

Instituto Tecnológico de Costa Rica

Carrera de Ingeniería Mecatrónica



Diseño de un sistema de monitoreo remoto de las máquinas de producción con conexión a una plataforma para la gestión de información en *Gutis Ltda.*

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura

Theodore José Kop Montero

Cartago, Noviembre 2018.

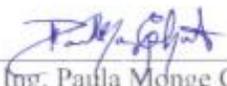
INSTITUTO TECNOLÓGICO DE COSTA RICA
CARRERA DE INGENIERÍA MECATRÓNICA
PROYECTO DE GRADUACIÓN
ACTA DE APROBACIÓN

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal



Ing. Ronald Bolaños Maroto
Profesor lector



Ing. Paula Monge Chanto
Profesora lectora



Ing. Carlos Piedra Santamaría
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Carrera de Ingeniería Mecatrónica

Cartago, 16 de noviembre de 2018

Declaratoria de Autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, Noviembre 2018



Theodore José Kop Montero

Céd: 1-1636-0872

Resumen

En el presente documento se describe el desarrollo de una plataforma de software SCADA con conexión a la red de Internet para el monitoreo de manera remota de equipos de producción en una empresa de la industria farmacéutica. Esto con la intención de generar un proyecto piloto altamente escalable para facilitar futuras implementaciones dentro de la empresa. Además, se potencia la transición de la empresa a la digitalización de sus datos de máquinas y el análisis inteligente de estos; conforme a las tendencias del mercado actual en el contexto de la Cuarta Revolución Industrial en busca de garantizar su competitividad.

En la compañía se cuenta con equipos altamente automatizados (con muy poca intervención humana), los cuales a su vez son fuentes de una enorme cantidad de información potencialmente valiosa que actualmente no se aprovecha ni se toma en consideración para la toma de decisiones. Además, los únicos indicadores de productividad utilizados para las máquinas en cuestión, se calculan de forma manual y toman en consideración solamente factores de disponibilidad de estas.

Se identifican variables de interés y sus fuentes con la intención de proveer al sistema con los datos necesarios para corregir las limitantes diagnosticadas y garantizar una solución robusta pero fácilmente replicable en otras máquinas. Se diseña una base de datos en *MySQL* que ofrece respaldo a los datos y permite la generación de históricos. De igual modo se diseña una plataforma en *Ignition* para la visualización remota en tiempo real del estado de las máquinas, la interacción con operarios y usuarios administrativos (de más alto nivel) y el cálculo del OEE para los equipos incluidos en el sistema. Se programan reportes automáticos que se distribuyen a sus destinatarios a través de email.

Palabras Clave: *SCADA, Ignition, MySQL, Base de Datos, OEE, Monitoreo remoto, IoT.*

Abstract

The project presented in this document describes the development of a SCADA system with internet connection for the remote monitoring of machines in a pharmaceutical company. The aim of the project is to create a prototype that can be highly scalable in order to facilitate future implementations and upcoming upgrades in the system. The digitalization of the information and the intelligence of data management are promoted accordingly to the impending market tendencies of the Forth Industrial Revolution, in an attempt to guaranty the competitiveness of the company.

The vast majority of machines in Gutis are almost fully automated (operate completely with minimum human intervention), and are sources of an enormous amount of data that are potentially valuable. These data are not being used or taken in consideration for the decision making processes in the company. Furthermore, the productivity indicator for each machine only takes into account the availability of the machines and is manually calculated.

The proposed solution identifies different data sources and variables of interest that provide the system with the information needed to solve the current limitations and guaranty a robust platform that can be easily replicated for other machines. A *MySQL* database is design to offer backup or support, and that allows the creation and management of historical data. Likewise, a program is design using *Ignition* software, which allows the remote and real-time monitoring of the state of the machines, the interaction with operators and administration users (higher level users), and calculates the Overall Equipment Effectiveness (OEE) for every machine included in the system. Finally the project offers automatic report generation that are distributed to those in charge via email.

Keywords: *SCADA, Ignition, MySQL, Database, OEE, Remote monitoring, IoT.*

Dedicatoria

A Dios.

Por nunca abandonarme en los estudios y principalmente por la fortaleza, sabiduría y consuelo en todos los momentos complicados. Por la tranquilidad y respaldo en los momentos de estrés que me permitieron perseverar y dar lo mejor de mí, confiando en TI. Por haberme facilitado una vida llena de bendiciones, en la que pueda enfocarme en mis estudios y cumplir mis sueños.

A mis Padres: Theodore Kop Vargas y Ana Ligia Montero González.

Por haberme enseñado a través de su ejemplo lo que significa el sacrificio y el esfuerzo que se requiere en la vida para conseguir las metas que uno se fije. Por enseñarme a soñar en grande pero a luchar el doble por conseguir los sueños y nunca dejar derrotarse o desanimarse por los tropiezos del camino. Por ser las dos personas que más admiro y respeto en el mundo. Son un modelo a seguir del cual siempre aprenderé algo nuevo y podré usar para motivarme en cualquier situación. Por su apoyo y amor incondicional durante mis 22 años de vida

A mi hermana: Mariana Kop Montero.

Por ser mi mejor amiga y consejera personal. Por enseñarme que el valor de la vida está en las relaciones y personas que te rodean y que con esfuerzo y el apoyo de tus seres queridos lo demás vendrá por añadidura. Por enseñarme el valor de una verdadera amistad y brindarme tu amor y apoyo en todo momento.

A mis familiares.

En especial a mi tía Cristina por ser mi segunda mamá y el ejemplo más claro del significado de fortaleza en mi vida. Por estar pendiente en todo momento de mis estudios y de mi estado. Por el esfuerzo, interés y dedicación demostrados durante todos estos años tanto para mí como para mi hermana. A mis tíos por ser consejeros y amigos de vida que me han enseñado el valor de la familia.

A mis amigos.

Por el apoyo mutuo durante la carrera, consejos y momentos compartidos. Sin dudarlo son mucho más que futuros colegas y el regalo más importante que me llevó de la carrera. Espero en el futuro poder continuar las amistades establecidas en estos 5 años y compartir ideas, consejos y hasta algún negocio. En especial a Alex, José y Fernando, aunque dedico este trabajo a todos mis compañeros con los que pude compartir algún momento y que no lo mencione directamente.

A mis profesores.

A todos los profesores que de algún modo hayan contribuido con mi formación profesional y hayan dedicado su tiempo y esfuerzo en el proceso. A aquellos profesores que se destacaron por

impartir más que una lección técnica y procuraron dejar alguna lección de vida, les agradezco de manera especial y les dedico este trabajo.

Agradecimientos

A Gutis Ltda.

Empezando principalmente por Don Norman Gutiérrez por haberme recibido en su empresa y abrirme las puertas para realizar mi proyecto de graduación. Agradecerle por el seguimiento y recurso humano dedicado para la elaboración de este proyecto. Sinceramente espero que este trabajo y la plataforma desarrollada sean de mucha utilidad e impulsen la competitividad de Gutis.

A todo el equipo del departamento de mantenimiento, en especial al Ing. Greivin Granados por su apoyo, consejo y sugerencias brindadas durante el proyecto y por hacerme parte de su equipo.

Al Tribunal Evaluador.

A mi profesor tutor Carlos Piedra Santamaría por su experiencia aportada al proyecto y su guía durante la elaboración del mismo. A los profesores lectores Paula Monge Chanto y Ronald Bolaños Maroto por sus correcciones, comentarios y sugerencias. Les agradezco por permitir que este proyecto sea posible y por su evaluación.

A mi cuñado Jovan Varga.

Le agradezco toda la ayuda, consejo y explicaciones brindadas para el manejo de datos. Su aporte al proyecto me permitió realizar un mejor diseño de la base de datos y fue vital para su desarrollo.

Contenido

Portada	1
Resumen	4
Abstract	5
Dedicatoria	6
Agradecimientos	8
Contenido	9
Índice de Figuras	11
Índice de Tablas	12
Índice de Ecuaciones	13
Capítulo 1 Introducción	14
1.1 Entorno del proyecto.....	14
1.2 Definición del problema	16
1.2.1 Diagrama Causa – Efecto (Ishikawa)	16
1.2.2 Generalidades y Justificación	16
1.2.3 Síntesis del problema	18
1.3 Enfoque de la solución.....	19
1.4 Objetivos.....	22
1.4.1 Objetivo General.....	22
1.4.2 Objetivos Específicos	22
1.5 Alcance	23
Capítulo 2 Descripción del proceso productivo en Gutis Ltda.	24
2.1 Procesos de Producción en Gutis Ltda.....	24
2.1.1 Proceso de producción de inyectables	24
2.1.2 Proceso de producción y envase de medicamentos en presentación líquida	25
2.1.3 Proceso de producción de supositorios	26
2.1.4 Proceso de producción de ungüentos	27
2.1.5 Proceso de producción de tabletas	28
2.1.6 Proceso de producción de medicamentos hormonales.....	29
2.2 Características de las máquinas blisteadoras marca Uhlmann	30
2.3 Funcionamiento de la máquina de blísteres B 1330/1440137 Uhlmann.	32
2.3.1 Elementos de señal presentes en el equipamiento de Uhlmann.	35
Capítulo 3 Estado del Arte	37
Capítulo 4 : Marco teórico	40
4.1 Sistemas de comunicación electrónicos.....	40
4.2 Redes	42
4.2.1 Topologías de Red	45

4.3 Software de red	49
4.4 Telemetría	50
4.5 SCADA.....	50
4.5.1 Ventajas de un sistema SCADA	51
4.5.2 Estructura de un sistema SCADA.....	51
4.5.3 Protocolos de comunicación para un SCADA	52
4.5.4 Tags	56
4.6 Bases de Datos.....	57
4.6.1 Datos.....	57
4.6.2 Información	57
4.6.3 Entidades	57
4.6.4 Atributos	57
4.6.5 Relaciones de datos.....	57
4.6.6 Llaves	59
4.6.7 Definición de una Base de Datos.	59
4.7 Semáforo Industrial	59
Capítulo 5 Estrategia de Solución	60
5.1 Identificación de las variables, señales o estados de interés.	60
5.2 Selección de indicadores de desempeño	67
5.3 Diseño de la Base de datos que alimenta al sistema	70
5.4 Selección de la plataforma tecnológica.....	74
5.4.1 Wonderware InTouch	75
5.4.2 Ignition.....	75
5.5 Arquitectura del sistema	76
5.6 Programación en Ignition.	79
5.6.1 Conceptos de Ignition	80
5.6.2 Variables Utilizadas.....	88
5.6.3 Ventanas Creadas.....	96
5.6.4 Lógica de programación	99
Capítulo 6 Resultados.....	115
6.1 Principales ventanas.....	116
6.1.1 Navegación	116
6.1.2 Monitoreo Uhlmann1 y Monitoreo Uhlmann 2	118
6.1.3 Ventana de Manejo de Usuarios y Ventana de Manejo de Horarios	122
6.1.4 Ventana para Añadir Máquinas y Ventana para Añadir Turnos de Trabajo a las máquinas.....	124
6.1.5 Ventana de Estadísticas	128
6.2 Reporte	130
Capítulo 7 Análisis Financiero	134
7.1 Inversión Inicial Proyecto Piloto	135

7.2 Inversión Inicial Proyecto Implementado	136
7.3 Rentabilidad del Proyecto	138
Capítulo 8 Conclusiones	139
Capítulo 9 Recomendaciones	141
Capítulo 10 Referencias	143
Capítulo 11 Apéndices	146
Manuales de Usuarios (Elaboración Propia, 2018).....	146
Manual Usuario para la instalación de Ignition 7.9.5 en Windows	146
Manual Usuario para la instalación de Base de Datos MySQL	152
Manual Usuario para la conexión de Base de Datos MySQL con plataforma de Ignition a través del Gateway.	167

Índice de Figuras

Figura 1.1 Diagrama Causa – Efecto.....	16
Figura 1.2 Uhlmann Blister line BEC 300.....	20
Figura 1.3 Diagrama de bloques general del proyecto.	21
Figura 2.1 Diagrama del proceso simplificado de producción de inyectables en Gutis Ltda.	25
Figura 2.2 Diagrama del proceso simplificado de producción de medicamentos líquidos en Gutis Ltda.	26
Figura 2.3 Diagrama del proceso simplificado de producción de supositorios en Gutis Ltda.	27
Figura 2.4 Diagrama del proceso simplificado de producción de ungüentos en Gutis Ltda.	28
Figura 2.5 Diagrama del proceso simplificado de producción de tabletas en Gutis Ltda.	29
Figura 2.6 Estructura de la placa de datos de maquinaria Uhlmann.	30
Figura 2.7 Plano de conjunto simplificado de la máquina de blísteres B 1330/1440137 de la marca Uhlmann.	32
Figura 2.8 Columna Luminosa Equipos Uhlmann.	36
Figura 3.1 Rédito previsto a nivel mundial para tecnologías de IoT para el año 2020.....	38
Figura 3.2 Mercado Global de SCADAs según usuario final para el año 2014.	39
Figura 4.1 Diagrama de bloques simplificado de un sistema de comunicación electrónica.	40
Figura 4.2 Efecto del ruido sobre una señal.....	41
Figura 4.3 Clasificación de las redes según escala.	44
Figura 4.4 Topología en Bus de una red de dispositivos.	45
Figura 4.5 Topología en Anillo de una red de dispositivos.	46
Figura 4.6 Topología en Estrella de una red de dispositivos.	47
Figura 4.7 Topología en malla de una red de dispositivos.	48
Figura 4.8 Topología en árbol de una red de dispositivos.	48
Figura 4.9 Estructura de red entre dos dispositivos, capas, protocolos e interfaces.	50
Figura 4.10 Estructura de un sistema SCADA.	52
Figura 4.11 Estructura base que utiliza el protocolo MQTT.	54
Figura 4.12 Diagrama de relación de Entidades (ER) Uno-a-Uno.	58
Figura 4.13 Diagrama de relación de Entidades (ER) Uno-a-Muchos.	58
Figura 4.14 Diagrama de relación de Entidades (ER) Muchos-a-Muchos.	59
Figura 5.1 Desglose de tiempo programado en Uhlmann 1 para la semana 31.	62
Figura 5.2 Desglose de tiempo programado en Uhlmann 2 para la semana 31.	62
Figura 5.3 Concepto general de monitoreo para las máquinas Uhlmann 1 y Uhlmann 2.....	66
Figura 5.4 Factores del OEE.....	69
Figura 5.5 Diagrama de Relación de Entidades (ERG) para el diseño de base de datos.	74
Figura 5.6 Arquitectura Estándar del sistema diseñado utilizando Ignition.	78
Figura 5.7 Importación del archivo que contiene el proyecto desarrollado a otro servidor de Ignition.	80
Figura 5.8 Scripts en Ignition.	82

Figura 5.9 Ejemplo de código para Eventos en el Tag en Ignition.	84
Figura 5.10 Ejemplo de código para expresiones en Tags de Ignition.	85
Figura 5.11 Ejemplo de ligar el valor de una propiedad de un componente en Ignition a través de una expresión. ..	86
Figura 5.12 Opciones disponibles para los componentes en Ignition.	87
Figura 5.13 Ejemplo de programación para componente cuando se realiza una acción sobre este.	88
Figura 5.14 Regreso a cero para los paros por espera y paros por material cuando sucede otro tipo de paro.	100
Figura 5.15 Diagrama de Flujo para la lógica de programación utilizada para el monitoreo de la Uhlmann 1.	102
Figura 5.16 Código ejecutado cuando se realiza una justificación de paro.	103
Figura 5.17 Cálculo de la diferencia en segundos ente el tiempo actual y el tiempo en que sucede el paro.	105
Figura 5.18 Diagrama de flujo de la lógica utilizada para la medición de los tiempos totales activa y detenida de la máquina.	107
Figura 5.19 Código ejecutado 1 segundo después del momento programado para el reset.	110
Figura 5.20 Diagrama de lógica de control implementada para el reset diario de las variables y la actualización de la base de datos.	111
Figura 5.21 Consultas a base de datos para la obtención de los valores estadísticos de una máquina dada para una fecha dada.	114
Figura 6.1 Acceso inalámbrico al Gateway de Ignition.	116
Figura 6.2 Ventana de Navegación.	117
Figura 6.3 Lista de opciones en la ventana de Navegación.	117
Figura 6.4 Ventanas de Monitoreo en la plataforma.	117
Figura 6.5 Ventanas de Administración en la plataforma.	118
Figura 6.6 Ventana de Estadísticas Básicas en la plataforma.	118
Figura 6.7 Vista General Ventana Monitoreo Uhlmann 1.	119
Figura 6.8 Ventana de justificación de Paro para la Uhlmann 1 al momento de detectar que la máquina se detuvo.	120
Figura 6.9 Cronómetros para registro de tiempos, gráfico de Estado de la Uhlmann1 en tiempo real y botón de más información.	120
Figura 6.10 Ventana de Gráficos Motivos de Paro para la Uhlmann 1.	122
Figura 6.11 Ventana Principal para Manejo de Usuarios en la plataforma.	123
Figura 6.12 Ventana Principal para Manejo de Usuarios en la plataforma.	124
Figura 6.13 Vista General Ventana Añadir Máquinas.	125
Figura 6.14 Error al tratar de introducir una máquina a la base de datos con el mismo nombre que otra dentro de la tabla.	125
Figura 6.15 Inserción correcta de una nueva máquina a la base de datos.	126
Figura 6.16 Negación de acceso a manipular la base de datos a un usuario con rol restringido.	126
Figura 6.17 Vista general de la ventana para añadir turnos de trabajo a las máquinas vinculadas a la plataforma.	127
Figura 6.18 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 1 el día 26 de Octubre del 2018.	128
Figura 6.19 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 1 el día 29 de Octubre del 2018.	129
Figura 6.20 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 2 el día 29 de Octubre del 2018.	129
Figura 6.21 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 2 el día 30 de Octubre del 2018.	130
Figura 6.22 Flujo de correos recibidos a una cuenta de correo personal con reportes diarios.	131

Índice de Tablas

Tabla 2.1 Características de las líneas blisteadoras y empacadoras de <i>Uhlmann (Blister line BEC 300)</i>	31
Tabla 5.1 Determinación de las señales de Interés para el diseño de la plataforma.	64
Tabla 5.2 Tipos de Tags en Ignition.	83
Tabla 5.3 Descripción de las variables de la Uhlmann1.	89
Tabla 5.4 Descripción de las variables del cronómetro de la Uhlmann1.	91
Tabla 5.5 Descripción de las variables de la Uhlmann2.	92

Tabla 5.6 Descripción de las variables del cronómetro de la Uhlmann1.	95
Tabla 5.7 Descripción de las variables utilizadas en el software para el desarrollo de indicadores de desempeño.	96
Tabla 5.8 Descripción de las ventanas del sistema diseñado.	98
Tabla 7.1 Costos totales Inversión Inicial Proyecto Piloto	135
Tabla 7.2 Costos Totales Inversión Inicial Implementación en Gutis.	137

Índice de Ecuaciones

Ecuación 4-1	41
Ecuación 5-1	69
Ecuación 5-2	69
Ecuación 5-3	70
Ecuación 5-4	70

Capítulo 1 Introducción

1.1 Entorno del proyecto

Gutis es una compañía de clase mundial, de origen 100% nacional, con más de 50 años de experiencia en la industria farmacéutica. La compañía se esfuerza continuamente en el desarrollo de nueva tecnología y estrategias en busca del cuidado de la salud de las personas y el mejoramiento en su calidad de vida. De este modo, la compañía concentra sus esfuerzos en la investigación y el desarrollo de nuevos productos farmacológicos que le permitan impactar, por medio de la innovación y tecnología, la vida de sus clientes. Gutis busca brindar un cuidado responsable y certero en la salud de las personas, ofreciendo alivio y permitiéndoles hacer más y por más tiempo (Gutis, 2017).

