if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

            // Si el estado del pulsador ha cambiado
            if (leer1 != estadoPulsador1) {
                estadoPulsador1 = leer1;
                if (estadoPulsador1 == HIGH) {
                    estadoLed1 = !estadoLed1;
                }
            }
        }
        digitalWrite(4, estadoLed1);           // Enciende luz giratoria
        digitalWrite(26, estadoLed1);          // Enciende figura C5/C6
        digitalWrite(42, estadoLed1);          // Apaga leyenda SIN NOVEDAD
        digitalWrite(46, estadoLed1);          // Enciende leyenda MANTENIMIENTO
        ultimoEstadoPuls1 = leer1;             // Guarda el ultimo estado del pulsador

        // Si esta activa leyenda MANTENIMIENTO
        if (digitalRead(46) == HIGH) {

            //Activa dato mantenimiento del registro ESP8266
            Serial2.write('2');

            // Activa MP3 melodía MANTENIMIENTO
            Serial1.write(0x7E);
            Serial1.write(0x07);
            Serial1.write(0xA0);
            Serial1.write(0x30);
            Serial1.write(0x30);
            Serial1.write(0x30);
            Serial1.write(0x30);
            Serial1.write(0x30);
            Serial1.write(0x32);
            Serial1.write(0x7E);

            // Si no esta activa leyenda
        }else {

            // Desactiva dato mantenimiento del registro ESP8266
            Serial2.write('0');

            // Desactiva MP3 melodía MANTENIMIENTO
            Serial1.write(0x7E);
            Serial1.write(0x02);
            Serial1.write(0xA3);
            Serial1.write(0x7E);
        }
    }
}

```

```

    // Sale del ciclo infinito
    pulso1=1;break;
}
}
}

// Servicio supervision
void c1S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso2=0;
    while (pulso2==0) {
        int leer2 = digitalRead(10);           // Lee puerto dato SUPERVISION
        if (leer2 != ultimoEstadoPuls2) {
            ultimoTiempoRebote2 = millis();
        }
        if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

            // Si el estado del pulsador ha cambiado:
            if (leer2 != estadoPulsador2) {
                estadoPulsador2 = leer2;
                if (estadoPulsador2 == HIGH) {
                    estadoLed2 = !estadoLed2;
                }
            }
        }
        digitalWrite(4, estadoLed2);           // Enciende luz giratoria
        digitalWrite(26, estadoLed2);          // Enciende figura C5/C6
        digitalWrite(42, estadoLed2);          // Apaga leyenda SIN NOVEDAD
        digitalWrite(48, estadoLed2);          // Enciende leyenda SUPERVISION
        ultimoEstadoPuls2 = leer2;             // Guarda el ultimo estado del pulsador
    }
}

// Si esta activa leyenda SUPERVISION
if (digitalRead(48) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
}

```

```

Serial1.write(0x02);
Serial1.write(0xA3);
Serial1.write(0x7E);

// Sale del ciclo infinito
pulso2=1;break;
}
}
}
//*****
//***** CELULA 4 *****
void cel4()
{
// Realiza un barrido de los pulsadores
c4C();           // Llama servicio calidad
c4M();           // Llama servicio mantenimiento
c4S();           // Llama servicio supervision
}

// Servicio calidad
void c4C(){
// Ciclo infinito hasta cuando int pulso cambie de estado.
int pulso=0;
while (pulso==0){
int leer = digitalRead(8);           // Lee puerto dato CALIDAD
if (leer != ultimoEstadoPuls) {
ultimoTiempoRebote = millis();
}
if ((millis() - ultimoTiempoRebote) > retardoRebote) {

// Si el estado del pulsador ha cambiado
if (leer != estadoPulsador) {
estadoPulsador = leer;
if (estadoPulsador == HIGH) {
estadoLed = !estadoLed;
}
}
}
digitalWrite(5, estadoLed);           // Enciende luz giratoria
digitalWrite(28, estadoLed);          // Enciende figura C7/C8
digitalWrite(42, estadoLed);          // Apaga leyenda SIN NOVEDAD
digitalWrite(44, estadoLed);          // Enciende leyenda CALIDAD
ultimoEstadoPuls = leer;             // Guarda el ultimo estado del pulsador

// Si esta activa leyenda calidad
if (digitalRead(44) == HIGH) {

//Activa dato calidad del registro ESP8266
Serial2.write('1');

// Activa MP3 melodía CALIDAD
Serial1.write(0x7E);
Serial1.write(0x07);
Serial1.write(0xA0);
Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x30);
}
}
}

```



```

Serial1.write(0x31);
Serial1.write(0x7E);

// Si no esta activa leyenda
}else {

    // Desactiva dato calidad del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodia CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso=1;break;
}
}
}

// Servicio mantenimiento
void c4M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){
        int leer1 = digitalRead(9);          // Lee puerto dato MANTENIMIENTO
        if (leer1 != ultimoEstadoPuls1) {
            ultimoTiempoRebote1 = millis();
        }
        if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

            // Si el estado del pulsador ha cambiado
            if (leer1 != estadoPulsador1) {
                estadoPulsador1 = leer1;
                if (estadoPulsador1 == HIGH) {
                    estadoLed1 = !estadoLed1;
                }
            }
        }
        digitalWrite(5, estadoLed1);          // Enciende luz giratoria
        digitalWrite(28, !estadoLed1);         // Enciende figura C7/C8
        digitalWrite(42, estadoLed1);          // Apaga leyenda SIN NOVEDAD
        digitalWrite(46, !estadoLed1);         // Enciende leyenda MANTENIMIENTO
        ultimoEstadoPuls1 = leer1;             // Guarda el ultimo estado del pulsador
    }

    // Si esta activa leyenda MANTENIMIENTO
    if (digitalRead(46) == HIGH) {

        //Activa dato mantenimiento del registro ESP8266
        Serial2.write('2');

        // Activa MP3 melodia MANTENIMIENTO
        Serial1.write(0x7E);
        Serial1.write(0x07);
        Serial1.write(0xA0);
        Serial1.write(0x30);
        Serial1.write(0x30);
    }
}

```

```

Serial1.write(0x30);
Serial1.write(0x30);
Serial1.write(0x32);
Serial1.write(0x7E);

// Si no esta activa leyenda
}else {

    // Desactiva dato mantenimiento del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodia MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso1=1;break;
}
}
}

// Servicio supervision
void c4S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso2=0;
    while (pulso2==0) {
        int leer2 = digitalRead(10);           // Lee puerto dato SUPERVISION
        if (leer2 != ultimoEstadoPuls2) {
            ultimoTiempoRebote2 = millis();
        }
        if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

            // Si el estado del pulsador ha cambiado:
            if (leer2 != estadoPulsador2) {
                estadoPulsador2 = leer2;
                if (estadoPulsador2 == HIGH) {
                    estadoLed2 = !estadoLed2;
                }
            }
        }
        digitalWrite(5, estadoLed2);           // Enciende luz giratoria
        digitalWrite(28,!estadoLed2);          // Enciende figura C7/C8
        digitalWrite(42,estadoLed2);           // Apaga leyenda SIN NOVEDAD
        digitalWrite(48,!estadoLed2);          // Enciende leyenda SUPERVISION
        ultimoEstadoPuls2 = leer2;             // Guarda el ultimo estado del pulsador
    }
}

// Si esta activa leyenda SUPERVISION
if (digitalRead(48) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodia SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);

```

```

    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

// Si no esta activa leyenda
}else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso2=1;break;
}
}
}
//*****
//***** CELULA 5 *****
void cel5()
{
    // Realiza un barrido de los pulsadores
    c5C(); // Llama servicio calidad
    c5M(); // Llama servicio mantenimiento
    c5S(); // Llama servicio supervision
}

// Servicio calidad
void c5C(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso=0;
    while (pulso==0){
        int leer = digitalRead(8);           // Lee puerto dato CALIDAD
        if (leer != ultimoEstadoPuls) {
            ultimoTiempoRebote = millis();
        }
        if ((millis() - ultimoTiempoRebote) > retardoRebote) {