Gutis cuenta con departamentos de Investigación y Desarrollo en Costa Rica que se encargan de conceptualizar nuevos productos y desarrollar nuevas fórmulas farmacéuticas. Además la compañía ya sea produce o distribuye medicamentos en ocho países latinoamericanos (Ecuador, Guatemala, Honduras, Nicaragua, Panamá, República Dominicana, El Salvador y Costa Rica) y tiene acceso al mercado farmacéutico europeo distribuyendo una serie de sus fármacos en España. Dentro de la familia de productos farmacéuticos elaborados por Gutis, se encuentran los: medicamentos del aparato respiratorio, anticonceptivos, medicamentos para el control de peso, dermatológicos, analgésicos, medicamentos del aparato gastrointestinal, psiquiátricos y una serie de suplementos (Gutis, 2017). Dichos productos a su vez se encuentran clasificados en distintas áreas de producción a lo interno de Gutis, según su composición química en: sólidos (tabletas), jarabes, emulsiones y supositorios (Granados G, 20187).

Hoy en día la compañía cuenta con alrededor de 400 empleados a nivel nacional (dentro de la zona industrial de Pavas), estos se encuentran distribuidos a lo interno en distintos departamentos. Gutis se encuentra estructurada en diferentes secciones que desempeñan actividades de producción, mantenimiento, finanzas, mercadeo, investigación y desarrollo entre otras. Este proyecto se realizará en conjunto con el departamento de producción y mantenimiento, teniendo presente que sus resultados irán enfocados al departamento de finanzas.

La compañía se encuentra “comprometida con la protección del medio ambiente a través de un enfoque de desarrollo sostenible” (Gutis, 2017). Por ello se busca la reducción de los riesgos

ambientales y la optimización del consumo del recurso energético, por medio del planteamiento de objetivos ambientales en **todas** las operaciones que se realicen en la empresa. De igual modo, Gutis presta atención especial en sus acciones para garantizar la calidad de sus productos. Por lo que todos sus procesos se realizan bajo estándares de calidad mundial de la industria farmacéutica a nivel internacional; y por lo tanto se encuentran regulados o sujetos a las normativas de la FDA, ANVISA, *The World Health Organization*, *The European Medicines Agency* y La Comisión Federal para la Protección contra Riesgos Sanitarios (Cofepris) (Gutis, 2017).

1.2 Definición del problema

1.2.1 Diagrama Causa – Efecto (Ishikawa)

A continuación en la Figura 1.1 se presenta un diagrama de causa-efecto que da paso a la definición del problema por resolver en este proyecto.

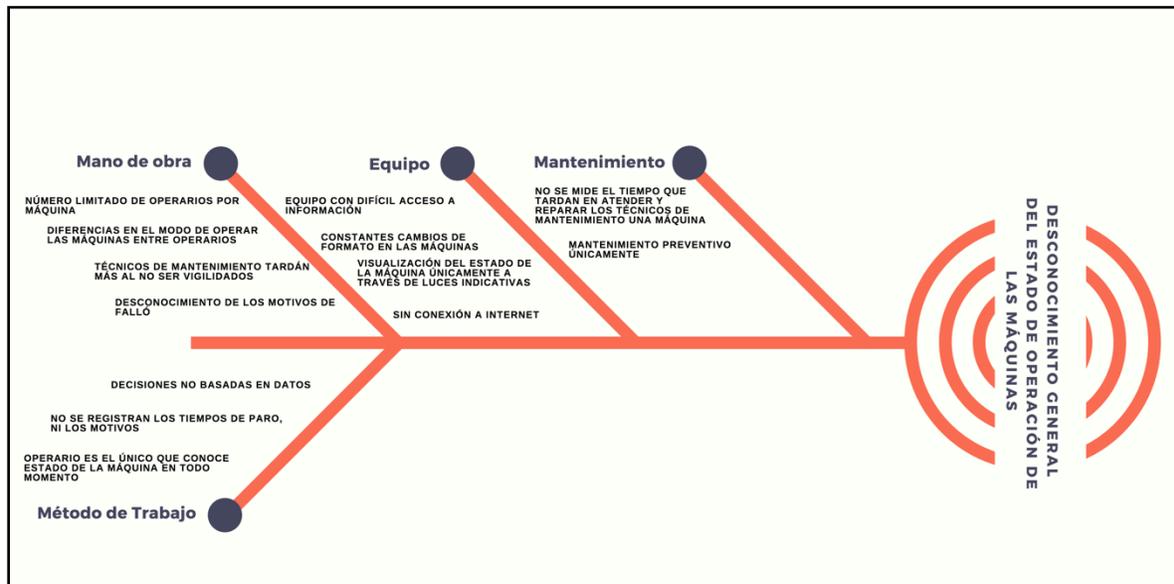


Figura 1.1 Diagrama Causa – Efecto.

[LUCIDCHART (Elaboración Propia, 2018)]

1.2.2 Generalidades y Justificación

Actualmente *Gutis* en todas sus líneas de producción de fármacos cuenta con maquinaria compleja, que en su mayoría trabaja de manera automatizada con una gran cantidad de variables (medibles) que pueden alterar el volumen de producción y por ende la eficiencia productiva de la empresa. A pesar de que la empresa cuenta con la tecnología para informar sobre algún tipo de fallo en alguna de las estaciones de producción, esta información se trabaja únicamente de manera individual y es conocida y atendida solamente por el operario o técnico encargado de dicha estación.

Dicho de otro modo, esta información al igual que otras variables que señalan el estado de la maquinaria y repercuten directamente en la eficiencia de los distintos procesos, se desconocen por los encargados de niveles superiores en donde se toman las decisiones. Además toda esta

información potencialmente valiosa, pasa inadvertida y desperdiciada en su totalidad una vez el operario haya solucionado los problemas que se le presenten.

Es precisamente esta estructura de fábrica en donde las líneas de producción se encuentran altamente automatizadas y en donde la información concerniente al estado de cada estación se maneja individualmente, la que evidencian la necesidad de la empresa a migrar a sistemas conectados que faciliten la transición de Gutis a una fábrica inteligente capaz de competir dentro de la cuarta revolución industrial. El éxito en la integración de la información de todas las máquinas de la fábrica, junto con información concerniente a decisiones administrativas y del departamento de mantenimiento y producción, crea nuevas oportunidades de crecimiento para la empresa y garantiza su competitividad. Tal como lo señala la edición correspondiente al 2014 del *Harvard Business Review*, las compañías que tarden demasiado en migrar a este tipo de estructuras y plataformas corren el riesgo de afectar negativamente su competitividad; y por ello recomienda “comenzar a recolectar y analizar información lo más inmediato posible” (Porter, M, 2014).

1.2.3 Síntesis del problema

Existe desconocimiento sobre información potencialmente pertinente, propia del equipamiento de las líneas de producción de blistado y empaque de tabletas, que repercute para la toma de decisiones fundamentadas o sostenidas en el análisis integral de dicha información que impulsen la eficiencia productiva en Gutis.

1.3 Enfoque de la solución

Para motivos de aplicación de este proyecto dentro de la empresa, se plantea el mismo inicialmente como un proyecto piloto que sea capaz de demostrar la importancia y el potencial de la digitalización de la información de las máquinas para el posterior análisis de esta.

De este modo se pretende iniciar e impulsar una transición hacia la industria 4.0 y la digitalización en Gutis, logrando la conectividad de las máquinas de producción con un centro de datos en donde se pueda monitorear de manera remota su funcionamiento y a la vez permitir el análisis de dicha información. Así como la generación de reportes y alarmas (vía email, mensaje de texto o algún otro medio) que se enlacen con algún software de gestión de información más amplio (que contemple las decisiones administrativas). Con esto se pretende evaluar información hasta el momento desconocida en la empresa, tales como tiempos de paro en los equipos, tipos de fallas presentadas (generación de un histórico de los fallos que se han presentado), duración de las acciones de mantenimiento, y otro tipo de información que pueda resultar valiosa en la construcción de indicadores de desempeño y en la toma de decisiones para aumentar la eficiencia productiva.

Sin embargo al tratarse únicamente de un proyecto piloto, el mismo se realizará únicamente sobre dos líneas de producción dentro de la empresa. Ambas líneas de producción seleccionadas, corresponden al área de sólidos (tabletas) y son responsables de producir individualmente más de 300 blísteres por minuto, además de ser capaces de manejar una gran variedad de medicamentos a través de cambios de formato. El proceso de empaque se realiza en su mayoría de manera automatizada gracias a dos sistemas de producción en línea con máquina blistera y cartonera, se utilizan operarios únicamente encargados de alimentar el sistema con el producto y materia prima, realizar cambios de formato y monitorear el correcto funcionamiento de las máquinas. Este proyecto se enfocará en la etapa de blisteo de ambas líneas de producción.

Las máquinas utilizadas son las *Uhlmann Blister line BEC 300* que se muestran en la Figura 1.2 y es precisamente sobre estas máquinas donde se enfocará la implementación del proyecto y por ende la solución del mismo.



Figura 1.2 Uhlmann Blister line BEC 300.

[Uhlmann, 2010]

Con el proyecto se pretende brindar una solución tecnológica que facilite la integración de la información del estado en tiempo real y condiciones de operación de las máquinas con información de carácter administrativo y de producción.

Permitiendo a Gutis la generación de indicadores de desempeño basados en información actualizada y un monitoreo remoto de los equipos en que aplique este proyecto o similares. Se propondrá la adquisición de las señales de interés, algún sistema electrónico encargado del acondicionamiento (en caso de requerirse), lectura y transmisión de las señales, un receptor y algún *software* de gestión de información (tales como Ignition de Inductive Automation, ManageEngine Applications Manager, SAP HANA, MySQL, WERMA-WIN, Wonderware), que se encargue de almacenar en una base de datos la información recibida. Este último debe ser capaz de ser accedido de manera remota y permitir el análisis de los datos. En la Figura 1.3 se muestran los bloques principales que se deben desarrollar durante la ejecución del proyecto.

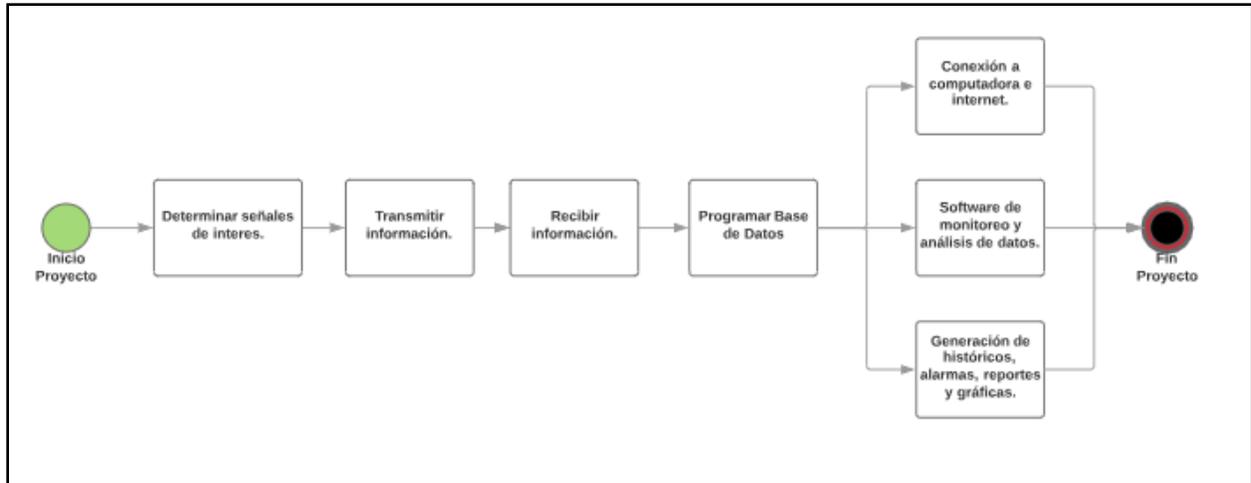


Figura 1.3 Diagrama de bloques general del proyecto.

[LUCIDCHART (Elaboración Propia, 2018)]

Para la selección de una propuesta de solución al problema se deberán considerar los siguientes aspectos:

- Precio del equipo requerido.
- Capacidad de utilización de infraestructura y equipo ya existente dentro de la empresa.
- Facilidad en el uso del software por emplear y comprensión de los datos.
- Capacidad de almacenamiento de la información.
- Accesibilidad y mantenimiento a la base de datos.
- Reparación sencilla en caso de fallos.
- Rápida instalación.

1.4 Objetivos

1.4.1 Objetivo General

Diseñar un sistema de monitoreo remoto que permita la integración de la información del estado y condiciones de operación en dos líneas de empaque de tabletas farmacéuticas para el establecimiento de indicadores de desempeño y criterios de toma de decisiones.

1.4.2 Objetivos Específicos

- Determinar las señales eléctricas, variables de la máquina u otras fuentes de información que son indicativas del estado de las máquinas.

Indicador: Tabla con las principales variables de interés aprobada por el Ingeniero Supervisor en la empresa.

- Seleccionar un sistema de comunicación capaz de enviar las señales adquiridas a un puesto de monitoreo remoto dentro de la empresa.

Indicador: Datos recibidos 100% correctos con respecto a los enviados, en una distancia no mayor a 300 metros utilizando un protocolo de comunicación estándar.

- Programar un *software* que permita el almacenamiento y la integración de la información y con acceso a la red de internet.

Indicador: Visualización de los datos desde una plataforma móvil.

- Establecer una serie de indicadores de desempeño y productividad de las máquinas en base al análisis de los datos obtenidos.

Indicador: Gráficas, reportes de productividad, históricos de fallos y alertas.

1.5 Alcance

Esta sección del proyecto es de vital importancia y debe quedar bastante claro al lector, ya que, junto con los objetivos planteados anteriormente, aquí se definen los límites que envuelven al proyecto. Aún más por la naturaleza propia del presente proyecto, al tratarse de un proyecto piloto, se debe aclarar el impacto que se pretende conseguir y las limitaciones con respecto a una implementación completa dentro de la planta productiva en Gutis.

Al igual que un prototipo cuenta con varias versiones antes de convertirse en una herramienta validada en la industria, este proyecto se enfoca en la consecución de un primer prototipo, el cual deberá ser lo suficientemente escalable y flexible para que eventualmente y siguiendo su debido proceso se consiga una herramienta completa. Esto no significa en un proyecto de menor calidad, por el contrario, al tratarse de la primera etapa en el diseño e implementación de un proyecto nunca antes abordado en Gutis, se podría decir que la eventual concreción de una herramienta completa para la industria farmacéutica depende del acertado diseño que se plantee en este proyecto. Aquí la importancia entonces de reconocer las limitaciones, y visualizar su potencial futuro y viabilidad en la empresa.

Concretamente el alcance del presente proyecto consiste en el diseño de un prototipo de un sistema de monitoreo remoto desde una plataforma que permita el análisis de la información para la gestión y toma de decisiones en la empresa. Y como parte de su diseño se debe considerar la escalabilidad del proyecto para su implementación en otras áreas de producción y para integrar nuevas funcionalidades y futuras modificaciones/mejorías; así como demostrar la capacidad del software seleccionado para resolver problemas de índoles similares dentro de Gutis. Para ello se cuenta con un tiempo de 16 semanas y el apoyo de los ingenieros de mantenimiento en Gutis y del profesor tutor a cargo de la supervisión de este proyecto (Ing. Carlos Piedra Santamaría).

Capítulo 2 Descripción del proceso productivo en Gutis Ltda.

La planta de producción farmacéutica de Gutis ubicada en Pavas, se estructuró a modo de distinguir entre tres grandes áreas de producción. Cabe destacarse que debido a la naturaleza de los productos producidos por la empresa (de índole farmacéuticos), estas tres grandes áreas de producción pueden cubrirse desde una perspectiva química en donde se detallan todos los procesos de formulación necesarios para conseguir los medicamentos o desde una perspectiva ingenieril, en donde se detalle el flujo de la materia prima hasta conseguir los productos terminados. En esta descripción que se pretende dar con la intención de contextualizar el proyecto aquí por realizar, se realizará desde una perspectiva ingenieril.

- Líquidos e inyectables: Se producen todos los productos o medicamentos que son de presentación líquida (inyectables y medicamentos líquidos de vía oral).
- Sólidos y semisólidos: Se producen la mayoría de medicamentos comprimidos de vía oral (tabletas) y semisólidos (supositorios y ungüentos).
- Hormonales: Se producen todos los medicamentos comprimidos relacionados con aplicaciones hormonales y anticonceptivos.

Estas tres grandes áreas productivas se encuentran subdivididas por 6 procesos de producción distintos.

2.1 Procesos de Producción en Gutis Ltda.

2.1.1 Proceso de producción de inyectables

El proceso se realiza inicialmente en paralelo con dos procesos simultáneos. Por un lado sucede el dispensado de la materia prima (droga) que pasa a un tanque agitador en donde se lleva a cabo el proceso (mezcla) para la formulación del líquido inyectable según las regulaciones farmacéuticas y el tipo de inyectable que se desee. Por otro lado, se realiza un proceso de esterilización del cristal (materia prima) que conforma las ampollas en donde se almacenará el medicamento (una vez concluido el proceso de formulación). Para ello se introduce el recipiente cristalino (ampollas vacías) en un horno a una temperatura determinada para esterilizar las ampollas antes de entrar al área en donde se realizará el llenado de estas.

Posteriormente se introducen los recipientes esterilizados a una máquina llenadora, marca *Bosh*, al igual que la formulación líquida proveniente del tanque agitador. Esta máquina se encarga del llenado de las ampollas y el sellado final de estas (aplicando calor). Una vez concluido este proceso se introducen las ampollas con el medicamento en un autoclave, en donde permanece por un periodo de hasta tres días. Seguidamente se lleva a cabo un proceso de inspección y etiquetado para finalizar con un proceso de encartonado manual en su presentación final para el mercado. Todo este proceso para la producción de inyectables en Gutis se ilustra en la Figura 2.1 *Diagrama del proceso simplificado de producción de inyectables en Gutis Ltda.*

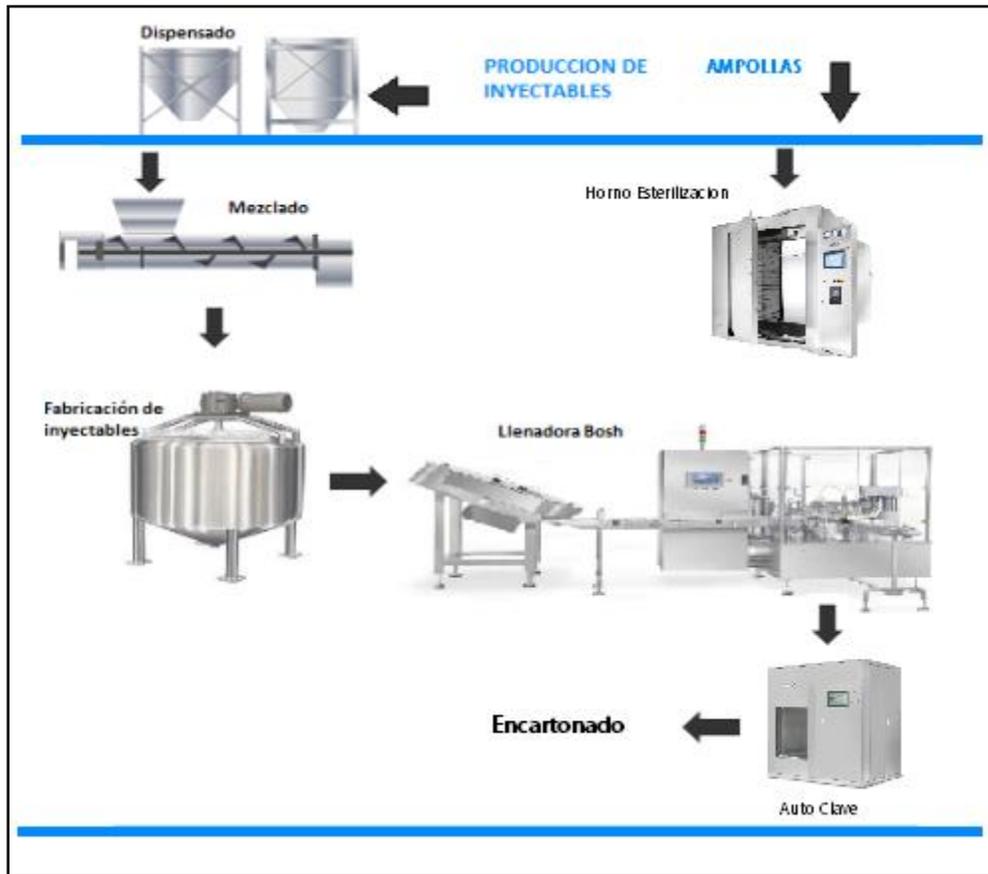


Figura 2.1 Diagrama del proceso simplificado de producción de inyectables en Gutis Ltda.

[PIXLR (Elaboración Propia, 2018)]

2.1.2 Proceso de producción y envase de medicamentos en presentación líquida

El proceso de producción y envase de medicamentos en presentación líquida en Gutis, es un proceso relativamente corto. Ya que este se realiza completamente en una sola línea de

producción. El proceso inicia con el dispensado de la materia prima (droga) que se introduce en un tanque de fabricación de líquidos en donde se realiza la formulación idónea según el medicamento que se desee. Una vez se haya concluido con la formulación se procede a introducir el medicamento en la llenadora marca *Kalish* que se encarga de el envasado, etiquetado, sellado y empaque de los medicamentos líquidos (jarabes por ejemplo) en su presentación final para el mercado. Este proceso de producción de líquidos se resume en la siguiente Figura 2.2.

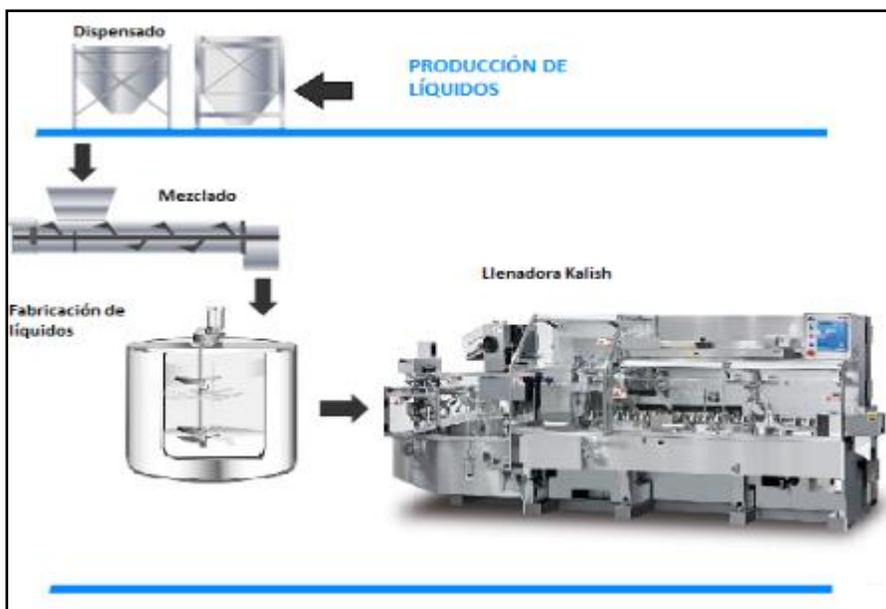


Figura 2.2 Diagrama del proceso simplificado de producción de medicamentos líquidos en Gutis Ltda.

[PIXLR (Elaboración Propia, 2018)]

2.1.3 Proceso de producción de supositorios

Tanto el proceso de producción de supositorios como el de ungüentos siguen un proceso bastante similar (por lo mismo se agrupan dentro del área de semisólidos). El proceso inicia en el dispensado de la materia prima (droga) que se transporta a un gran tanque mezclador, de la marca *Unimix*, en donde se realiza todo el proceso de la mezcla necesario para conseguir el medicamento en su consistencia y formulación adecuada. Una vez concluida la mezcla se introduce en la empacadora marca *Sarong* en donde se finaliza el proceso de llenado con el recipiente que conforma el supositorio para conseguir los supositorios en su presentación final. La Figura 2.3 muestra este proceso.

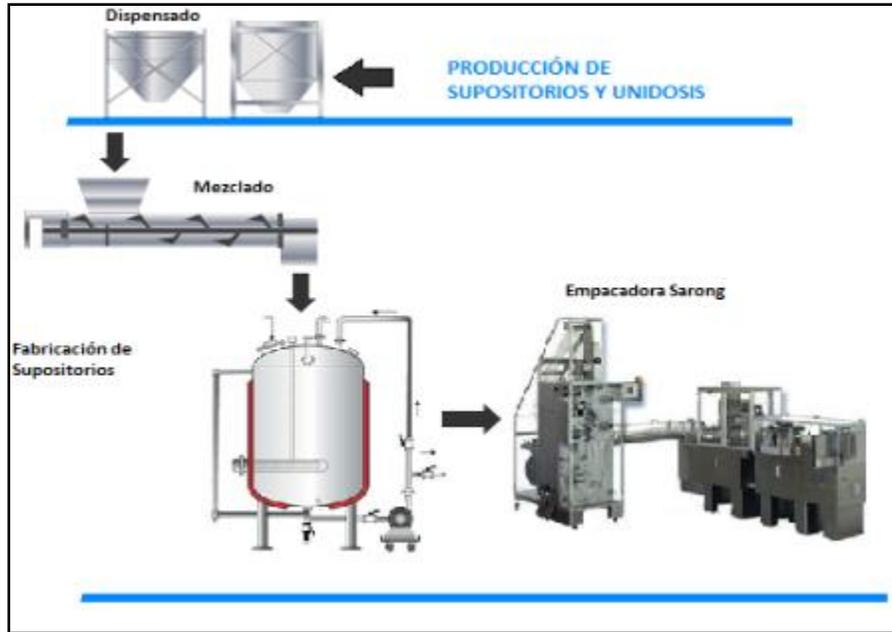


Figura 2.3 Diagrama del proceso simplificado de producción de supositorios en Gutis Ltda.

[PIXLR (Elaboración Propia, 2018)]

2.1.4 Proceso de producción de ungüentos

Como se mencionó anteriormente, el proceso de producción de ungüentos en Gutis, resulta bastante similar al proceso de producción de supositorios. La formulación de los ungüentos se realiza igualmente en los tanques mezcladores *Unimix* (con la configuración adecuada para la fabricación del medicamento deseado), y posteriormente se introduce en una máquina llenadora (marca *Sarong*) en donde se llenan los recipientes que contienen el medicamento en su versión final para el mercado. Este proceso se resume en el diagrama simplificado de la Figura 2.4.

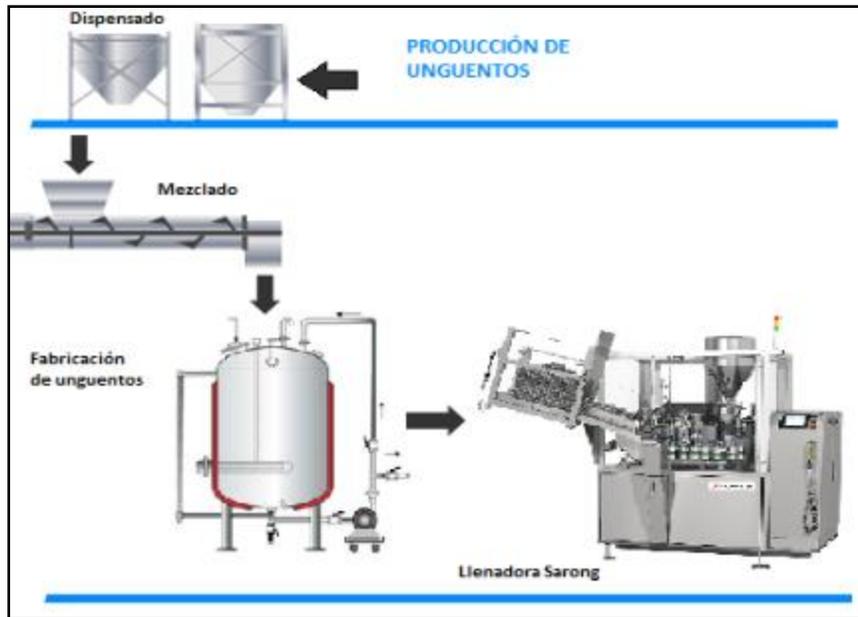


Figura 2.4 Diagrama del proceso simplificado de producción de ungüentos en Gutis Ltda.

[PIXLR (Elaboración Propia, 2018)]

2.1.5 Proceso de producción de tabletas

El proceso de producción de tabletas forma parte del área de sólidos en Gutis y se puede dar de dos formas distintas. Se puede obtener de un proceso de granulación en seco o granulación húmeda. Ambas inician con el dispensado de la materia prima, posteriormente en el caso de la granulación húmeda se realiza el mezclado de la materia prima y se introduce en un lecho fluido (equipo marca *Huttlin* en la planta de producción en Pavas). Posteriormente se pasa la mezcla obtenida a un molino marca *Bohle* en donde se consigue la granulación fina. Si se trata de una granulación seca se introduce la materia prima a una máquina de granulación seca (equipo marca *Fitzpatrick*) en donde se obtiene una granulación fina.

Seguidamente, e independientemente del proceso de granulación utilizado, se transporta la granulación hasta la máquina tableteadora marca *Fette*, en donde se consigue compactar la granulación en forma de tableta sólida. Estas tabletas luego se introducen en una máquina recubridora marca *Bohle*, en donde se les aplica una capa recubridora que protege el medicamento y le da su color de acuerdo a su presentación comercial. Por último las tabletas se transportan a las máquinas blisteadoras y empaquetadoras marca *Uhlmann* en donde se empaqueta cada tableta dentro de un blíster y cartón con el formato correspondiente al medicamento que se trate con intervención

mínima de operarios. Logrando hasta 300 blísteres por min. Este proceso se muestra de manera simplificada en la Figura 2.5.

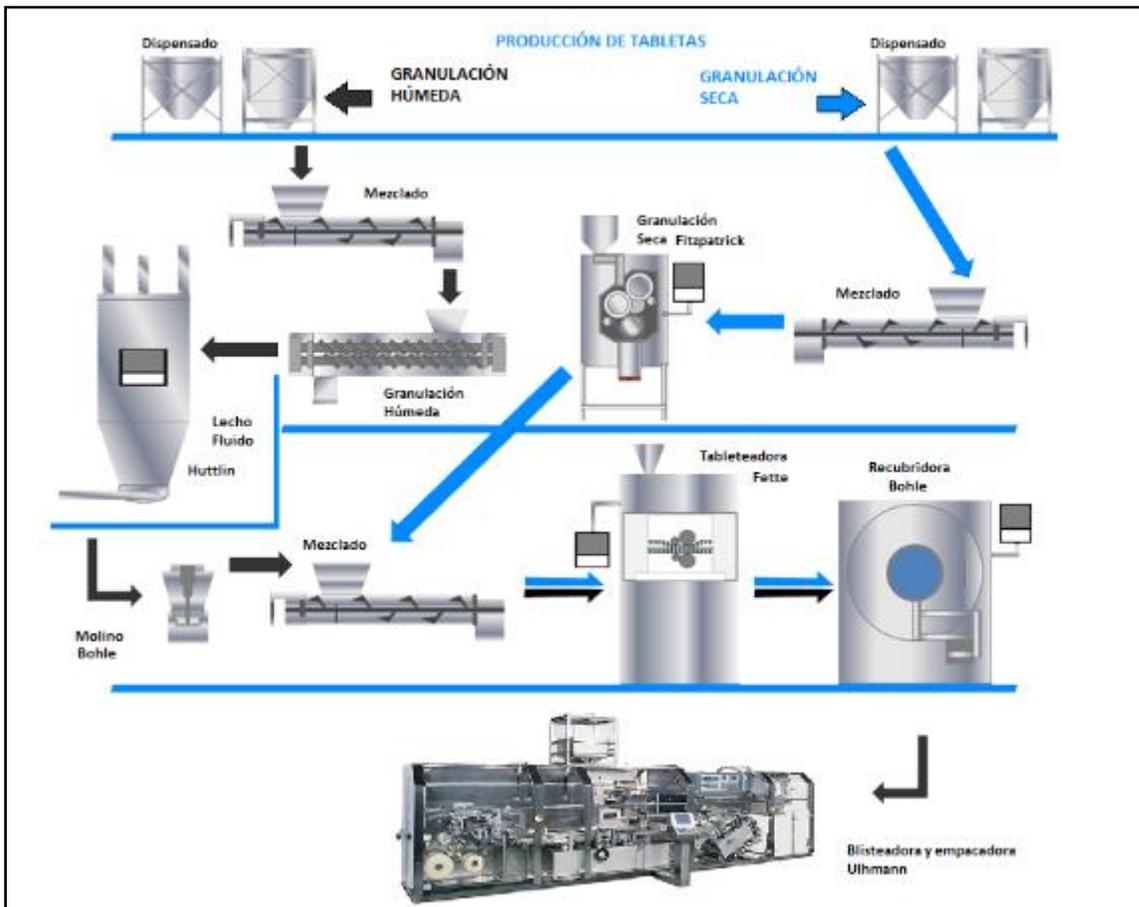


Figura 2.5 Diagrama del proceso simplificado de producción de tabletas en Gutis Ltda.