            // Si el estado del pulsador ha cambiado
            if (leer != estadoPulsador) {
                estadoPulsador = leer;
                if (estadoPulsador == HIGH) {
                    estadoLed = !estadoLed;
                }
            }
        }
    }
    digitalWrite(6, estadoLed);           // Enciende luz giratoria
    digitalWrite(30,!estadoLed);         // Enciende figura C9/C10
    digitalWrite(42,estadoLed);          // Apaga leyenda SIN NOVEDAD
    digitalWrite(44,!estadoLed);         // Enciende leyenda CALIDAD
}

```

```

ultimoEstadoPuls = leer;           // Guarda el ultimo estado del pulsador

// Si esta activa leyenda calidad
if (digitalRead(44) == HIGH) {

    //Activa dato calidad del registro ESP8266
    Serial2.write('1');

    // Activa MP3 melodia CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x31);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato calidad del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodia CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso=1;break;
}
}
}

// Servicio mantenimiento
void c5M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){
        int leer1 = digitalRead(9);           // Lee puerto dato MANTENIMIENTO
        if (leer1 != ultimoEstadoPuls1) {
            ultimoTiempoRebote1 = millis();
        }
        if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

            // Si el estado del pulsador ha cambiado
            if (leer1 != estadoPulsador1) {
                estadoPulsador1 = leer1;
                if (estadoPulsador1 == HIGH) {
                    estadoLed1 = !estadoLed1;
                }
            }
        }
    }
    digitalWrite(6, estadoLed1);           // Enciende luz giratoria
    digitalWrite(30, !estadoLed1);        // Enciende figura C9/C10
}

```

```

digitalWrite(42,estadoLed1);           // Apaga leyenda SIN NOVEDAD
digitalWrite(46,estadoLed1);           // Enciende leyenda MANTENIMIENTO
ultimoEstadoPuls1 = leer1;             // Guarda el ultimo estado del pulsador

// Si esta activa leyenda MANTENIMIENTO
if (digitalRead(46) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x32);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
}else {

    // Desactiva dato mantenimiento del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso1=1;break;
}
}
}

// Servicio supervision
void c5S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso2=0;
    while (pulso2==0) {
        int leer2 = digitalRead(10);           // Lee puerto dato SUPERVISION
        if (leer2 != ultimoEstadoPuls2) {
            ultimoTiempoRebote2 = millis();
        }
        if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

            // Si el estado del pulsador ha cambiado:
            if (leer2 != estadoPulsador2) {
                estadoPulsador2 = leer2;
                if (estadoPulsador2 == HIGH) {
                    estadoLed2 = !estadoLed2;
                }
            }
        }
    }
}

```

```

digitalWrite(6, estadoLed2);           // Enciende luz giratoria
digitalWrite(30, estadoLed2);          // Enciende figura C9/C10
digitalWrite(42, estadoLed2);          // Apaga leyenda SIN NOVEDAD
digitalWrite(48, estadoLed2);          // Enciende leyenda SUPERVISION
ultimoEstadoPuls2 = leer2;            // Guarda el ultimo estado del pulsador

// Si esta activa leyenda SUPERVISION
if (digitalRead(48) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
}else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso2=1;break;
}
}
}
//*****
//***** AUTOMATAS 1 *****
void A1()
{
    // Realiza un barrido de los pulsadores
    A1C();           // Llama servicio calidad
    A1M();           // Llama servicio mantenimiento
    A1S();           // Llama servicio supervision
}

// Servicio calidad
void A1C(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso=0;
    while (pulso==0){
        int leer = digitalRead(8);           // Lee puerto dato CALIDAD
        if (leer != ultimoEstadoPuls) {

```

```

    ultimoTiempoRebote = millis();
}
if ((millis() - ultimoTiempoRebote) > retardoRebote) {

    // Si el estado del pulsador ha cambiado
    if (leer != estadoPulsador) {
        estadoPulsador = leer;
        if (estadoPulsador == HIGH) {
            estadoLed = !estadoLed;
        }
    }
}
digitalWrite(7, estadoLed);           // Enciende luz giratoria
digitalWrite(32, !estadoLed);         // Enciende figura AUT
digitalWrite(42, estadoLed);          // Apaga leyenda SIN NOVEDAD
digitalWrite(44, !estadoLed);         // Enciende leyenda CALIDAD
ultimoEstadoPuls = leer;              // Guarda el ultimo estado del pulsador

// Si esta activa leyenda calidad
if (digitalRead(44) == HIGH) {

    //Activa dato calidad del registro ESP8266
    Serial2.write('1');

    // Activa MP3 melodía CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x31);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato calidad del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía CALIDAD
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso=1; break;
}
}

// Servicio mantenimiento
void A1M(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.
    int pulso1=0;
    while (pulso1==0){

```

```

int leer1 = digitalRead(9); // Lee puerto dato MANTENIMIENTO
if (leer1 != ultimoEstadoPuls1) {
    ultimoTiempoRebote1 = millis();
}
if ((millis() - ultimoTiempoRebote1) > retardoRebote1) {

    // Si el estado del pulsador ha cambiado
    if (leer1 != estadoPulsador1) {
        estadoPulsador1 = leer1;
        if (estadoPulsador1 == HIGH) {
            estadoLed1 = !estadoLed1;
        }
    }
}
digital Write (7, estadoLed1); // Enciende luz giratoria
digitalWrite(32, !estadoLed1); // Enciende figura AUT
digitalWrite(42, estadoLed1); // Apaga leyenda SIN NOVEDAD
digitalWrite(46, !estadoLed1); // Enciende leyenda MANTENIMIENTO
ultimoEstadoPuls1 = leer1; // Guarda el ultimo estado del pulsador

// Si esta activa leyenda MANTENIMIENTO
if (digitalRead(46) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x32);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
} else {

    // Desactiva dato mantenimiento del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía MANTENIMIENTO
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso1=1; break;
}
}
}

// Servicio supervision
void A1S(){
    // Ciclo infinito hasta cuando int pulso cambie de estado.

```



```

int pulso2=0;
while (pulso2==0) {
int leer2 = digitalRead(10);           // Lee puerto dato SUPERVISION
if (leer2 != ultimoEstadoPuls2) {
    ultimoTiempoRebote2 = millis();
}
if ((millis() - ultimoTiempoRebote2) > retardoRebote2) {

    // Si el estado del pulsador ha cambiado:
    if (leer2 != estadoPulsador2) {
        estadoPulsador2 = leer2;
        if (estadoPulsador2 == HIGH) {
            estadoLed2 = !estadoLed2;
        }
    }
}
digitalWrite(7, estadoLed2);           // Enciende luz giratoria
digitalWrite(32,!estadoLed2);          // Enciende figura AUT
digitalWrite(42,estadoLed2);           // Apaga leyenda SIN NOVEDAD
digitalWrite(48,!estadoLed2);          // Enciende leyenda SUPERVISION
ultimoEstadoPuls2 = leer2;            // Guarda el ultimo estado del pulsador

// Si esta activa leyenda SUPERVISION
if (digitalRead(46) == HIGH) {

    //Activa dato mantenimiento del registro ESP8266
    Serial2.write('2');

    // Activa MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x07);
    Serial1.write(0xA0);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x30);
    Serial1.write(0x33);
    Serial1.write(0x7E);

    // Si no esta activa leyenda
}else {

    // Desactiva dato supervision del registro ESP8266
    Serial2.write('0');

    // Desactiva MP3 melodía SUPERVISION
    Serial1.write(0x7E);
    Serial1.write(0x02);
    Serial1.write(0xA3);
    Serial1.write(0x7E);

    // Sale del ciclo infinito
    pulso2=1;break;
}
}
}
}
//*****FIN DEL PROGRAMA*****

```

ANEXO B: CONFIGURACIÓN MÓDULO ESP8266 (servidor página web).

//Registro de datos

//Desarrollo: Alex Noboa Castro.

//Este programa debe ser cargado en el modulo ESP8266.

#include <ESP8266WiFi.h>

#include <WiFiClient.h>

#include <ESP8266WebServer.h>

const char ssid[] = "SISTEMA ANDON"; //Definimos la SSDI de nuestro servidor
WiFi -nombre de red-

const char password[] = "Nobologic"; //Definimos la contraseña de nuestro
servidor

WiFiServer server(80); //Creamos el objeto server.

boolean LED_state[12] = {0};

int segundo=0;

int minuto=0;

int hora=0;

int segundo1=0;

int minuto1=0;

int hora1=0;

int segundo2=0;

int minuto2=0;

int hora2=0;