[PIXLR (Elaboración Propia, 2018)]

2.1.6 Proceso de producción de medicamentos hormonales

Este proceso es exactamente igual que el proceso de producción de tabletas, con la diferencia de que al tratarse de medicamentos en donde el ingrediente activo es hormonal, toda su producción requiere estar aislada de las demás áreas de producción y se debe evitar al máximo el contacto con los operarios (se prohíbe la intervención de operarios o ingenieras en el área de producción).

2.2 Características de las máquinas blisteadoras marca Uhlmann

El proyecto se plantea a modo de proyecto piloto sobre los equipos de blisteado del área de sólidos dentro del proceso de producción de tabletas. Estos corresponden a dos líneas de producción distintas con equipos de la marca alemana *Uhlmann*, por lo que a lo interno de la empresa se les suele llamar como “*Uhlmann 1*” y “*Uhlmann 2*”. Por ello y con la finalidad de determinar las variables de interés, así como la propuesta de indicadores de desempeño del equipo, se debe conocer el funcionamiento básico, al igual que las características principales de este.

Ambas líneas cuentan con maquinaria diseñada exclusivamente para el blisteo de productos farmacéuticos y construida conforme toda la normativa aplicable según la directiva de máquinas 2006/42/CE (CITA:Manual de *Uhlmann* otorgado a Gutis).

La placa de datos o de características de las máquinas de *Uhlmann* se encuentra siempre en el costado lateral izquierdo de estas, y contiene los datos principales del equipo y esenciales para su conexión. Se sigue la estructura que se muestra en la Figura 2.6.

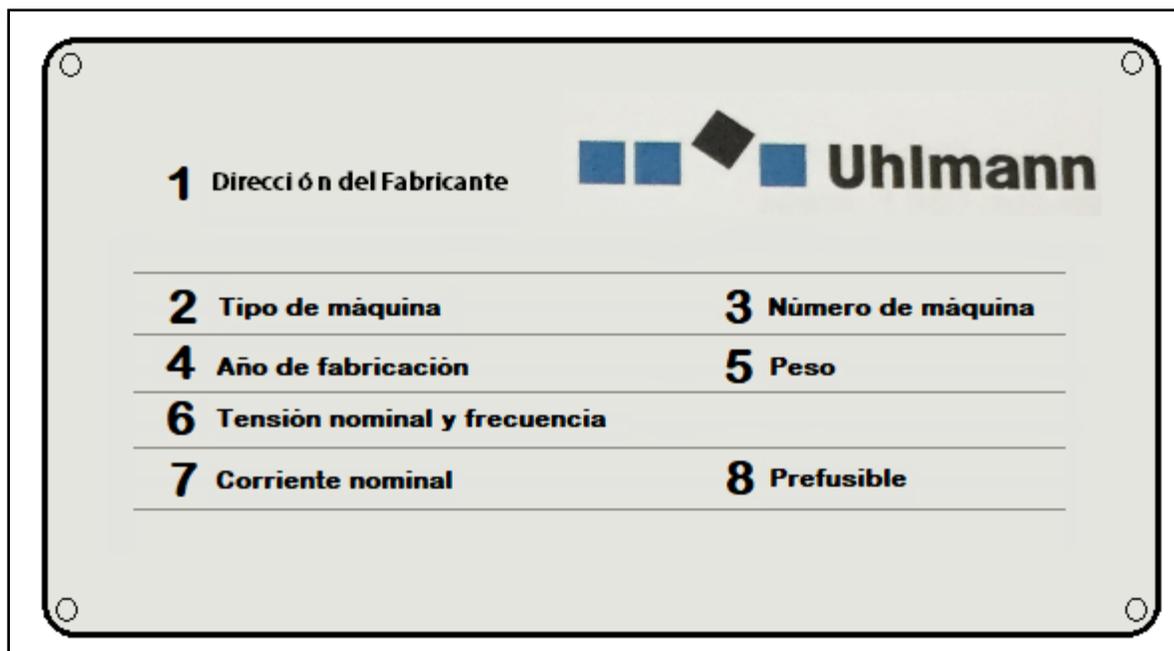


Figura 2.6 Estructura de la placa de datos de maquinaria *Uhlmann*.

[PIXLR (Elaboración Propia, 2018)]

Las principales características de la máquina se resumen en la Tabla 2.1.

Tabla 2.1 Características de las líneas blisteadoras y empacadoras de *Uhlmann (Blister line BEC 300)*.

Condiciones de servicio	
Temperatura ambiente	18°C - 25°C
Humedad relativa	40% - 50%
Dimensiones físicas	
Longitud blisteadora	Aprox. 3.880 m
Altura blisteadora	Aprox. 2.240 m
Ancho blisteadora	Aprox. 1.281 m
Peso blisteadora	Aprox. 2500 kg
Carga superficial	2.5 kN/m ²
Conexión Sistema Eléctrico blisteadora	
Tensión de Conexión	3/N/PE, CA 400V, 60Hz
Protección por fusible	50 A
Valor de Conexión	15-25 kW
Conexión Sistema Neumático *	
Conexión	NW19
Presión de entrada	[6 – 10] bar
Capacidad de la máquina **	
Cantidad de blísteres	Mín. 40 blísteres/min Máx. 300 blísteres/min

[Elaboración Propia pero tomado de manual del fabricante, 2018]

Notas:

* El aire comprimido debe cumplir lo estipulado en la norma DIN ISO 8573-1 para el sector farmacéutico

** La capacidad de la máquina depende del tamaño del blíster, tipo de producto, diseño de la cavidad, forma del producto, alimentaciones, transferencias y películas por procesar.

2.3 Funcionamiento de la máquina de blísteres B 1330/1440137 Uhlmann.

La máquina fue diseñada con el objetivo de colocar el producto (tabletas) en su cobertura de plástico o aluminio (blíster) con el grabado e impresión de la información correspondiente al medicamento y al formato de su presentación comercial, para posteriormente transferirse a la máquina encargada del encartonado. Todo esto mientras se pretende reducir al mínimo la pérdida de producto (tabletas) y los residuos de material.

Para ello la máquina sigue una serie de procesos modulares (en cadena) que inician con el ingreso de la materia prima (material de la película de moldeo) y el producto, y finaliza con el blíster completamente listo transportándose a la zona de transferencia con la máquina encartonadora postconectada.

A continuación se detalla el proceso o funcionamiento de la máquina, para ello se hará referencia a los distintos componentes que conforman la máquina (según su nombre en el manual del fabricante), y por lo tanto se incluye el plano o esquema de conjunto/ensamble simplificado brindado por *Uhlmann* en su manual de fabricante. De este modo cuando se haga referencia a algún componente de la máquina se invita al lector a ubicarlo en el plano de la Figura 2.7.

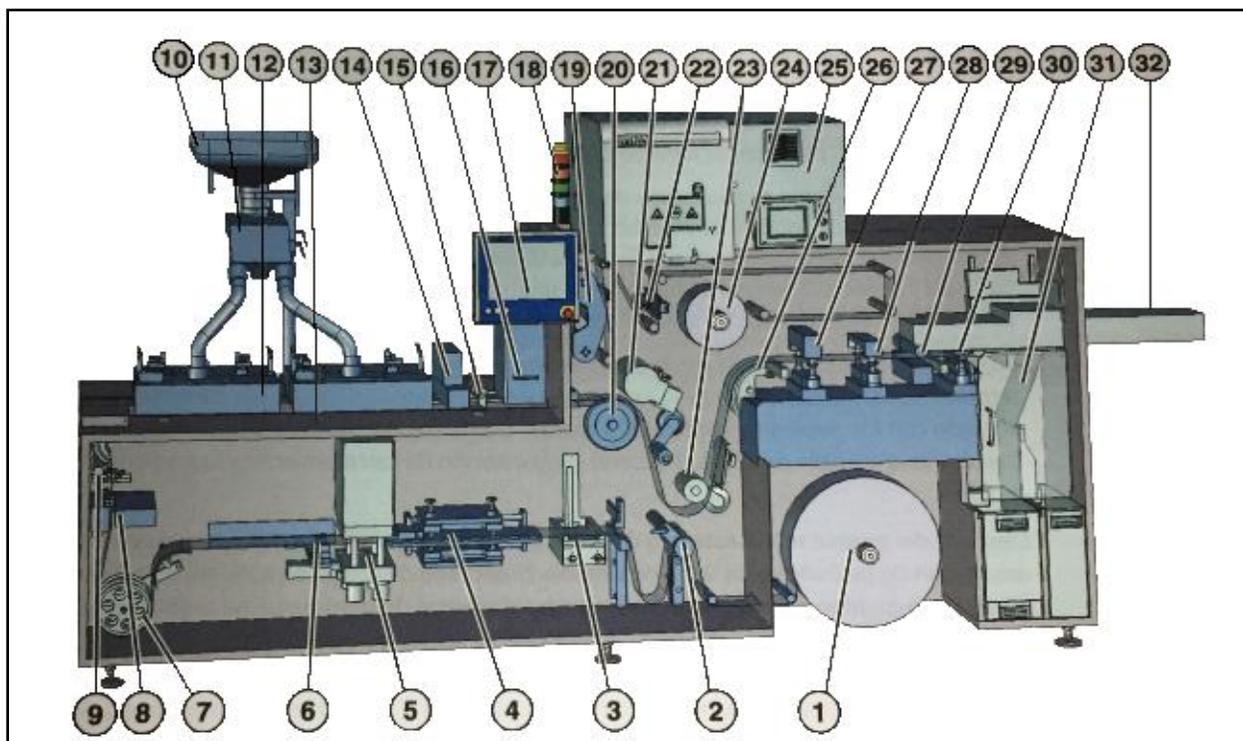


Figura 2.7 Plano de conjunto simplificado de la máquina de blísteres B 1330/1440137 de la marca *Uhlmann*.

[Manual del fabricante Uhlmann, 2010]

Donde:

- (1) Alojamiento del rollo de película de moldeo
- (2) Desbobinadora de la película de moldeo
- (3) Mesa de corte y pegado
- (4) Estación de calentamiento
- (5) Estación de moldeo
- (6) Avance posterior a estación de moldeo
- (7) Péndulo de compensación tras estación de moldeo
- (8) Detección de fisuras y poros
- (9) Detección de juntas de pegado de la película de moldeo
- (10) Tolva de llenado
- (11) Canal vibrador
- (12) Alimentación por cepillado/SIMTAP 2k
- (13) Vía de llenado
- (14) Cepillo inclinado
- (15) Tapa de seguridad
- (16) Control de producto de llenado
- (17) Sistema de operación
- (18) Columna luminosa
- (19) Control de imágenes de impresión con estiramiento
- (20) Avance de rodillo CS
- (21) Estación de sellado
- (22) Detección de juntas de pegado (para película de cubierta)
- (23) Péndulo de compensación tras estación de sellado
- (24) Alojamiento de rollo de película de cubierta
- (25) Mecanismo de impresión
- (26) Desviación de la película antes de la estación de compactado
- (27) Estación de compactado/estación de estampado
- (28) Estación de compactado/estación de perforación
- (29) Avance de rodillo CM
- (30) Estación de compactado/Estación de corte
- (31) Expulsión de blísteres defectuosos
- (32) Transferencia

El proceso inicia con el desbobinado del rollo de película de moldeo (1) mediante la Desbobinadora de la película de moldeo (2), se puede ajustar la velocidad de extracción de la película (es regulable). Se detecta siempre la reserva de película y el final de una película de moldeo, a modo de poder cortar y unir con exactitud la película de moldeo antigua con la nueva en la mesa de corte y pegado (3). Esto con la intención de aprovechar al máximo la materia prima.

Posteriormente la película de moldeo se transporta, gracias al avance posterior a la estación de moldeo (6), a través de la estación de calentamiento (4) donde se eleva la temperatura de la película de moldeo hasta alcanzar la temperatura de procesamiento. Y luego a través de la estación de moldeo (5) en donde se moldea la película de acuerdo al formato de blísteres que se deseen producir (se contemplan detalles como tamaño y forma de la tableta, o la cantidad de tabletas por blíster así como su posición entre otros).

Tras salir de la estación de moldeo la película (moldeada) se dirige hacia la vía de llenado pasando por una desviación que cambia el sentido en el que se transporta y en donde se compensan (7) los distintos movimientos que pueda tener la película debido al avance por las distintas estaciones. Además se verifican las juntas de pegado de la película (9), para bloquear la alimentación en la vía de llenado para el ciclo correspondiente y evacuar el blíster del proceso de empaquetado. En el caso de que la película de moldeo sea de aluminio, se realiza también un control de verificación (8) en donde se detectan posibles daños (fisuras y poros).

Una vez en la vía de llenado, los productos (tabletas) se depositan en la película moldeada a través de un proceso de alimentación. El proceso de alimentación consiste de la tolva de llenado (10) en donde se depositan las tabletas, y descienden a la vía de llenado a través del canal vibrador (11), para seguidamente ser depositadas en la película utilizando un sistema de alimentación por cepillado (12). Además se utiliza un cepillo inclinado (14) que se encarga tanto de cepillar las tabletas que sobresalen de la película de moldeo, como también de limpiar la película de cualquier impureza que pueda quedar dentro antes del sellado. Todo este proceso se monitorea mediante el control de producto de llenado (16) garantizando el llenado completo y correcto de los blísteres.

De manera paralela, se desbobina la película de cubierta del rollo de película de cubierta (24) mediante la combinación o interacción del avance de rodillo CS (20) y la estación de sellado (21). Además, de manera similar que con la película de moldeo, se identifica y controla la reserva de película y el final de película de cubierta (22). Se realiza una impresión sobre la película de cubierta según el formato del producto (se toma en consideración aspectos, como nombre del

medicamento, número de lote, dosis, fecha de producción y de caducidad entre otros) utilizando una impresora o mecanismo de impresión (25) y el control de imágenes de impresión con estiramiento (19), el cual permite que la imagen impresa selle de manera adecuada y precisa sobre el blíster.

Inmediatamente después se realiza el sellado (21), uniendo la película moldeada (llena) con la película de cubierta (con la impresión). La cinta de película sellada se transporta pasando por un bucle de compensación (23) hasta la estación de compactado (27 y 28). Es entonces el avance de rodillo CM (29) el que se encarga de transportar la cinta de película a través de la estación de compactado. Aquí se le aplica en caso de ser necesario un estampado y/o perforación a los blísteres, dependiendo del formato y presentación final que deba tener el blíster (por ejemplo algún grabado de número de lote).

Finalmente se separan (cortan) los blísteres de la cinta de película en la estación de compactado/corte (30). Los blísteres defectuosos se trasladan a una posición donde son expulsados (31), esto se hace según el registro de errores MSR (dentro del sistema operativo de la máquina, el MSR almacena cada error generado durante el proceso de embalaje y en base a esta información permite identificar los blísteres defectuosos a modo de que dejen de ser embalados y se extraigan al final de la línea). Mientras que los blísteres válidos se transportan mediante un brazo de aspiración hacia la zona de transferencia con la máquina empacadora postconectada (32).

2.3.1 Elementos de señal presentes en el equipamiento de Uhlmann.

Los equipos de *Uhlmann* cuentan con un semáforo industrial (columna luminosa) previamente instalado por el fabricante. Dicha columna cuenta con 4 módulos LED tal y como se muestra en la Figura 2.8. Donde la luz roja se utiliza para indicar un paro en la máquina y puede ser continua (cuando la máquina se detuvo sin errores) o intermitente cuando la máquina se detuvo debido a algún error. La luz naranja se utiliza para indicar advertencias, únicamente se utiliza la señal luminosa continua para indicar una advertencia por control de alimentación a la máquina. Por otro lado la luz verde se utiliza para indicar el funcionamiento correcto de la máquina cuando es una luz continua, mientras que una luz intermitente es indicativa de que la máquina se encuentra en espera de solucionar averías en alguna máquina postconectada.



Figura 2.8 Columna Luminosa Equipos Uhlmann.

[Manual del fabricante Uhlmann, 2010]

Capítulo 3 Estado del Arte.