```

int cont=0,cont1=0,cont2=0;

void setup() {
  Serial.begin(115200);
  server.begin();           //inicializamos el servidor
  WiFi.mode(WIFI_AP);
  WiFi.softAP(ssid, password); //Red con clave, en el canal 1 y visible
}

WiFiClient client;
String HTTP_req;
void loop()
{
  client = server.available();

  if (client) { // got client?
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) { // dato del cliente
        char c = client.read(); // lectura 1 caracter del cliente
        HTTP_req += c; // salva requerimiento 1 caracter a la vez
        if (c == '\n' && currentLineIsBlank) {
          // Envio respuesta http
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: keep-alive");
          client.println();
          if (HTTP_req.indexOf("ajax_switch") > -1) {

```

```

        //Lectura del puerto
        GetSwitchState(client);
        SetLEDStates(client);

    }
    else
    {
        client.println("<!DOCTYPE html>");
        client.println("<html>");
        client.println("<head>");
        client.println("<title>SISTEMA ANDON</title>");
        client.println("<script>");
        client.println("var led1, x1, led2, x2, led3, x3, led4, x4;");
        client.println("function GetSwitchState() {");
        client.println("    nocache = led1 + led2 + led3 + led4 + \"&nocache=\\\"+
Math.random() * 1000000;");
        client.println("var request = new XMLHttpRequest();");
        client.println("request.onreadystatechange = function() {");
        client.println("    if (this.readyState == 4) {");
        client.println("        if (this.status == 200) {");
        client.println("            if (this.responseText != null) {");

            client.println("document.getElementById(\"switch_txt\").innerHTML =
this.responseText;");

            client.println("if (this.responseText.indexOf(\"LED1ON\") > -
1){ document.getElementById(\"LED1\").style.fill = \"yellow\";});");
            client.println("else { document.getElementById(\"LED1\").style.fill =
\"black\";});");

```

```

        client.println("if      (this.responseText.indexOf(\"LED2ON\")    >    -
1){document.getElementById(\"LED2\").style.fill = \"yellow\";});

        client.println("else    {document.getElementById(\"LED2\").style.fill    =
\"black\";});

        client.println("if      (this.responseText.indexOf(\"LED3ON\")    >    -
1){document.getElementById(\"LED3\").style.fill = \"yellow\";});

        client.println("else    {document.getElementById(\"LED3\").style.fill    =
\"black\";});

        client.println("if      (this.responseText.indexOf(\"LED4ON\")    >    -
1){document.getElementById(\"LED4\").style.fill = \"yellow\";});

        client.println("else    {document.getElementById(\"LED4\").style.fill    =
\"black\";});

        client.println("if      (this.responseText.indexOf(\"S1:ON\")      >    -
1){document.getElementById(\"SW1\").style.fill
=
\"red\";document.getElementById(\"text1\").innerHTML =\"CALIDAD\";});

        client.println("else    {document.getElementById(\"SW1\").style.fill    =
\"white\"; document.getElementById(\"text1\").innerHTML =\"
\";});

        client.println("if      (this.responseText.indexOf(\"S2:ON\")      >    -
1){document.getElementById(\"SW2\").style.fill          =          \"red\";
document.getElementById(\"text2\").innerHTML =\"MANTENIMIENTO\";});

        client.println("else    {document.getElementById(\"SW2\").style.fill    =
\"white\"; document.getElementById(\"text2\").innerHTML =\"
\";});

        client.println("if      (this.responseText.indexOf(\"S3:ON\")      >    -
1){document.getElementById(\"SW3\").style.fill          =          \"red\";
document.getElementById(\"text3\").innerHTML =\"SUPERVISION\";});

        client.println("else    {document.getElementById(\"SW3\").style.fill    =
\"white\"; document.getElementById(\"text3\").innerHTML =\"
\";});

client.println("} } }");

client.println("request.open(\"PUT\", \"ajax_switch\" + nocache, true);");
client.println("request.send(null);");
client.println("setTimeout('GetSwitchState()', 500);");
client.println("}");

```

```

client.println("function SetLEDStates(num){");

client.println("switch(num){");

client.println("case                                     1:
if(x1==1){led1=\"&LED1ON\";x1=0;}else{led1=\"&LED1OFF\";x1=1;}break;")
;

client.println("case                                     2:
if(x2==1){led2=\"&LED2ON\";x2=0;}else{led2=\"&LED2OFF\";x2=1;}break;")
;

client.println("case                                     3:
if(x3==1){led3=\"&LED3ON\";x3=0;}else{led3=\"&LED3OFF\";x3=1;}break;")
;

client.println("case                                     4:
if(x4==1){led4=\"&LED4ON\";x4=0;}else{led4=\"&LED4OFF\";x4=1;}break;")
;

client.println("case      0:      led1=\"&LED1OFF\";x1=1;
led2=\"&LED2OFF\";x2=1;      led3=\"&LED3OFF\";x3=1;
led4=\"&LED4OFF\";x4=1;break;");

client.println("} }");

client.println("</script>");

```

```

client.println("<style>");
client.println(".IO_box {");
client.println("height: 240px;");
client.println("margin: auto;");
client.println("border: 1px solid black;");
client.println("margin-top: 10px;");
client.println("width: 270px;");
client.println("}");
client.println("</style>");

```

```

client.println("</head>");

```

```

client.println("<body
onload=\"GetSwitchState();SetLEDStates(0);\"><center>");

client.println("<h1>Gestion Imprevistos Aparado Central</h1>");

client.println("<div id=\"text1\"></div>");

client.println("<svg width=\"40\" height=\"40\"><circle id=\"SW1\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"white\" stroke-width=\"4\"
fill=\"white\"/></svg>");

client.println("<div id=\"text2\"></div>");

client.println("<svg width=\"40\" height=\"40\"><circle id=\"SW2\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"white\" stroke-width=\"4\"
fill=\"white\"/></svg>");

client.println("<div id=\"text3\"></div>");

client.println("<svg width=\"40\" height=\"40\"><circle id=\"SW3\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"white\" stroke-width=\"4\"
fill=\"white\"/></svg>");

client.println("<p><button type=\"L1\"
onclick=\"SetLEDStates(1);\">C9/10</button>&nbsp;");

client.println("<button type=\"L2\"
onclick=\"SetLEDStates(2);\">C7/8</button>&nbsp;");

client.println("<button type=\"L3\"
onclick=\"SetLEDStates(3);\">C5/6</button>&nbsp;");

client.println("<button type=\"L4\"
onclick=\"SetLEDStates(4);\">C3/4</button>&nbsp;</p>");

client.println("<p><center>");

client.println("<svg width=\"50\" height=\"50\"><circle id=\"LED1\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"green\" stroke-width=\"4\"
fill=\"black\"/></svg>");

client.println("<svg width=\"50\" height=\"50\"><circle id=\"LED2\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"green\" stroke-width=\"4\"
fill=\"black\"/></svg>");

client.println("<svg width=\"50\" height=\"50\"><circle id=\"LED3\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"green\" stroke-width=\"4\"
fill=\"black\"/></svg>");

```

```

        client.println("<svg width=\"50\" height=\"50\"><circle id=\"LED4\"
cx=\"20\" cy=\"20\" r=\"10\"stroke=\"green\" stroke-width=\"4\"
fill=\"black\"/></svg>");

        client.println("<BR></BR>");

        client.println("<FONT SIZE= +3 COLOR=blue>REGISTRO DE
DATOS</font>");

        client.println("<p id=\"switch_txt\">Estado del Sistema:
Cargando...</p>");

        client.println("</center></p>");
        client.println("</center></body>");
        client.println("</html>");
    }

    // se muestra respuesta HTTP en el puerto serial
    Serial.print(HTTP_req);

    HTTP_req = "";

    break;
}

if (c == '\n')
{
    // el ultimo caracter empieza en linea nueva
    currentLineIsBlank = true;
}

else if (c != '\r')
{
    currentLineIsBlank = false;
}

}

}

delay(1);

// cierra la conexion
client.stop();

```



```

    Serial.println("client disonnected");
}
}

void GetSwitchState(WiFiClient cl)
{
    if (Serial.available()) { //Mientras la comunicacion serial este disponible...

        char inChar = Serial.read(); //Guardamos lo que llegue por el puerto serie en la
        variable inChar

        switch (inChar) { //Con este switch comparamos para asi actuar segun sea el
        caso...

//Datos recibidos desde Arduino

        case '1'://CALIDAD

            LED_state[0] = 1;

            cont++; // Activo contador

            break;

        case '2'://MANTENIMIENTO

            LED_state[1] = 1;

            cont1++; // Activo contador

            break;

        case '3'://SUPERVISION

            LED_state[2] = 1;

            cont2++; // Activo contador

            break;

        case '0'://Desactivo todos los servicios

            LED_state[0] = 0;Serial.read();

```

```

    LED_state[1] = 0;Serial.read();
    LED_state[2] = 0;Serial.read();

    break;

}

}

}

void SetLEDStates(WiFiClient cl)//Presento los cuadros de los registros
{
    cl.print("<?xml version = \"1.0\" ?>");
    cl.print("<div class=\"IO_box\">");
    if (LED_state[0]) { //Si esta activada la bandera
        cl.print("<FONT SIZE= +1 COLOR=blue>CALIDAD</font>");
        cl.println("<p></p>");
        cl.print("<FONT SIZE= +1 COLOR=blue>Tiempo de Paro</font>");
        cl.println("<p></p>");
        segundo+=4;
        if (segundo>59)
        {
            segundo=0;
            minuto++;
            if (minuto>59)
            {
                minuto=0;
                hora++;
                if (hora>23)
                {
                    hora=0;

```

```

    }
    }
    }
    cl.print(hora);
    cl.print(":");
    cl.print(minuto);
    cl.print(":");
    cl.println(segundo);
    cl.print("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=blue>Llamadas</font>");
    cl.println("<p></p>");
    cl.println(cont);
    cl.println("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=blue>Estado XBee</font>");
    cl.println("<p></p>");
    cl.println("S1:ON");
    cl.println("LED1ON");
} else
{
    cl.print("<FONT SIZE= +1 COLOR=blue>CALIDAD</font>");
    cl.println("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=blue>    </font>");
    cl.println("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=blue>Tiempo de Paro</font>");
    cl.println("<p></p>");
    cl.print(hora);
    cl.print(":");
    cl.print(minuto);
    cl.print(":");

```

```

cl.println(segundo);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=blue>Llamadas</font>");
cl.println("<p></p>");
cl.println(cont);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=blue>Estado XBee</font>");
cl.println("<p></p>");
cl.println("S1:OFF");
cl.println("LED1OFF");
cl.println("</div>");

}

cl.print("<div class=\"IO_box\">");
if (LED_state[1]) {
cl.print("<FONT SIZE= +1 COLOR=green>MANTENIMIENTO</font>");
cl.println("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=green>Tiempo de Paro</font>");
cl.println("<p></p>");
segundo1+=4;
    if (segundo1>59)
    {
segundo1=0;
        minuto1++;
        if (minuto1>59)
        {
minuto1=0;
            hora1++;
            if (hora1>23)

```

```

{
    hora1=0;
}
}
}

cl.print(hora1);
cl.print(":");
cl.print(minuto1);
cl.print(":");
cl.println(segundo1);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=green>Llamadas</font>");
cl.println("<p></p>");
cl.print(cont1);
cl.println("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=green>Estado XBee</font>");
cl.println("<p></p>");
cl.println("S2:ON");
cl.println("LED1ON");
}else
{
    cl.print("<FONT SIZE= +1 COLOR=green>MANTENIMIENTO</font>");
    cl.println("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=green>Tiempo de Paro</font>");
    cl.println("<p></p>");
    cl.print(hora1);
    cl.print(":");
    cl.print(minuto1);
    cl.print(":");

```

```

cl.println(segundo1);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=green>Llamadas</font>");
cl.println("<p></p>");
cl.print(cont1);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=green>Estado XBee</font>");
cl.println("<p></p>");
cl.println("S2:OFF");
cl.println("LED1OFF");
cl.print("</div>");

}

cl.print("<div class=\"IO_box\">");
if (LED_state[2]) {
cl.print("<FONT SIZE= +1 COLOR=orange>SUPERVISION</font>");
cl.println("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=orange>Tiempo de Paro</font>");
cl.println("<p></p>");
segundo2+=4;
    if (segundo2>59)
    {
segundo2=0;
        minuto2++;
        if (minuto2>59)
        {
minuto2=0;
            hora2++;
            if (hora2>23)

```

```

{
    hora2=0;
}
}
}

cl.print(hora2);
cl.print(":");
cl.print(minuto2);
cl.print(":");
cl.println(segundo2);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=orange>Llamadas</font>");
cl.println("<p></p>");
cl.print(cont2);
cl.println("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=orange>Estado XBee</font>");
cl.println("<p></p>");
cl.println("S3:ON");
cl.println("LED1ON");
}else
{
    cl.print("<FONT SIZE= +1 COLOR=orange>SUPERVISION</font>");
    cl.println("<p></p>");
    cl.print("<FONT SIZE= +1 COLOR=orange>Tiempo de Paro</font>");
    cl.println("<p></p>");
    cl.print(hora2);
    cl.print(":");
    cl.print(minuto2);
    cl.print(":");

```