La vida y evolución del ser humano han sido marcadas o definidas a partir de las grandes invenciones y desarrollos tecnológicos a lo largo de la historia. Como muestra de ello, los últimos tres siglos han sido dominados o de algún modo moldeados por algún desarrollo tecnológico y/o invención. En el siglo XVIII fueron los grandes sistemas mecánicos acompañados de la Primera Revolución Industrial en donde la sociedad vio el cambio de una economía rural basada en actividades agrícolas a una economía urbana basada en actividades industriales mecanizadas (De Vries J, 1994). El siglo XIX vio la continuación de la industrialización y tuvo el impacto de la máquina de vapor introduciéndose las primeras técnicas de control automático en la industria. Para el siglo 20 la tecnología clave se desarrolló en torno a la adquisición (sensado), procesamiento y distribución de la información, acompañada con la revolución tecnológica que significo la digitalización de la información y la red de Internet.

Las tendencias del siglo actual se ven dominadas por la revolución 4.0 y la tecnología de la información de los productos inteligentes conectados. Esto ha dado paso a la aplicación de nuevos conceptos y nuevas áreas de investigación, tales como el IoT (Internet de las cosas o *Internet of things* en inglés), la inteligencia artificial (AI por sus siglas en inglés), la minería de datos y el análisis de la enorme cantidad de información que se dispone en la actualidad de una gran variedad de fuentes de almacenamiento de datos. Esto, a su vez, sugiere un cambio poderoso y global que atravesamos actualmente y que trasciende únicamente de una transformación a nivel técnico a una evolución con impacto social y económico que impactará en el modo de hacer negocios y en la estructura de una empresa (Porter M, 2015).

Estimaciones y tendencias en el mercado indican a una explosión en la cantidad de dispositivos y conexiones que requieren *software*. De acuerdo con PwC (PricewaterhouseCoopers, la segunda compañía de servicios más grande a nivel mundial), se calcula que habrá un gasto de alrededor de \$6T (6 trillones de dólares americanos en el sistema inglés) entre el año 2015 y el año 2020 asociados a soluciones de IoT. Además prevé que para el año 2020 las empresas invertirán hasta \$832 billones (sistema inglés) en tecnologías de IoT (PwC, 2017). Estas estimaciones se muestran en la Figura 3.1. Además en el año 2011 Ericsson y Cisco estimaron que para el año 2020 unos 50 billones de dispositivos estarían conectados a Internet, esto ha impulsado aún más el desarrollo de sistemas/plataformas que incursionen en estas tecnologías, tal y como lo señalan

Derhamy, Eliasson y Delsing (2015) en su revisión de las plataformas de IoT existentes en el mercado actual.

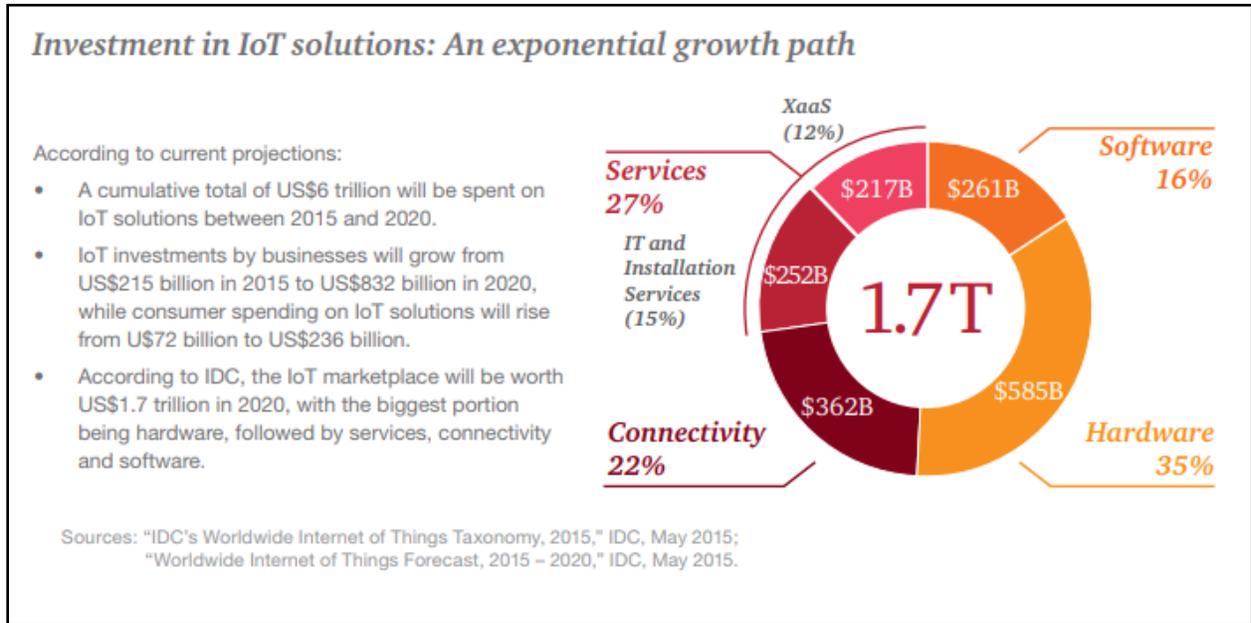


Figura 3.1 Rédito previsto a nivel mundial para tecnologías de IoT para el año 2020.

[PWC, 2017]

Como resultado, las empresas deberán demostrar su capacidad para adaptarse a los cambios que se desarrollan en la actualidad en el contexto de la Cuarta Revolución Industrial. Ya que, y son varios los autores que lo señalan entre ellos James Heppelmann en la revisión económica de Harvard para el año 2015 o Cristina Fonseca en el fórum económico mundial en el año 2017, “si continuamos comportándonos como si el mundo actual fuese el mismo que hace una década, el resultado no será prometedor”.

Por ello, al tratarse de un proyecto sumergido completamente en el contexto de Industria 4.0 y en busca de la conectividad de los equipos de producción con la participación de nuevos agentes en una plataforma conectada a la red que permita el monitoreo remoto así como el análisis de la información, se opta por tomar inspiración en las tendencias actuales del mercado.

Más específicamente en el mercado de SCADAs en LAMEA (Latinoamérica-Medio Oriente-África) donde se pronostica una tasa anual compuesta de crecimiento (CAGR) de un 7.5% para el periodo del 2018 al 2024 (KBVResearch, 2018), esta representa la tasa de crecimiento en 5 años en que la inversión aumentará, considerando que cada año crezca a la misma tasa, y a

diferencia del promedio de retorno anual (AAR) es una mejor medida para el retorno de una inversión en el tiempo (Wagner H, 2015). Mientras que el mismo reporte de KBV Research (el cual prácticamente coincide en cifras con el reporte de análisis de oportunidad global y pronóstico para la industria en el periodo 2017-2023 de la compañía Allied Market Research) señala que las soluciones relacionadas con sistemas SCADA alcanzaran un valor en el mercado de unos \$43.1 billones (sistema inglés) para el año 2024, de los cuales \$2369 millones serán del mercado farmacéutico y de químicos.

Desde el año 2014 un 8% de los usuarios finales a nivel global de sistemas SCADAs corresponden a la industria farmacéutica, como se observa en la Figura 3.2.

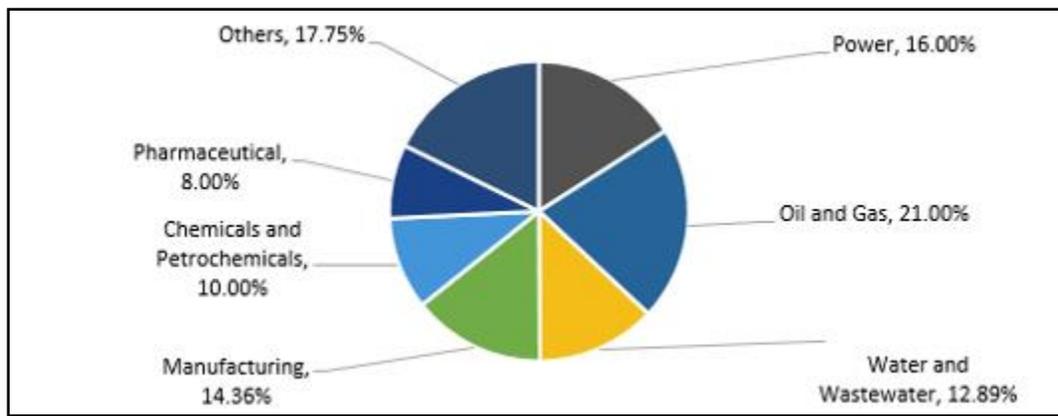


Figura 3.2 Mercado Global de SCADAs según usuario final para el año 2014.

[Techanvio, 2015]

Capítulo 4 : Marco teórico

4.1 Sistemas de comunicación electrónicos.

El objetivo primordial de un sistema de comunicación electrónico consiste en la transmisión de información analógica o digital (procesada) de un punto a otro. Para ello se requiere de una estructura básica que consiste de un transmisor, un medio de transmisión o canal y un receptor, tal y como se ilustra en la Figura 4.1. (W. Tomasi, 2003)

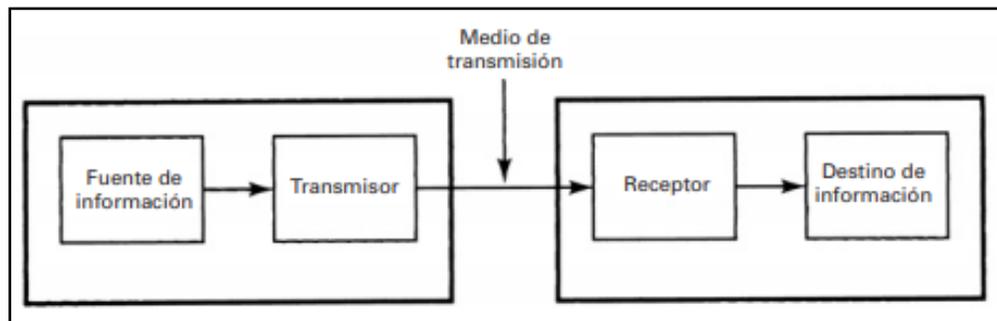


Figura 4.1 Diagrama de bloques simplificado de un sistema de comunicación electrónica.

[W. Tomasi, 2003]

El transmisor se encarga de convertir una señal de mensaje a partir de la fuente de información en una forma adecuada según el medio de transmisión. El canal se encarga de transportar las señales desde el transmisor hasta el receptor (tiende a deteriorar la calidad de la señal original). Por último, el receptor actúa sobre la señal recibida a modo de reconstruirla a su forma original para ser interpretada por algún destinatario. (S, Haykin, 2001)

A Samuel Morse se le atribuye la invención del primer sistema electrónico de comunicación en el año 1837, el telégrafo. El cual era capaz de transferir información en forma de puntos y rayas entre un transmisor y un receptor sencillo a través de un conductor metálico, utilizando el principio de inducción electromagnética. Sin embargo es hasta el año 1894 cuando Guillermo Marconi transmite información a través de la atmósfera utilizando señales de radiofrecuencia, cuando se logra establecer una comunicación inalámbrica (W. Tomasi, 2003).

Las dos limitaciones más importantes en el funcionamiento de un sistema de comunicaciones son el ancho de banda y el ruido (W. Tomasi, 2003). El ancho de banda de una señal se define como la diferencia entre las frecuencias máxima y mínima contenidas en la información, mientras que el ancho de banda del canal se define como la diferencia entre la

frecuencia máxima y mínima que pueden pasar por el canal. El ancho de banda del canal debe ser igual o mayor que el ancho de banda de la señal (información), esto para que logren pasar todas las frecuencias importantes de la información. De este modo, el ancho de banda influye directamente sobre la cantidad de información que se puede propagar en un determinado tiempo en un sistema de transmisión. Esta relación se plasma en la Ley de Hartley, en donde se indica que la capacidad de información es una función lineal y directamente proporcional tanto al ancho de banda como al tiempo de transmisión, ver *Ecuación 4-1*.

$$I \propto B \times t$$

Ecuación 4-1

Donde: I: capacidad de información

B: ancho de banda (Hertz)

t: tiempo de transmisión (segundos)

En los sistemas de comunicación electrónicos, el ruido eléctrico se define como “cualquier energía eléctrica indeseable que queda entre la banda de paso de la señal” (W. Tomasi, 2003, p.34). La presencia de ruido en un sistema de comunicación es capaz de altera la señal principal, tal y como se puede observar en el ejemplo de la Figura 4.2, en donde la señal de color negro muestra una señal sinusoidal sin ruido, mientras que la señal en rojo muestra el efecto de la presencia de ruido sobre la misma señal sinusoidal. La presencia de ruido puede tener consecuencias prácticamente despreciables o por el contrario ser catastróficas dependiendo de la aplicación en la que suceda y/o de la intensidad de este.

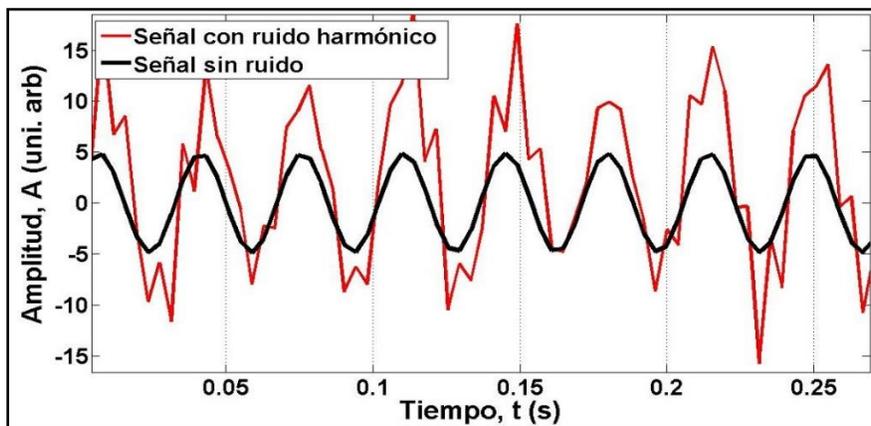


Figura 4.2 Efecto del ruido sobre una señal.

[V. Torres, 2011]

El ruido puede clasificarse según la dependencia o relación existente entre la señal que se desea transmitir y la presencia de ruido. De este modo se distinguen dos clasificaciones distintas: ruido no correlacionado y ruido correlacionado.

El primero hace alusión a cualquier ruido que esté presente independientemente de la existencia o transmisión de la señal. Se categoriza a la vez en ruido externo cuando la fuente del ruido, su origen, sucede fuera del circuito de transmisión; por ejemplo el ruido por electricidad estática o ruido atmosférico, ruido por actividad solar o ruido extraterrestre, ruido industrial o ruido producido por el hombre, interferencia armónica por otros equipos. Se categoriza como ruido interno cuando la perturbación se genera en el circuito de transmisión; por ejemplo el ruido térmico también conocido como ruido blanco que se origina debido al movimiento de los electrones dentro del conductor producto de la agitación térmica de estos (W. Tomasi, 2003, p.34-40).

El ruido correlacionado es dependiente de la existencia de al menos una señal de transmisión, en otras palabras solo es posible que ocurra este tipo de ruido cuando hay una señal de entrada. Es una consecuencia de la amplificación no lineal. Dado que todos los amplificadores son elementos no lineales en cierto grado, toda amplificación de señal produce distorsión no lineal.

4.2 Redes

Con la explosión tecnológica que se ha atravesado en el último siglo y los avances en sistemas micro-electrónicos, la incursión de la nanotecnología para la elaboración de microprocesadores potentes, la masificación de circuitos integrados y sistemas embebidos, y el impacto de los avances en las tecnologías de comunicación e interconexión de dispositivos han transformado completamente el concepto de lo que se entiende como un sistema de cómputo y el modo en que se organizan. De manera que “el concepto una vez dominante del centro de cómputo como un salón con una sola gran computadora a la que los usuarios llevaban su trabajo para procesarlo es ahora totalmente obsoleto” (Tanenbaum, 2012, p.2).

Esto ha sido reemplazado por un modelo basado en un gran número de computadoras con diferentes capacidades interconectadas capaces de resolver tareas en conjunto (incluso en los centros modernos de datos se cuenta con una enorme cantidad de servidores, clústeres y unidades de procesamiento individuales que resuelven tareas complejas en conjunto). A estos sistemas compuestos por conjuntos de dispositivos autónomos e interconectados se les conoce como una red de computadores o una red de equipos.

Las redes, sin importar el tamaño de estas, tienen como principales objetivos o finalidades los siguientes:

- Compartir recursos, información e instrucciones con otros equipos o usuarios finales.
- Respalda la información en caso de fallos en algún equipo (creación de las bases de datos).
- Potenciar la globalización y la conectividad (capacidad de conectarse a redes remotas).
- Facilitar el acceso a información remota o desconocida y la capacidad de compartirla con otros dispositivos fácilmente.