```

cl.println(segundo2);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=orange>Llamadas</font>");
cl.println("<p></p>");
cl.print(cont2);
cl.print("<p></p>");
cl.print("<FONT SIZE= +1 COLOR=orange>Estado XBee</font>");
cl.println("<p></p>");
cl.println("S3:OFF");
cl.println("LED1OFF");
cl.print("</div>");
}
}

```

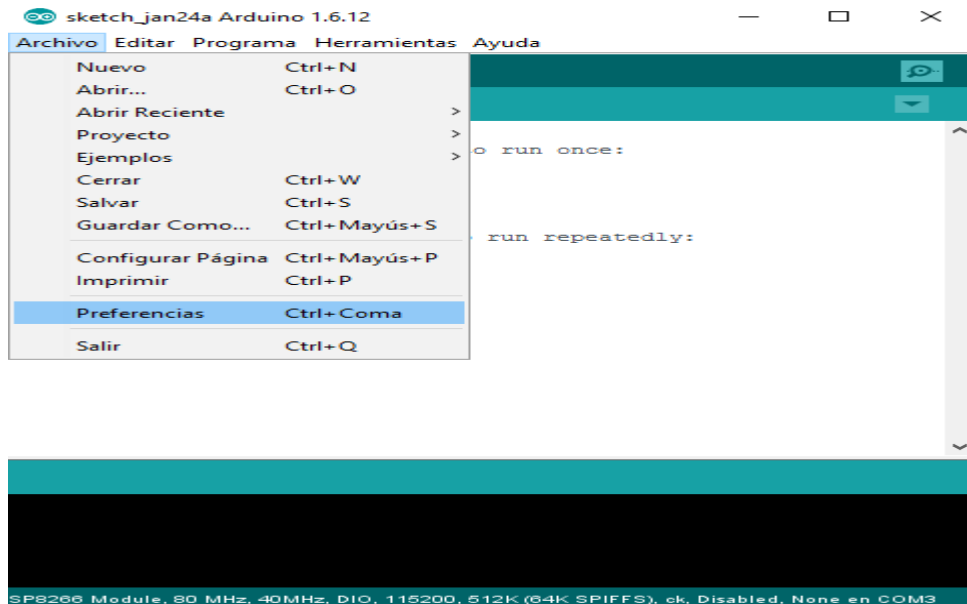
//-----FIN-----

Resultado del programa anterior.



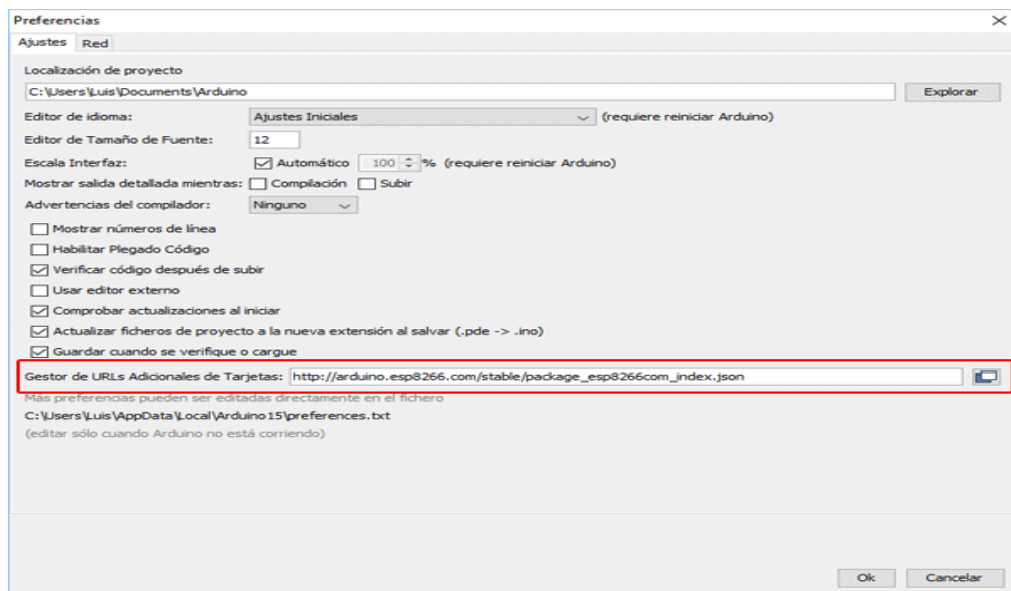
Programación del módulo ESP8266-01 con el IDE Arduino.

Abrir preferencias que se encuentran en *Archivo > Preferencias*



Poner el siguiente enlace:

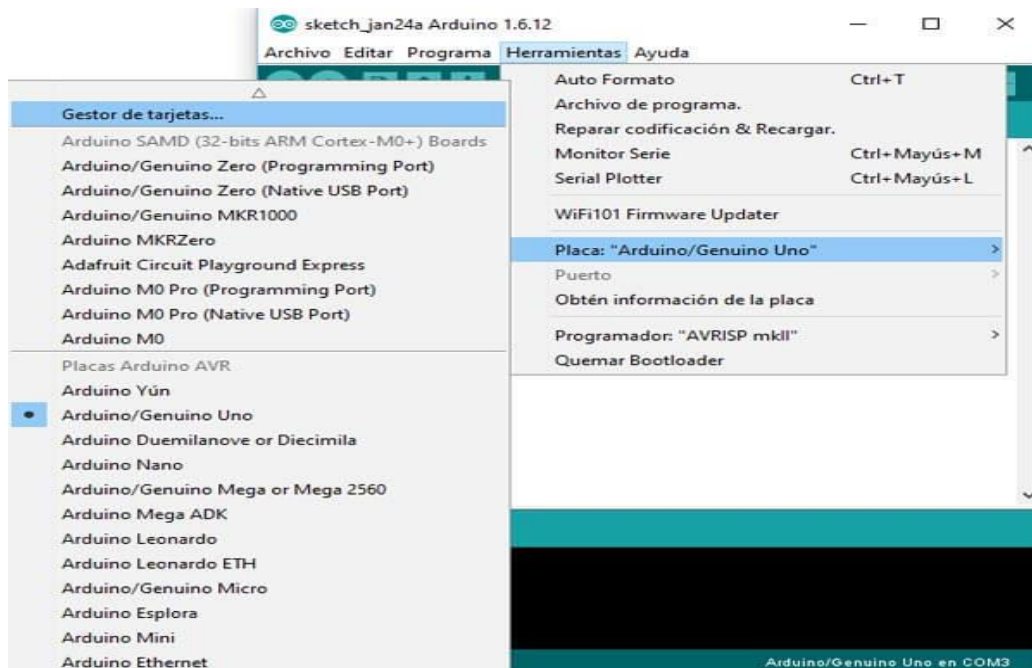
http://arduino.esp8266.com/stable/package_esp8266com_index.json



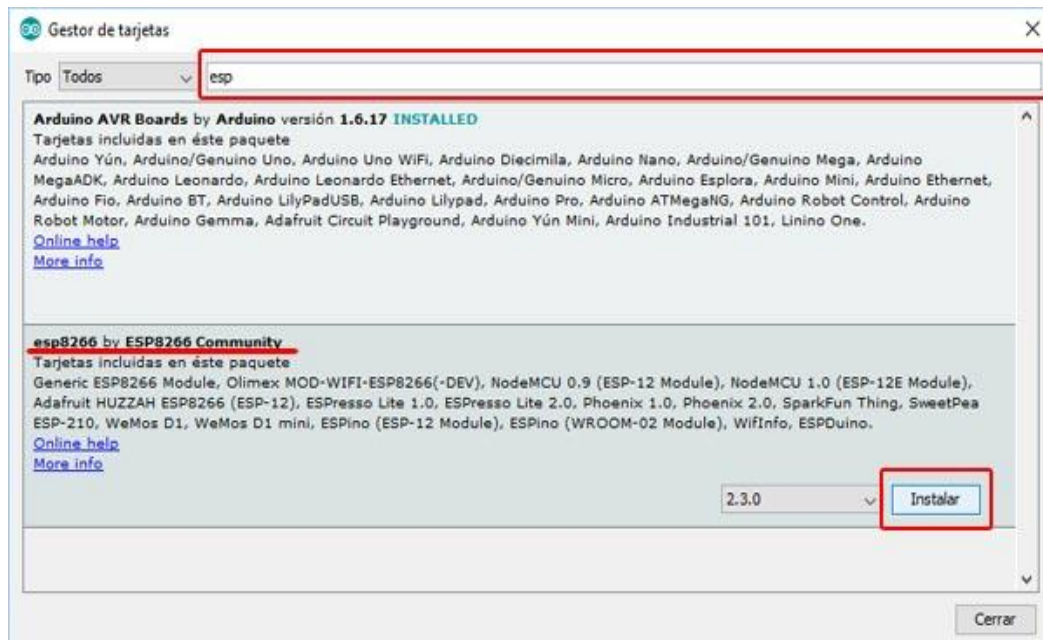
Click ok.

Paso 2: añadir los drivers del ESP8266

Ir a *Herramientas > Placa: "Arduino UNO" > Gestor de Tarjetas...*

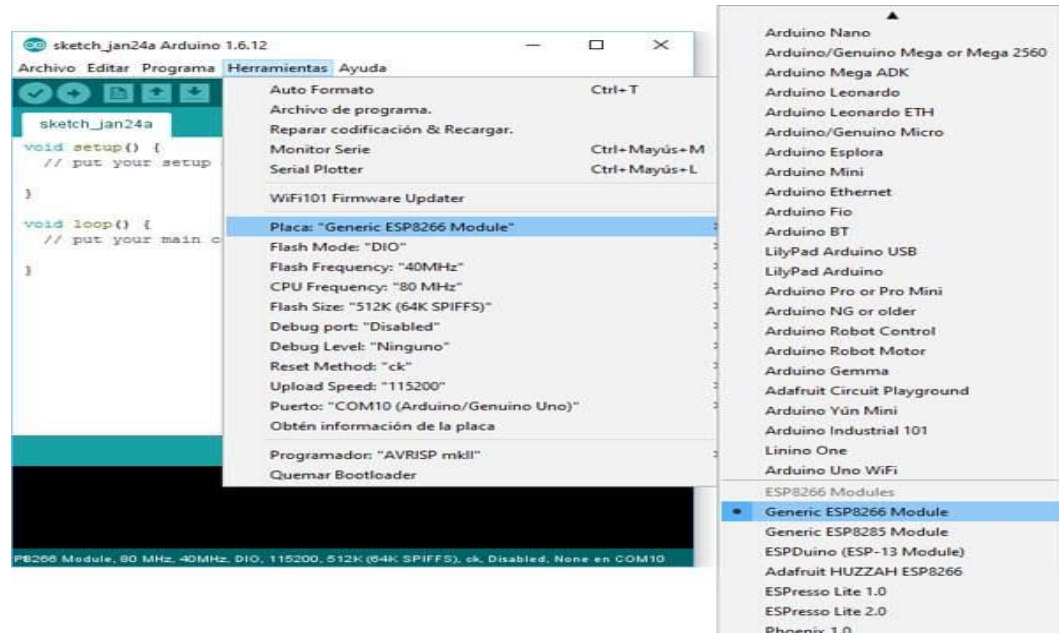


En el gestor de tarjetas buscar “esp” e instalar el *ESP8266 by ESP8266 Community*



Paso 3: seleccionar la placa Generic ESP8266 module

Una vez instalado la placa, ya se puede elegir entre los diferentes tipos de módulos ESP que se nos ofrecen: para utilizar el ESP-01 seleccionaremos Generic ESP8266 module.



ANEXO C: CONFIGURACIÓN MÓDULO MP3.

Instructions for WT9501M03

Features

- Support MP3 audio format, perfect compression mode, small capacity, good quality of the tone.
- Support 8 ~ 320Kbps MP3 audio files.
- Support SD card and U-disk(USB Flash Drive) with maximum capacity of 32G Byte.
- Support the playback of audio files in U-disk and SD card.
- Update the sound easily by changing the content in SD card or U-disk.
- Arrange the file by the sequence of copying.
- Control mode: key, serial control
- Support playbacing any voice sections.
- Dismension: 41mmX39mm
- Operating voltage: DC5V

3、Introduction of operating mode function

3.1、key mode

Under key mode, the normal state of I/O P01 ~ P06 is high, keeping 10ms negative pulse effective. Specific functions show as following:

| I/O port | P01 | P02 | P03 | P04 | P05 | P06 |
|----------|------|----------|------|------|------|------|
| function | play | previous | next | VOL+ | VOL- | stop |

3.2、serial mode

Based on UART serial commutation sequence, serial mode applies 9600 baud and customize the below communication protocol .It includes start code, data length, word size and end code.

Data format

| Start code | Data length | Operation code | Folder name ten thounds | Folder name thounds | Folder name hundreds | Folder name tens | Folder name ones | End code |
|------------|-------------|----------------|-------------------------|---------------------|----------------------|------------------|------------------|----------|
| 7E | 07 | XX | XX | XX | XX | XX | XX | 7E |

The numbers of operating code is ten.A0,B0 with data bit, other commands only need transmit operating code.

Folder tens, ones :

WT9501M03 can recognize MP3 files in memory card and distribute sequence numbers according to the coping sequence of files.Decimal number with five digits shows the sequence number.Decimal number should be converted into ASCII code.

The ten thousands of folder name is "0",the corresponding ASCII code is "30H"

The thousands of folder name is "1",the corresponding ASCII code is "31H"

The hundreds of folder name is "0",the corresponding ASCII code is "30H"

The tens of folder name is "4",the corresponding ASCII code is "34H"

The ones of folder name is "5",the corresponding ASCII code is "35H"

End code :7E

The end position must be 7E after **transmitting** each command.when the module checks that the data transmission is completed,the data will take into effect.

If playing the song with number 1045 in SD card, then **transmits** the following nine-byte data.The communication format as follows:

| Start code | Data length | Operating code | Folder name ten thousands | Folder name thousands | Folder name hundreds | Folder name tens | Folder name ones | End code |
|------------|-------------|----------------|---------------------------|-----------------------|----------------------|------------------|------------------|----------|
| 7E | 07 | A0 | 30 | 31 | 30 | 34 | 35 | 7E |

If stopping the broadcasting sound,then transmits the following data.

| Start code | Data length | Advertise Stop Operating code | End code |
|------------|-------------|-------------------------------|----------|
| 7E | 02 | A3 | 7E |

Volume control operation : 26-lever volume can be adjustable from 00H to 19H,00H for mute, 19H for the highest volume.

When the volume down to "0",then transmits the following operation data:

| Start code | Data length | Volume control code | End code |
|------------|-------------|---------------------|----------|
| 7E | 02 | A4 | 7E |

ANEXO D: TRAMA RECIBIDA POR EL COORDINADOR.

La trama recibida es:

7E 00 0A **83** 00 **C1** 2A 00 01 00 07 00 **07** 82

Donde:

7E: es el inicio de la trama.

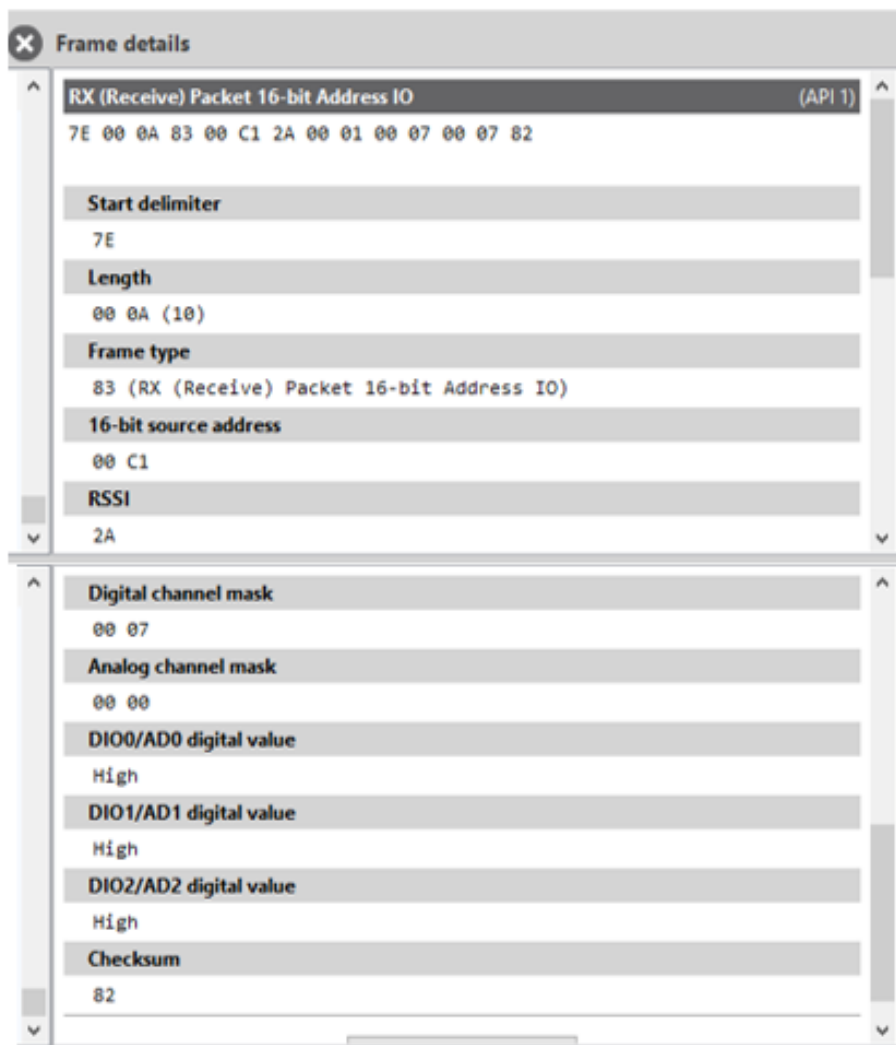
0A: longitud de la trama, (10 en decimal), la longitud se mide sin contar estos bytes, los de inicio y el checksum (83 00 C1 2A 00 01 00 07 00 07).

83: Recepción de datos conteniendo información de entradas digitales / analógicas

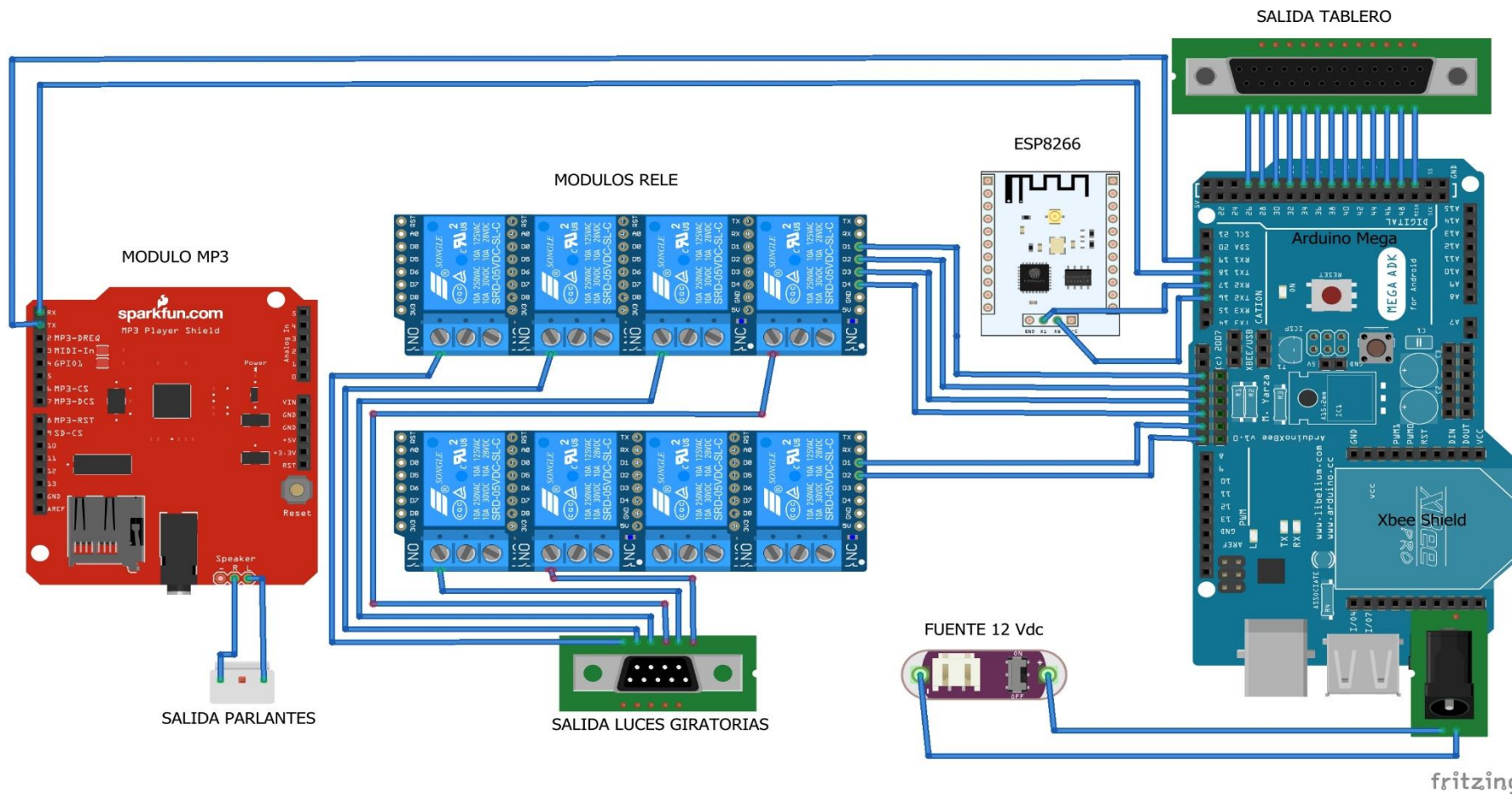
C1: Dispositivo final desde donde se transmite (célula 1) es el parámetro MY.

07: Son las máscaras de los puertos habilitados para nuestro caso los pulsadores. Es decir que, 0x00 0x07 = 0000 0000 0000 0111 (DI2, DI1, DI0) que indican que puertos están habilitados como entradas.

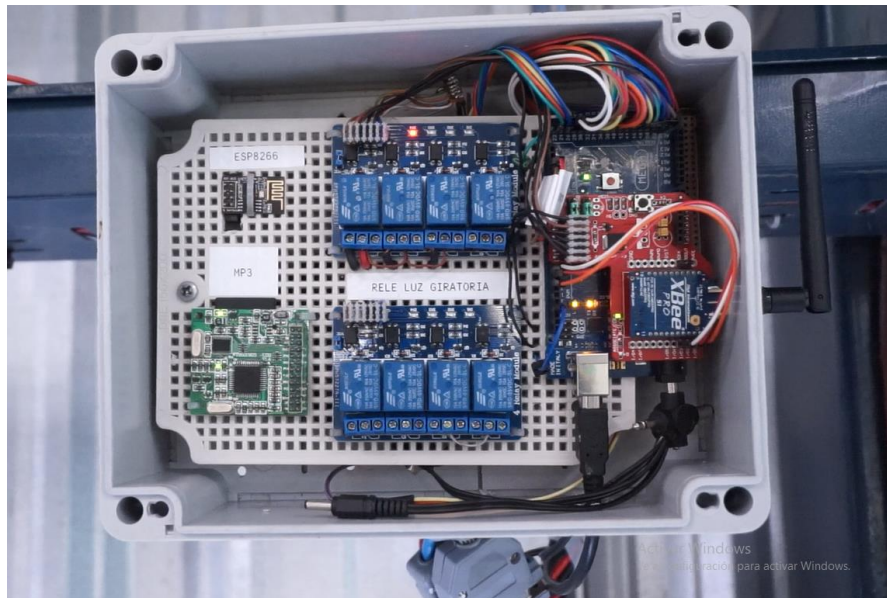
82: Es el checksum.



ANEXO E: SISTEMA ELECTRÓNICO ANDON.





Caja de control terminada.



Amplificador de audio del sistema.



ANEXO F: MÓDULOS USADOS EN LA PROPUESTA.

| | |
|---|--|
|  | ARDUINO MEGA |
| <p>Arduino es una plataforma física computacional open-hardware basada en una sencilla placa con entradas y salidas (I/O), analógicas y digitales y en un entorno de desarrollo que implementa el lenguaje Processing/Wiring.</p> |  |
| Datos Técnicos | |
| Microcontrolador | Atmega2560 |
| Alimentación | 3-12 Vdc |
| Memoria | 256 k tipo flash |
| Puertos I/O | 54 digitales / 16 analógicos |

| | |
|---|--|
|  | MÓDULO WI-FI ESP8266-01 |