Por lo general se suele utilizar ya sea la tecnología de transmisión o la escala para clasificar las distintas redes de computadoras/equipos. Los sistemas de comunicación electrónicos se pueden diseñar para establecer el flujo de transmisión de datos en un único sentido o en ambos sentidos (ya sea ambos sentidos a la vez o no). Cuando el flujo de datos se realiza únicamente en un sentido se le conoce como modo de transmisión Símplex. En caso de que el flujo sea bidireccional se distinguen al menos dos tipos de modos de transmisión, según la simultaneidad o no de las transmisiones. De modo que en el funcionamiento Semidúplex las transmisiones se pueden realizar en ambos sentidos pero no al mismo tiempo (por ejemplo un sistema PTT, *push-to-talk*), mientras que en el modo de transmisión Dúplex Total se pueden realizar transmisiones bidireccionales al mismo momento. (W. Tomasi, 2003)

A la vez, de manera general se pueden describir dos tipos de tecnologías de transmisión en función del tipo de conexión al enlace (medio de transmisión entre los dispositivos):

Enlaces de difusión (*broadcast/multicast*): utilizan un único canal de comunicación el cual es compartido por más de dos dispositivos en red. En otras palabras, los dispositivos de la red comparten el mismo enlace. Cuando una máquina envía un mensaje o paquete, este pasa por todas las demás, y por lo mismo se suele utilizar un campo de dirección dentro del paquete que especifica el destinatario. Este campo se verifica por cada equipo en la red, y en caso de existir coincidencia procesa el resto del paquete, en caso contrario lo ignora. Cuando un equipo desea compartir un paquete con todo el resto de los equipos en la red se suele utilizar un código pre-asignado en el campo de dirección y se conoce a esta acción como difusión (*broadcasting*). En algunos sistemas de difusión también es posible realizar transmisiones que se encuentren dirigidas únicamente a un subconjunto de equipos dentro de la red. Para ello se suele utilizar una cantidad de bits definida en

el campo de dirección para contener un número de grupo, y cada equipo dentro de la red puede “suscribirse” a un grupo o más. De este modo cuando se envía un paquete que está dirigido a un grupo, este será procesado por todos los equipos suscritos a este (A, Tanenbaum, 2012, p.15). A esta acción se le conoce como multidifusión (*Multicasting*).

Enlaces de punto a punto: Se trata de una conexión directa entre pares de equipos individuales. Se le conoce también como redes de unidifusión cuando se da la transmisión directa con un emisor y un receptor. En otras palabras dos dispositivos de la red se encuentran conectados mediante un enlace directo entre ellos. Esto implica que en caso de tratarse de una red con más dispositivos y por lo tanto múltiples enlaces punto a punto, un mensaje tendrá que pasar por una cantidad de dispositivos antes de llegar a su destinatario correcto (P, Gil, 2010, p.18).

Finalmente, también se suele clasificar los tipos de red según su escala y topología. En cuanto a escala Andrew S Tanenbaum propone la clasificación que se observa en la Figura 4.3, según la cercanía entre dispositivos conectados o pertenecientes a la misma red (área de cobertura de la red). Si bien es cierto el autor hace referencia a Internet como una clasificación de mayor escala para una red, se debe hacer la aclaración que Internet realmente es un ejemplo de lo que se conoce como una interred (la conexión de dos o más redes).

Distancia entre procesadores	Procesadores ubicados en el (la) mismo(a)	Ejemplo
1 m	Metro cuadrado	Red de área personal
10 m	Cuarto	
100 m	Edificio	Red de área local
1 km	Campus	
10 km	Ciudad	Red de área metropolitana
100 km	País	Red de área amplia
1000 km	Continente	
10000 km	Planeta	Internet

Figura 4.3 Clasificación de las redes según escala.

[Tanenbaum, 2012]

Las redes de área local (LAN's) por lo general son redes de propiedad privada que se utilizan para conectar dispositivos personales o estaciones de trabajo que se encuentren

relativamente cerca (un máximo de pocos kilómetros de separación) y transportar información dentro de las empresas, edificios o fábricas. Por su naturaleza se encuentra restringidas en cuanto a su tamaño y por lo tanto se limita el tiempo de transmisión en el peor de los casos. De este modo la mayoría de LAN's se diseñan tradicionalmente para transmitir a velocidades de 10 a 100 Mbps (aunque las redes de área local más nuevas son capaces de funcionar hasta a velocidades de 10 Gbps), con un retardo bajo en el nivel de los micro o nanosegundos.

4.2.1 Topologías de Red

Por lo general suelen utilizar alguna tecnología de transmisión de difusión en varias topologías físicas. Estas definen el modo de enlazar los dispositivos físicos en una red. Algunas topologías clásicas son: en bus, en anillo, en estrella, en malla y en árbol (A, Perpinan, 2004).

Topología en Bus: Se cuenta únicamente con un único canal de comunicación (enlace) que conecta a todos los dispositivos de la red, tal como se observa en la Figura 4.4. Presenta una principal desventaja que consiste en que en caso de que se dañe el cable en algún punto del enlace o bus, se daña la transmisión de información para todos los dispositivos dentro de la red. Además al contarse con un único enlace es posible que se presenten problemas de tráfico como colisiones.

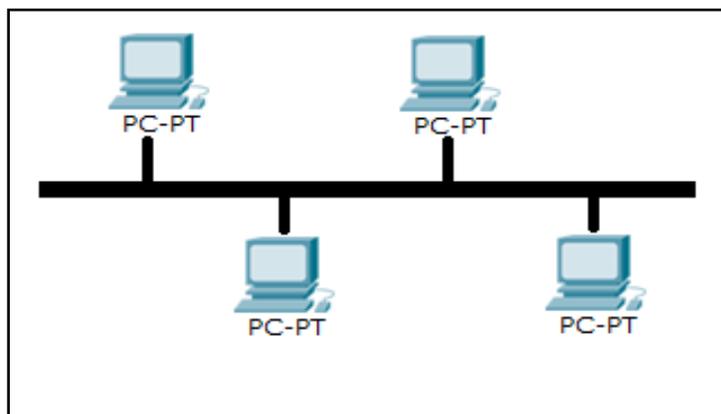


Figura 4.4 Topología en Bus de una red de dispositivos.

[CISCO PACKET TRACER (Elaboración Propia, 2018)]

Topología anillo: En este tipo de red, los dispositivos se conectan en un bucle cerrado (anillo) de modo que todos los nodos de la red se encuentran conectados uno con otro formando una cadena que finaliza con el último dispositivo compartiendo nodo con el primero, esto se ilustra en la Figura 4.5. El mensaje o la información que se desea transmitir viaja a través de la red, así

cada dispositivo que reciba el mensaje verifica la dirección de destino y en caso de coincidir acepta el mensaje. Pero en caso de que no coincidan se regenera la señal y se pasa al nodo siguiente de la red (siguiendo una dirección preestablecida). Esto supone a su vez una ventaja y una desventaja, la primera de ellas siendo que, gracias a la regeneración de la señal en cada nodo, este tipo de redes permite cubrir distancias superiores en comparación con una red en bus. Mientras que la principal desventaja de esta topología consiste en la lentitud en la transferencia de datos que sucede a partir de esta regeneración y retransmisión dada en cada nodo de la red. De igual modo una desventaja o advertencia que se debe tener presente si se utiliza una topología de anillo tradicional es precisamente que en caso de que se dañe algún nodo de la red esta dejará de funcionar correctamente afectando el flujo de información en la red.

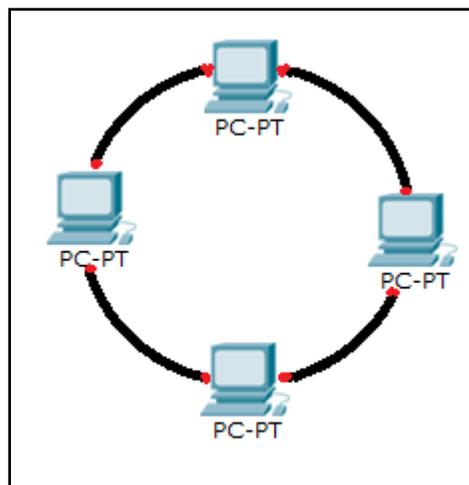


Figura 4.5 Topología en Anillo de una red de dispositivos.

[CISCO PACKET TRACER (Elaboración Propia, 2018)].

Topología estrella: En este tipo de red se utiliza un elemento central de conexiones (*Hub* o concentrador) a la cual se conectan todos los nodos de la red (Montúfar L, 2006, p.90). Un concentrador es un dispositivo electrónico pasivo (sin software) que permite la comunicación entre dispositivos para crear una red. Este concentrador se encarga de administrar el flujo de información dentro de la red. Se puede utilizar un elemento central con mayores capacidades y con software integrado que regule el flujo de información, evitando colisiones, tal como un conmutador (*switch*) o un enrutador como en el caso que se muestra en la Figura 4.6. Esta topología solventa el problema de desconectar algún dispositivo o que suceda un fallo en algún nodo de la red sin que esta falle o afecte a otro nodo. Por lo mismo con este tipo de topología se aumenta la fiabilidad de la red,

aunque se debe aclarar que siempre se es susceptible a un fallo en el concentrador que resulte en un colapso de la red.

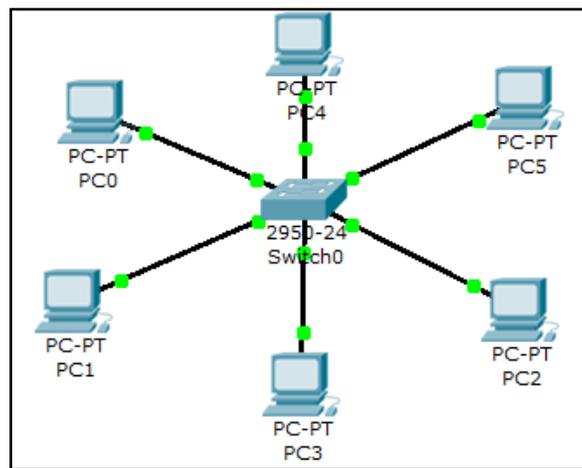


Figura 4.6 Topología en Estrella de una red de dispositivos.

[CISCO PACKET TRACER (Elaboración Propia, 2018)].

Topología en malla: Este tipo de topología se diferencia de las demás ya que todos los nodos se encuentran conectados entre sí con todos, ver Figura 4.7. Esto permite la transmisión de información de un nodo a otro a través de varias rutas dentro de la red, lo que elimina la posibilidad de que se presenten interrupciones y otorga redundancia a la red y sin la existencia de un nodo central o un concentrador. En caso de que falle algún enlace o nodo la red sigue siendo funcional. La principal desventaja de utilizar esta topología para crear una red, consiste en la gran cantidad de medios (canales, cable, etc.) para los enlaces y la enorme cantidad de conexiones con los enlaces por cada nodo que supone una red de gran tamaño o con muchos dispositivos dentro de la red. Por ello resulta conveniente únicamente para redes pequeñas con pocos nodos.

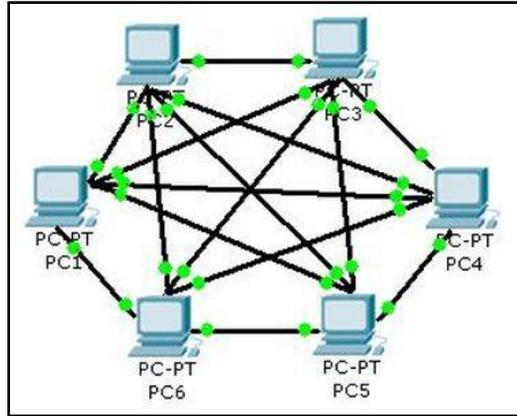


Figura 4.7 Topología en malla de una red de dispositivos.

[CISCO PACKET TRACER (Elaboración Propia, 2018)].

Topología en árbol: Es similar a una combinación de varias topologías en estrella, con la diferencia de que no se cuenta con un concentrador central. En cambio se utiliza una topología jerárquica que parte de un nodo de enlace troncal (con un *hub* o *switch*) y desde el cual se desprenden o ramifican el resto de nodos de la red, los cuales a su vez contienen concentradores o conmutadores secundarios que permiten la expansión de la red y de nuevas ramas conectadas en configuración de bus, esto se observa claramente en la Figura 4.8. En segmentos individuales se utiliza cableado punto a punto por lo que en caso de que suceda un fallo en algún enlace, la red se mantendrá funcional (siempre y cuando el fallo no se de en el segmento principal de ella).

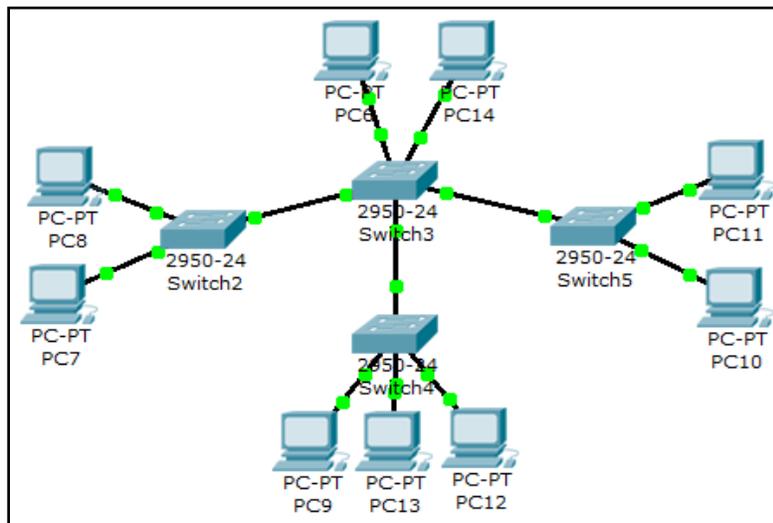


Figura 4.8 Topología en árbol de una red de dispositivos.

[CISCO PACKET TRACER (Elaboración Propia, 2018)].

4.3 Software de red

Hasta el momento se han definido algunos conceptos básicos del hardware de las redes de computadoras y el modo de clasificarlas según una serie de criterios. Sin embargo no se ha explicado los mecanismos por los cuales las redes son capaces de transferir información, ni su organización interna para el manejo de datos.

Con la intención de reducir la complejidad de las redes y facilitar su estructuración interna, la mayoría de redes se organizan en lo que se denominan capas o niveles. Estas se desarrollan de manera modular, de modo que cada capa se construye a partir de otra (similar a los pisos de un edificio, donde el segundo piso se construye a partir del primer piso que le sirve de soporte y el tercer piso a partir del segundo) y en donde cada capa inferior le ofrece ciertos servicios a las capas superiores a través de una interfaz de capa, pero les oculta los detalles de forma relativos a como se implementan estos servicios (A, Tanenbaum, 2012, p.25-46). Queda claro entonces que debe de existir una interfaz entre cada par de capas adyacentes que defina las operaciones y servicios que ofrecen las capas inferiores a sus capas superiores inmediatas.

Para que se establezca una comunicación entre la capa n de una máquina y la capa n de otra máquina debe existir una serie de normas o convenciones que regulen este proceso de comunicación conocidas como un protocolo de la capa n . Un protocolo de comunicación, es un conjunto de reglas que rigen el formato y el significado de los paquetes, o mensajes, que se intercambian entidades iguales en una misma capa (A, Tanenbaum, 2012, p.30).

Realmente la transferencia de datos no se da directamente entre las entidades iguales, más bien la información se transmite por medio de las interfaces de capa a las capas inmediatamente inferiores, regidas por un protocolo correspondiente según sea la capa y en donde se suele agregar un encabezado al mensaje, hasta llegar a la capa más baja en donde se encuentra el medio físico y se realiza la transmisión a la otra máquina. La Figura 4.9 muestra el modelo básico de la estructura de una red con dos máquinas y la comunicación que se establece entre ellas, nótese también que las líneas punteadas representan el flujo virtual (no real) de los datos por transmitir, mientras que las líneas sólidas representan el flujo real de la información.

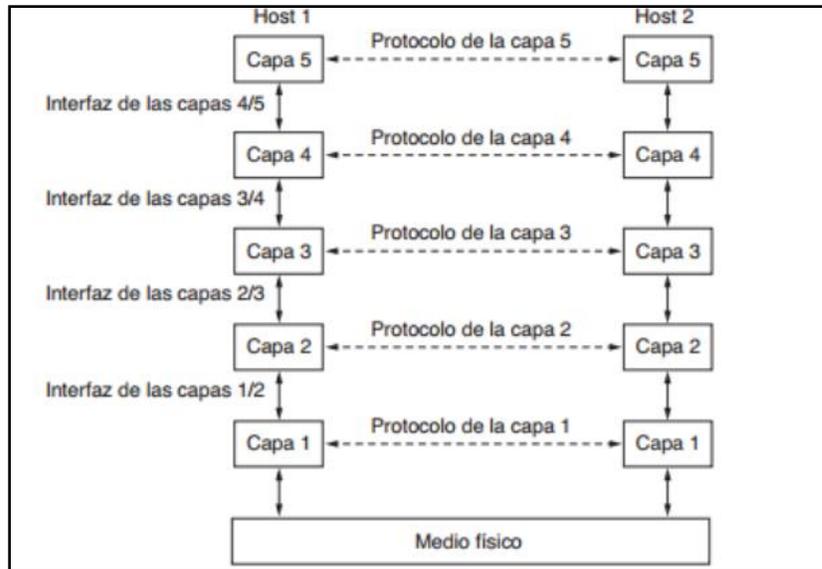


Figura 4.9 Estructura de red entre dos dispositivos, capas, protocolos e interfaces.

[A, Tanenbaum, 2012]

4.4 Telemetría

La telemetría es un proceso automático de comunicación, mediante el cual resulta posible realizar lecturas y adquisición de datos en lugares remotos o inclusive de puntos inaccesibles y transmitir esta información a dispositivos para su monitorización y en algunos casos su control.