| <p>Se trata de un chip integrado con conexión WiFi y compatible con el protocolo TCP/IP.</p> <p>El objetivo principal es dar acceso a cualquier microcontrolador a una red.</p> |  |
| Datos Técnicos | |
| Módulo WiFi | ESP8266-01 |
| Voltaje de Operación | 3 v - 3,6 v |
| Corriente de Operación | 80 mA |
| Temperatura de Operación | - 40 °C v 125 °C |

| | |
|---|--|
|  | MÓDULO XBee PRO S1 |
| <p>Esta es la versión Pro (de mayor potencia) del popular XBee.</p> <p>Este es un módulo inalámbrico de la Serie1 (protocolo 802.15.4) de 60 mW, ideal para comunicación punto a punto, multipunto y convertible en un punto de red en malla. Estos son mucho más potentes que los módulos simples XBee, ideales para cuando necesita más rango de alcance.</p> |  |
| Datos Técnicos | |
| Módulo XBee | PRO S1 |
| Voltaje de Operación | 3.3V |
| Velocidad máxima de datos | 250 kbps |
| Rango de Alcance | 1500m (1milla) |
| Tipo de Comandos | AT O API |

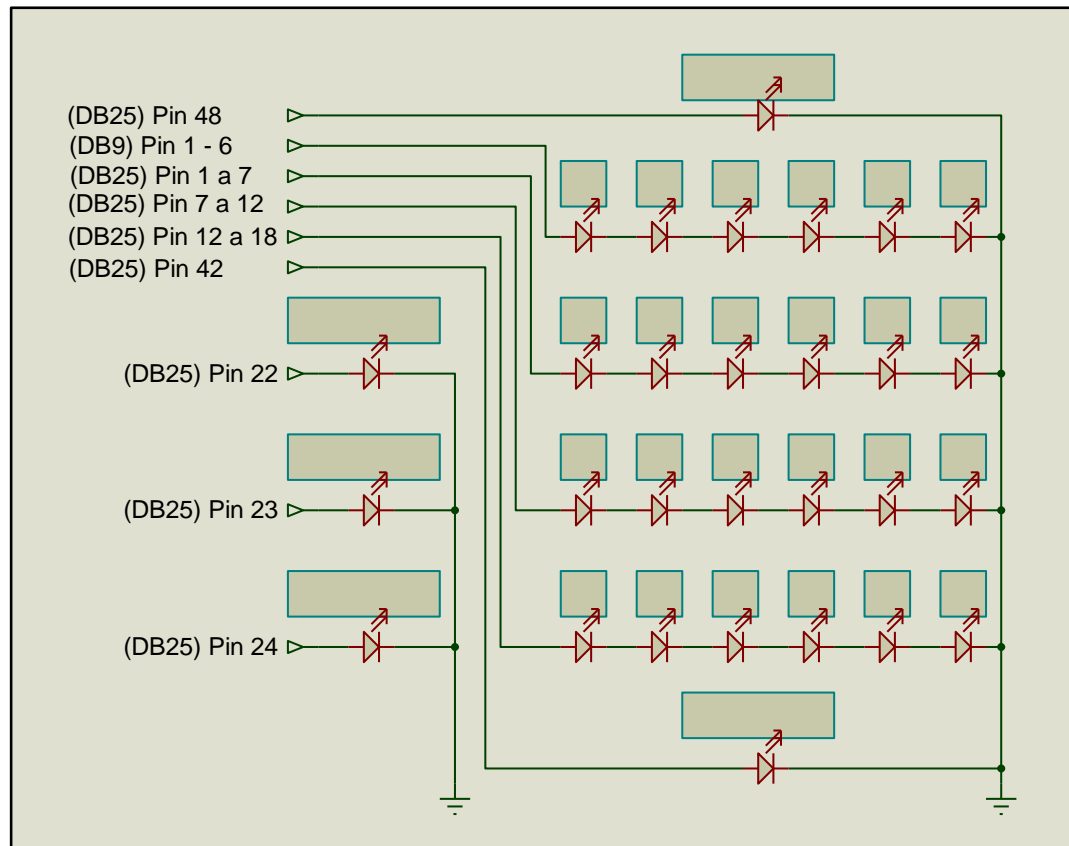
| | |
|---|--|
|  | MÓDULO XBee PRO RF RPSMA S1 |
| <p>Este modulo satisface las necesidades únicas de las redes de sensores inalámbricos de bajo consumo y bajo costo.</p> <p>El módulo es fácil de usar, requiere una potencia mínima y proporciona una entrega confiable de datos críticos entre dispositivos.</p> |  |
| Datos Técnicos | |
| Módulo XBee | PRO RF RPSMA S1 |
| Voltaje de Operación | 3.3V |
| Velocidad máxima de datos | 250 kbps |
| Rango de Alcance | 1500m (1milla) |
| Temperatura de Operación | - 40°C ~ 85°C |

| | |
|--|--|
|  | MÓDULO XBee SHIELD RF WIRELESS |
| <p>Este módulo (una placa de expansión sin módulo XBee) es una solución compatible diseñada para cumplir con redes de sensores inalámbricos de bajo costo y bajo consumo con necesidades especiales.</p> |  |
| Datos Técnicos | |
| Módulo XBee | SHIELD RF WIRELESS |
| Tamaño | 60x62x20 mm |

| | |
|--|--|
|  | MÓDULO XBee Explorer USB |
| <p>Esta es una unidad base de USB a serie fácil de usar para la línea XBee. Esta unidad funciona con todos los módulos XBee, incluidas las Series 1 y Series 2.5, estándar y Pro.</p> <p>Conecte la unidad al XBee Explorer, conecte un cable mini USB y tendrá acceso directo a los pines de serie y de programación en la unidad XBee.</p> |  |
| Datos Técnicos | |
| Módulo XBee | Explorer USB |

| | |
|--|--|
|  | PULSADOR |
| <p>Permiten la conmutación de elementos de un estado a otro.</p> <p>Se los puede encontrar según el caso como normalmente abiertos N.A. y normalmente cerrados N.C. como los de emergencia</p> |  |
| Datos Técnicos | |
| Voltaje de operación | 110 - 220 Vac |
| Tipos | N.A / N.C |
| Material | Plásticos, Metálicos, mixtos |
| Temperatura de funcionamiento | - 25 ~ 55 ° C |

ANEXO G: DIAGRAMA ELÉCTRICO DE LA CONEXIÓN DE LOS LEDS.



ANEXO H: FUNCIONAMIENTO DEL PROCESO DE APARADO.

| 4.- POLITICAS: | | |
|----------------|--|---|
| 4,1 | Recepción de cortes de

Troquelado | La recepción de cortes se realizara paulatinamente de acuerdo al flujo de producción. La última recepción del día se lo hará hasta las 09:00am del día siguiente, lo que se recepte desde esa hora en adelante entrará en la calificación al proveedor como atraso.
El responsable de subir los cortes es la sección de troquelado. No serán tomados en cuenta los cortes entregados en el turno de la mañana desde las 13:30 hasta las 14:00 y en la tarde desde las 21:30 a 22:00. |
| 4,2 | Cosido en máquinas automátatas | Los cortes que deban ser cosidos en máquinas automátatas deberán ser entregados por el abastecedor al personal operativo y se deben marcar las jabs con código asignado a cada operador. Los modelos serán rifados cada día para que no existan preferencias.
Los operadores deben notificar los tiques de cada jaba al final del turno para el pago de su premio. |
| 4,3 | Colocado de punteras | Los cortes que se deban colocar punteras serán entregados por el abastecedor al personal operativo y se deben marcar las jabs con código asignado a cada operador.
Los operadores deben notificar los tiques de cada jaba al final del turno para el pago de su premio. |
| 4,4 | Servicios | El trabajo de servicios debe ser entregado por el líder o abastecedor de turno (lazos, pegado tela termoadherible) según el std establecido.
Los operadores deben notificar los tiques de cada jaba al final del turno para el pago de su premio. |
| 4,5 | Distribución de planes a células | La distribución de cortes se debe hacer en el turno anterior, con todos los materiales e insumos necesarios. |
| 4,6 | Aparado de cortes | Es responsabilidad del líder de célula recibir el plan de cortes y coordinar el trabajo con los integrantes de la célula tomando en consideración lo siguiente:
1.- El plan que se entregue en cada jornada debe ser terminado en el mismo día.
1.1 En caso de que la célula no terminara el plan dentro de sus 8 horas de trabajo, esta deberá quedarse a terminar cuando estén en el turno de la mañana y venir más temprano el día siguiente cuando se encuentren en la tarde.
2.- La célula debe entregar los pares de reproceso del día anterior antes de empezar el nuevo plan.
3.- La células de aparado deben entregar la mitad de la producción asignada hasta la media jornada. |
| 4,7 | Moldeado de talón | Los operadores de las máquinas moldeadoras deben completar su std asignado a su jornada de trabajo.
Los operadores deben notificar los tiques de cada jaba al final del turno para el pago de su premio. |
| 4,8 | Revisión de calidad | Verificadores.- Es obligación de los verificadores de turno revisar al 100% los trabajos que se ejecutasen en su turno, identificando desviaciones de proceso o de especificaciones.
Revisores.- Los revisores de calidad deben revisar al 100% los cortes terminados cumpliendo el procedimiento de revisión, colocando el sello de identificación (trazabilidad). |
| 4,9 | Entrega de cortes a bodega | Responsable: Supervisor y líder de turno
Horario de entrega.- la producción será entregada hasta el 95% en el siguiente turno y el 5% restante (reproceosos) hasta 2 horas después del siguiente día. |
| 4,10 | Incumplimientos a responsabilidades | El incumplimiento a las actividades asignadas serán motivo para la aplicación de sanciones según establece el reglamento interno de trabajo |