4.5 SCADA

Un SCADA (por sus siglas en inglés: Supervisory Control and Data Acquisition) es un sistema con elementos de *Software* y *Hardware* que permite a las organizaciones Industriales monitorear, almacenar y procesar información en tiempo real, permitiendo además controlar procesos industriales de forma local o inclusive de manera remota (Rake R, 2018).

Este tipo de sistemas permite a los usuarios finales interactuar de manera directa con dispositivos de campo, tales como sensores, motores y máquinas a través de interfaces humano-máquina (HMI) que en la actualidad, y debido a la explosión del IoT y de los dispositivos móviles, pueden ser desde un teléfono celular o una tableta hasta una pantalla táctil o inclusive un computador portátil. Además en la actualidad la mayoría de estos sistemas cuentan con algún tipo de base de datos (o posibilidad de conexión a una) para el almacenamiento, procesamiento y

análisis de los datos recopilados. Por lo que en el contexto actual, tal como lo señala Rake (en el reporte de la perspectiva del mercado global de los sistemas SCADA para los años 2017 al 2024 de la compañía investigadora y asesora Allied Market Research) debido a los avances tecnológicos y el creciente interés de otros actores del mercado para la investigación y desarrollo de sensores inalámbricos, nuevos protocolos de comunicación ligeros y el rápido crecimiento del *Big Data* acompañado de la analítica han mejorado considerablemente la eficiencia de los sistemas SCADA. Y pronostican una mayor oportunidad de expansión en el mercado, llegando inclusive a incursionar en la inteligencia artificial (IA) y el auto-aprendizaje (*machine-learning*).

4.5.1 Ventajas de un sistema SCADA

Rake (2018) enumera las principales ventajas de un sistema SCADA para la industria en las siguientes:

- Ayudan a mantener al máximo la eficiencia dentro de una organización.
- Facilitan y promueven el procesamiento de información para la toma de decisiones más inteligentes.
- Comunican errores, variaciones o inclusive pronostican fallos, ayudando a mitigar el tiempo no productivo debido a paros.
- Facilitan el diagnóstico de problemas en equipos previniendo daños en estos.

4.5.2 Estructura de un sistema SCADA

La estructura de un sistema SCADA depende de cada solución que se vaya a realizar, sin embargo se puede generalizar a que cumplen una arquitectura maestro-esclavo. En donde la estación central (servidor/maestro) se comunica con el resto de dispositivos (esclavos/clientes) requiriendo de estos una serie de acciones o datos (Rodríguez A, 2007). Debe existir de igual modo algún sistema de comunicación que se encargue de transmitir la información proveniente de los diversos dispositivos que conformen el sistema hasta una estación de procesamiento y manejo de la información. De este modo la unidad central debe contar entonces con un programa de desarrollo en donde se ejecuten las acciones de monitoreo y control propias del diseño del sistema.

Por último debe de existir alguna interfaz humano-máquina (HMI) en donde se permita la interacción entre los usuarios finales y el proceso productivo y/o máquina. Cabe destacar que por lo general se puede alimentar a estos dispositivos con información proveniente de una base de datos (la cual puede o no ubicarse en el mismo equipo donde se encuentre el servidor del sistema)

o inclusive algunos sistemas SCADA ofrecen la capacidad de conectarse con otras plataformas de índole administrativas o gerenciales.

La Figura 4.10 muestra la estructura básica de un sistema SCADA.

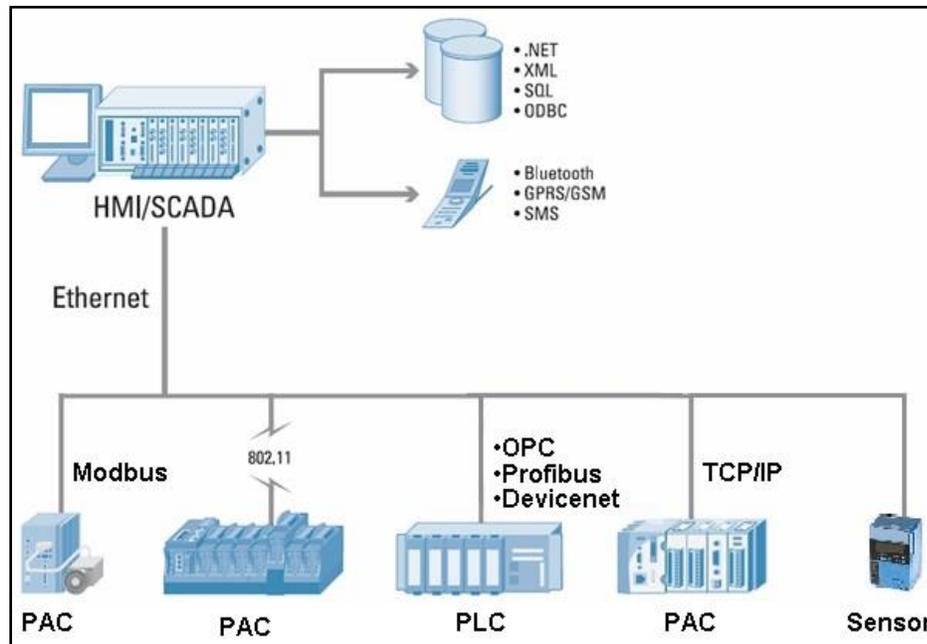


Figura 4.10 Estructura de un sistema SCADA.

[Castellanos L, 2011]

4.5.3 Protocolos de comunicación para un SCADA

Para lograr aplicaciones de automatización industrial, sistemas de SCADA, y aplicaciones de Telemetría, entre otras, se debe establecer de algún modo eficiente la transmisión de información de los dispositivos involucrados para su posterior manipulación dependiendo de la aplicación deseada. Esto evidencia la necesidad de profundizar en los protocolos de comunicación relevantes en el contexto de los sistemas SCADA actuales, que permitan explotar al máximo la interconexión y las redes de equipos (cabe recalcar que estos protocolos deben contar con la pila o serie de protocolos TCP/IP que definen la comunicación a través de Internet).

4.5.3.1 M2M (Machine to Machine)

Se refiere a la comunicación directa entre dispositivos a través de algún canal de comunicación, ya sea por medio de una conexión física (cableada) o una conexión lógica por un medio multiplexado como en el caso de las telecomunicaciones y algunas redes de cómputo. En

otras palabras, se refiere a toda comunicación punto a punto entre máquinas/dispositivos, sea esta cableada o no (*Wireless*), donde no hay intervención por medio de algún usuario o administrador humano. Este tipo de comunicación permite transmitir la información obtenida por un sensor (sensada) hacia una aplicación de software que pueda interpretar y utilizar esta información.

En sus inicios la comunicación M2M se lograba por medio de la conexión de los dispositivos de interés en alguna configuración de red que les permitiese transmitir su información a un concentrador (*Hub*) para el análisis. Posteriormente se redirigía esta información a un sistema como un computador central a través de una red independiente.

En la actualidad y debido a la expansión de las redes de IP la comunicación M2M se ha vuelto más rápida y sencilla, al mismo tiempo que se consume menos potencia, esto ha permitido la evolución de las comunicaciones M2M a un sistema de redes que pueden transmitir información en cualquier momento a un dispositivo personal de manera directa.

Las redes inalámbricas interconectadas de gran alcance, ofrecen el potencial para introducir las ventajas de las aplicaciones móviles desde el hogar, en sistemas de entretenimiento, estaciones meteorológicas e impactando inclusive en la producción y la eficiencia en varias áreas de la industria, permitiendo introducir nuevos conceptos a la vida de los productos y redefiniendo el rol del sector productivo y el concepto de competencia industrial. A partir de ellas se logra establecer contacto con los clientes, y vincular sencillamente labores de mercadotecnia con el diseño de productos y ordenes de producción, garantizando una toma de decisiones fundamentada en el análisis integral de información de todos los actores involucrados en la vida de los productos. Además estas redes permiten monitorear sistemas de manera remota, lo que puede derivar en la consecución de tiempos muertos o de paro cercanos a cero, potencializando al máximo el desempeño de las máquinas. De igual modo con esta conectividad que se pretende lograr se estimula el surgimiento de nuevas prácticas de mantenimiento tales como el mantenimiento a distancia de los equipos, sin la necesidad de desplegar una flotilla de ingenieros en sitio e inclusive el desarrollo y la ejecución de algún software de análisis que permita un mantenimiento predictivo que reconozca un fallo o un patrón de fallos antes de que ocurran o de que puedan resultar en un paro o daño catastrófico.

4.5.3.2 OPC-UA (OPC-Unified Architecture)

OPC-UA es un protocolo de comunicación M2M para la automatización industrial desarrollada por la fundación OPC. La cual es un consorcio industrial que se encarga de crear y mantener estándares para la conectividad de dispositivos y sistemas para la automatización industrial. Se encarga de especificar y regular el flujo de información entre sensores, controladores (microcontroladores, PLC, etc.), software de sistemas y dispositivos de notificación. Es un protocolo abierto (disponible gratuitamente), multiplataforma (no se limita a un único sistema operativo ni lenguaje de programación). Publicado en el año 2006, sin embargo la versión actualizada es la 1.04 (disponible desde el 22 de Noviembre del año 2017).

4.5.3.3 MQTT (Message Queue Telemetry Transport):

Es un protocolo de comunicación utilizado para la comunicación M2M orientada en el contexto del “Internet de las cosas” (IoT), orientado a la comunicación de sensores (poco ancho de banda), ideado por la multinacional IBM. Sigue una topología de estrella, con un nodo central que hace la función de servidor, también conocido como *broker*, y con capacidad para hasta 10 000 clientes (Badami V, 2016).

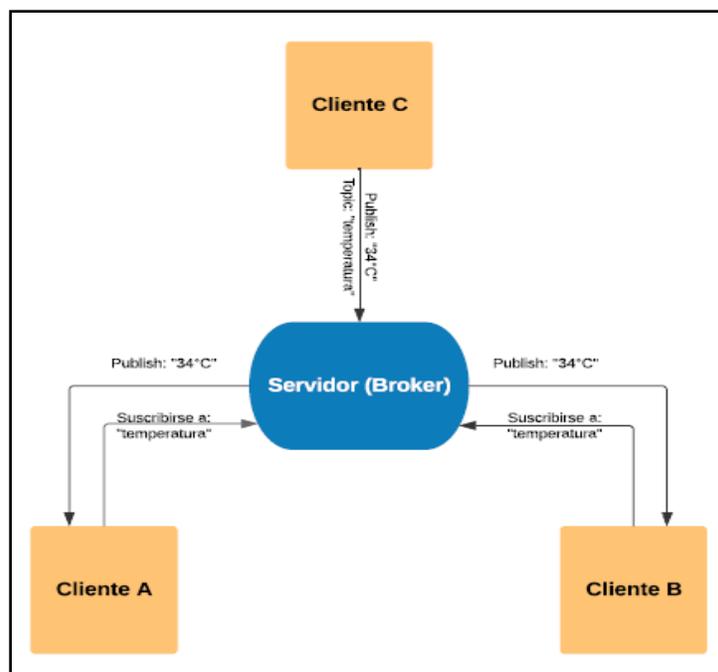


Figura 4.11 Estructura base que utiliza el protocolo MQTT.

[FLUIDCHART (Elaboración Propia, 2018)]

El *broker* es el encargado de gestionar la red y transmitir los mensajes, que dentro del protocolo se conocen como *Application Messages* (AM). Además el canal se debe mantener activo constantemente (por lo que los clientes periódicamente deben enviar un paquete PINGREQ y esperar la confirmación del *broker* PINGRESP). La comunicación se basa en la creación de temas o *Topics* que el cliente crea (al momento de realizar una publicación) y los nodos que desean recibirlo deben suscribirse a este. Como cliente se entiende a todo aquel programa o dispositivo que establece conexión en la red con el servidor. Dentro de las capacidades del cliente, este puede publicar información a través de mensajes (AM) que otros clientes les pueden interesar. Puede también suscribirse a algún *Topic* de interés para solicitar información que esté interesado en recibir, de igual modo que desuscribirse para dejar de recibir información de algún tema. Por último un cliente tiene la potestad de desconectarse del servidor.

La comunicación puede darse uno a uno o uno a muchos (*broadcast* o *multicast* no recuerdo) y puede estar cifrada. Un *Topic* se representa mediante una cadena y tiene una estructura jerárquica. Cada mensaje transportado por MQTT tiene asociado una calidad de servicio (QoS) y un nombre de tema (*Topic Name*). Existen tres calidades de servicio distintas dentro del protocolo MQTT:

- “*At most Once*” (QoS=0): Los paquetes de mensajes son enviados de acorde a los mejores esfuerzos del entorno de operación. Puede ocurrir pérdida de mensajes. Útil en aplicaciones donde la transmisión exitosa de información en todo momento no resulta crítico y en el muestreo de los datos se da de modo que la pérdida de un dato se sustituya en poco tiempo por la lectura del siguiente valor. Por ejemplo un sensor de temperatura en un hogar, en donde el efecto de la pérdida de información de una lectura individual no resulta crítico, ya que la próxima lectura será publicada próximamente.
- “*At least once*” (QoS=1): Se asegura el arribo de los mensajes, pero existe la posibilidad de generar duplicados.
- “*Exactly once*” (QoS=2): Se asegura el arribo de los mensajes exactamente una sola vez. Puede resultar necesaria en aplicaciones de mucha fidelidad o de información sensible, tales como una transacción bancaria.

Los nombres de temas (*Topic Names*) tienen la funcionalidad de actuar como una etiqueta asociada a los mensajes de aplicación (AM), la cual se compara con las suscripciones que

administra el servidor y en caso de encontrarse una suscripción coincidente, el servidor se encarga de enviar una copia del mensaje (AM) a cada cliente coincidente. Los nombres de temas, al igual que los filtros de temas (expresión dentro de una suscripción para indicar interés en uno o más temas) siguen una estructura jerárquica que cumple una nomenclatura definida por el protocolo MQTT. En esta se indica que puede utilizarse el *foward slash* ('/') como separador de nivel de temas y de este modo crear una cadena o árbol de temas. Por ejemplo se puede tener las siguientes ramas de temas vinculadas: [Gutis/Producción/Tabletas/Uhlmann 1] o también [Gutis/Producción/Tabletas/Uhlmann 2].

Se puede utilizar el símbolo '#' para emparejar cualquier cantidad de niveles a un tema. Por ejemplo si un cliente se suscribe a [Gutis/Producción/#] recibirá mensajes publicados que utilicen temas como:

[Gutis/Producción]
[Gutis/Producción/Tabletas]
[Gutis/Producción/Líquidos/Festel]
Etc...

Además se puede utilizar el símbolo '+' para emparejar únicamente un nivel de un tema. Por ejemplo si un cliente se suscribe a [Gutis/Producción/+] recibirá mensajes publicados que utilicen temas como:

[Gutis/Producción/Tabletas] y [Gutis/Producción/Líquidos] pero no de [Gutis/Producción/Tabletas/Uhlmann].

4.5.4 Tags

Algunas plataformas de SCADAs utilizan puntos de información (similares a una variable de algún lenguaje de programación), las cuales pueden tener valores estáticos o dinámicos (ser modificadas por el programa, usuario o algún dispositivo). Estos "tags" pueden venir de la dirección OPC de un dispositivo conectado al sistema, una expresión, o alguna consulta SQL de una base de datos. Ellos permiten un diseño orientado a objetos, facilitan el manejo de la información y permiten generar históricos de alguna variable de interés.

4.6 Bases de Datos

Como ya se ha mencionado anteriormente, en el contexto competitivo actual la información y en especial el manejo efectivo de esta (su análisis) es uno de los factores más críticos dentro de una empresa. Con una enorme cantidad de datos disponibles, provenientes de variadas fuentes (sensores, decisiones administrativas, operarios, fechas), una empresa enfrenta el reto de adquirir esta potencial información de manera precisa, confiable y a tiempo. Un sistema de base de datos es una herramienta que simplifica estas labores (Singh S, 2011)

4.6.1 Datos

Según Singh (2011) los datos se pueden definir como un hecho/valor conocido que puede ser almacenado y que tiene algún significado implícito. De manera que un dato es un único hecho acerca de algo que tiene importancia según el contexto en el que se encuentre.

4.6.2 Información

A menudo se suele confundir datos con información, sin embargo esta última es precisamente el resultado de procesar algunos datos relacionados entre sí que juntos revelan un mensaje con significado. Hoy en día los sistemas de bases de datos pueden contener datos, información o ambos.

4.6.3 Entidades

Las entidades son los objetos físicos o eventos reales al que se le desea almacenar información.

4.6.4 Atributos

Los atributos son propiedades o valores o características propias de una entidad.

4.6.5 Relaciones de datos

Las relaciones de datos son aquellas formas en que diferentes entidades se vinculan o entrelazan. Se distinguen tres tipos de relaciones:

- Relaciones uno a uno (1:1): Un registro en una tabla (entidad) se puede asociar únicamente con solo un registro de otra tabla (entidad). En la Figura 4.12 se muestra un

ejemplo de esta relación, en donde cada profesor tiene un número de identificación (id) único y cada número de identificación es asignado únicamente a un solo profesor.

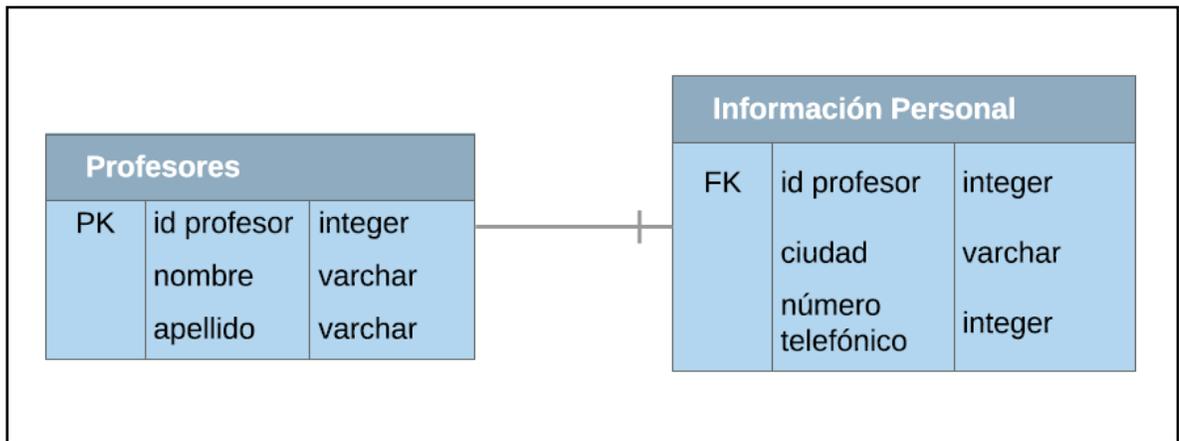


Figura 4.12 Diagrama de relación de Entidades (ER) Uno-a-Uno.

[FLUIDCHART (Elaboración Propia, 2018)]

- Relaciones uno a muchos (1:m): Un registro en una tabla (entidad) puede asociarse con uno o más registros de otra tabla (entidad). En la Figura 4.13 se muestra un ejemplo de esta relación, en donde cada profesor puede dar más de un curso pero todos los cursos solo pueden ser impartidos por un único respectivo profesor. De este modo la relación retornara registros relacionados (posiblemente más de uno) cuando el valor en id profesor en la tabla de Cursos sea igual al valor de id profesor en la tabla de Profesores.

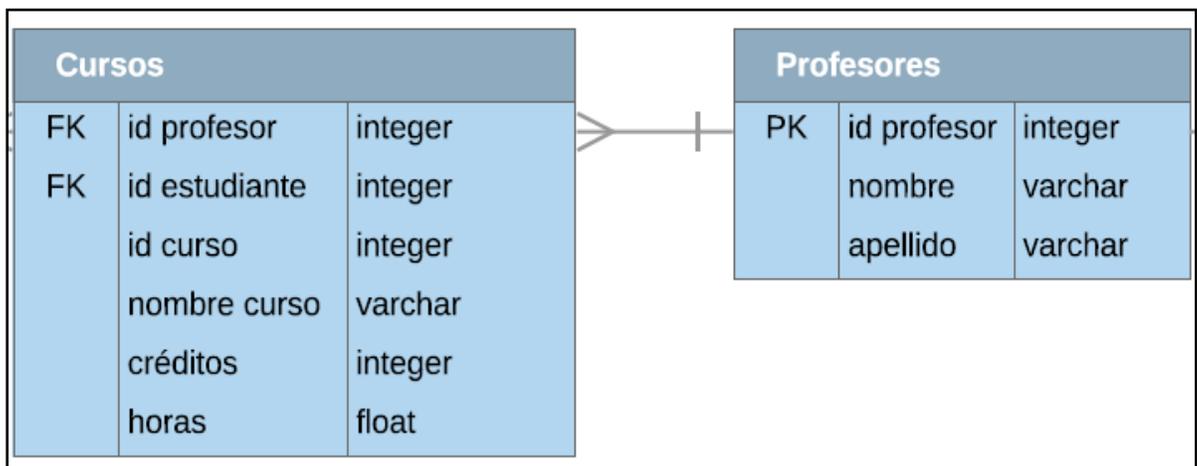


Figura 4.13 Diagrama de relación de Entidades (ER) Uno-a-Muchos.

[FLUIDCHART (Elaboración Propia, 2018)]

- Relaciones muchos a muchos (n:m): Suceden cuando múltiples registros en una tabla (entidad) se asocian con múltiples registros de otra. En la figura Figura 4.14 se muestra un ejemplo de esta relación. Un estudiante puede asistir a varios cursos, mientras que un curso puede estar integrado por varios estudiantes.

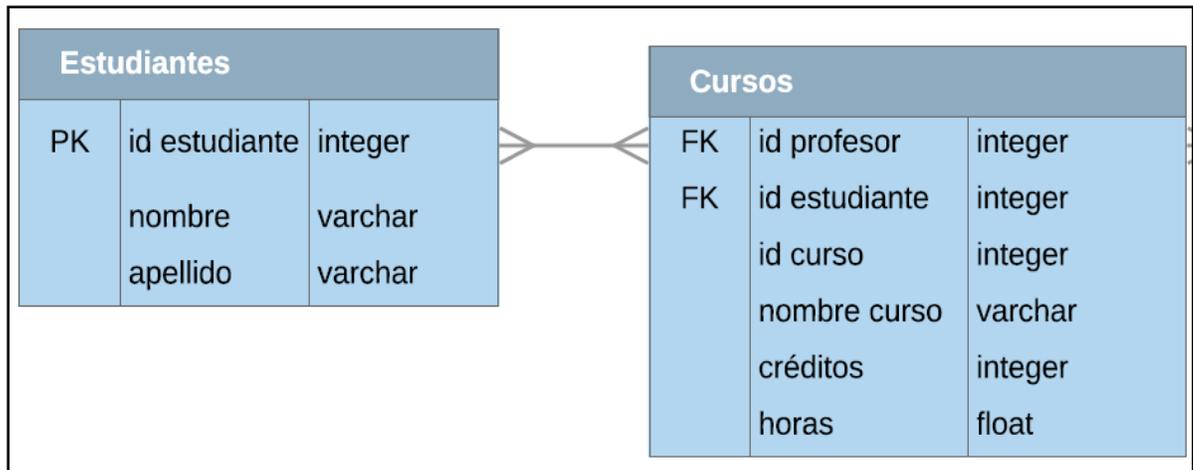


Figura 4.14 Diagrama de relación de Entidades (ER) Muchos-a-Muchos.

[FLUIDCHART (Elaboración Propia, 2018)]

4.6.6 Llaves

El ítem que se utiliza en los sistemas de bases de datos para identificar un registro (Singh S, 2011)

4.6.7 Definición de una Base de Datos.

Singh (2011) define una base de datos como una colección de datos almacenados juntos y que poseen alguna relación lógica, diseñada para obtener información valiosa.

4.7 Semáforo Industrial

Un semáforo industrial, también conocido como columna luminosa es un elemento compuesto generalmente por dos o más módulos de luz led que se utilizan en entornos industriales para indicar de manera visual el estado de una máquina o proceso, así como alertar en caso de algún problema.

Capítulo 5 Estrategia de Solución

Para la solución del proyecto se adopta una estrategia modular, en donde se resuelvan situaciones específicas del problema y en donde la resolución de una permita o facilite el desarrollo de la siguiente. De modo que al final se construya modularmente una solución capaz de conseguir los objetivos planteados para este proyecto.

5.1 Identificación de las variables, señales o estados de interés.

Una parte clave o esencial de todo proyecto es conocer que es lo que realmente se busca y desea controlar/monitorear (en otras palabras la información necesaria para el proyecto). En especial cuando se tiene un problema desconocido por la empresa, esta etapa de determinación de los principales actores resulta primordial, ya que de ello dependerá el resto del desarrollo del proyecto y por ende su éxito o fracaso final.

Con lo anterior se hace referencia a que sin importar la eficiencia del sistema diseñado y su programación, si las variables que proveen al sistema de datos no son representativas del problema entonces no se obtendrá una solución correcta. Es por esto que para la determinación de las señales se recurrió al criterio experto y experiencia de los ingenieros y técnicos en Gutis, al igual que un periodo de monitorización manual de los equipos en cuestión.

Según estadísticas adquiridas manualmente y reportadas a lo interno de Gutis se encontraron una serie de irregularidades y limitaciones en el método actual para medir la productividad en las máquinas en cuestión para este proyecto. El método actual cual consiste en sumar las horas programadas para la Uhlmann, que por cierto no toma en consideración cambios de turnos o modificaciones al horario productivo de las máquinas (en otras palabras se asumen jornadas continuas de seis de la mañana a diez de la noche de lunes a viernes, para un total de 16 horas programadas; y de seis de la mañana a dos de la tarde los sábados, para un total de 8 horas programadas).

Sumar las horas de mantenimiento requerido en cada máquina proveniente del sistema automático de avisos implementado a lo interno de la compañía, el cual consiste en órdenes abiertas para indicar alguna necesidad/problema o aviso de mantenimiento requerido en los equipos y a cargos de un responsable que debe cerrar el aviso una vez se haya resuelto la situación

a modo de registrar la duración de este evento. Sumar las horas productivas (activas) reportadas al final de la producción. Finalmente se procede a transcribir esta información en una hoja de cálculo y realizar una serie de indicadores básicos (desglose de horas programadas y cálculo de horas perdidas).

Como se mencionó este sistema se encuentra envuelto por limitaciones y defectos que se deben resolver con el sistema por diseñar. Entre ellos destaco:

- La principal limitación se debe a que toda la adquisición de esta información se hace de manera manual y se introduce en una hoja de cálculo, lo cual requiere tiempo valioso del Ingeniero en Mantenimiento Gustavo Rodríguez y está lejos de representar un método confiable para el reporte de información. Esta adquisición se debería realizar automáticamente, proveniente idealmente directo de la máquina (eliminando la intervención humana), en especial el tiempo activa real de la máquina, ya que en el método actual existe demasiada incertidumbre en dicho valor como para ser un valor confiable.
- La segunda limitante encontrada, es la calidad de la información capturada de manera manual. He aquí la importancia de la correcta identificación de las variables/señales de interés para el problema. Tómese como ejemplo la productividad reportada para la semana 31 del año en la Uhlmann 1 y Uhlmann 2. Para la Uhlmann 1 se reporta una cantidad de horas semanales programadas de 88 horas de las cuales 34.5 horas se reportan como horas productivas (horas activas) y un total de horas perdidas por mantenimiento de 8.05. Mientras que para la Uhlmann 2 se reportan de igual modo una cantidad de horas programadas de 88 horas de las cuales 56 horas se reportan como productivas con un total de 3.5 horas perdidas por mantenimiento. Estos escenarios se visualizan claramente en la gráfica de la Figura 5.1 y la Figura 5.2 respectivamente (las series en gris representan el resto de horas perdidas sin incluir los paros por mantenimiento). Esto significa apenas un rendimiento de apenas un 39.2% en la Uhlmann 1 (calculando el rendimiento como horas activa/horas programadas) y un 63.6% en la Uhlmann 2.



Figura 5.1 Desglose de tiempo programado en Uhlmann 1 para la semana 31.

[G. Rodríguez, 2018]

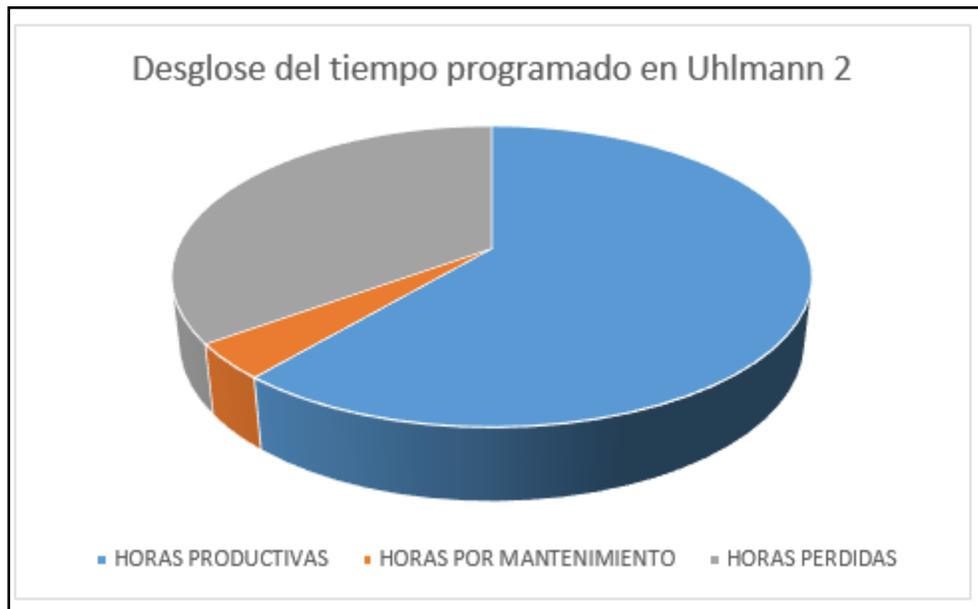


Figura 5.2 Desglose de tiempo programado en Uhlmann 2 para la semana 31.

[G. Rodríguez, 2018]

A su vez estos resultados revelan tres situaciones importantes para el desarrollo de este proyecto, por un lado evidencian la necesidad mejorar la productividad en ambas líneas. Se demuestra además que se tiene un desconocimiento absoluto de los motivos de paro que están ocasionando el resto de tiempo perdido no relacionados a mantenimiento. Los cuales según estos resultados tienen un impacto mucho mayor en la productividad de las máquinas Uhlmann. Apenas un 15% aproximado de los paros de la semana 31 en la Uhlmann 1 se dieron por mantenimiento, mientras que un 11% de los paros de esa semana para la Uhlmann 2 fueron por causas de mantenimiento. Por lo que se puede afirmar que se desconoce qué es lo que realmente está ocasionando los paros productivos y disminuyendo el rendimiento.

Por último este reporte también evidencia una gran discrepancia entre ambas líneas productivas, situación que puede atribuirse a un caso atípico (conocido en inglés como un “*statistical outlier*”) o a un motivo de fondo. Sin importar cuál de estos dos casos sea, el hecho relevante es que con la información actual es imposible distinguir entre uno o el otro y por ende se elimina cualquier posibilidad de tomar decisiones inteligentes basadas en datos.

- La tercer limitante o deficiencia del sistema actual encontrada, es que no se vincula ningún otro valor de productividad o eficiencia de la máquina, que no sean los del tiempo activa y tiempo detenida (el cual como se mencionó tampoco se realiza de un modo confiable). Por ello, actualmente no se puede construir algún otro indicador para el desempeño de las líneas productivas en Gutis que no tenga que ver con la disponibilidad de los equipos.
- El sistema actual no se distribuye a otro departamento dentro de la empresa y en caso de hacerse se hace de manera manual a través del Ingeniero Gustavo Rodríguez.
- Con el sistema actual no existe ninguna monitorización en tiempo real del estado de la máquina, por lo que en caso de un paro hay que esperar a la apertura de un aviso, o el operario puede ocasionar paros sin reportarlos y sin ser supervisado.

De estos descubrimientos tempranos se logra plasmar las primeras ideas para el diseño de un sistema SCADA con conectividad a la red que se ajuste a estas necesidades urgentes en Gutis. A partir de las limitaciones mencionadas se proponen las fuentes de información que puedan evitar

estos mismos problemas en el sistema por diseñar que se muestran en la Tabla 5.1. Otro criterio para la selección de dichas señales/fuentes de información se basó en la conveniencia de dedicar el mayor peso (responsabilidad) del proyecto en la plataforma por diseñar y no en la necesidad de equipamiento extra o modificaciones al equipo actual. Por ejemplo que la medición de los tiempos de paro y los motivos de estos se realicen en la plataforma y no en un PLC externo que envíe sus resultados finales a la plataforma, o que la selección de los tiempos programados (planificados) se realice directamente en la plataforma por desarrollar.

Al mismo tiempo se consideró la escalabilidad del proyecto y su fácil modificación para la selección de estas fuentes de información. En otras palabras se optó por tomar señales sencillas provenientes de las Uhlmann que puedan ser fácilmente transmitidas sin necesidad de que cada vez que se realice un cambio haya que modificar el diseño del sistema, al igual que, y con proyección a una eventual implementación a más máquinas de otras áreas se seleccionaron señales que pueden extraerse de la mayoría de máquinas sin importar su tipo o marca.

Tabla 5.1 Determinación de las señales de Interés para el diseño de la plataforma.

Señales Físicas (Provenientes de las máquinas Uhlmann)	
Nombre Señal	Descripción
<i>LuzRojaContinuaU1</i>	Señal proveniente de la columna luminosa de la Uhlmann 1, indicativa del estado. Indica un paro en la máquina sin errores asociados.
<i>LuzRojaIntU1</i>	Señal proveniente de la columna luminosa de la Uhlmann 1, indicativa del estado. Indica un paro en la máquina debido a un error.
<i>LuzVerdeContinuaU1</i>	Señal proveniente de la columna luminosa de la Uhlmann 1, indicativa del estado. Indica que la máquina se encuentra en funcionamiento.
<i>LuzVerdeIntU1</i>	Señal proveniente de la columna luminosa de la Uhlmann 1, indicativa del estado. Indica que la máquina se encuentra en espera de la reactivación de la máquina postconectada.
<i>LuzAmarillaU1</i>	Señal proveniente de la columna luminosa de la Uhlmann 1, indicativa del estado. Indica un paro en la máquina debido a problemas de alimentación.
<i>LuzRojaContinuaU2</i>	Señal proveniente de la columna luminosa de la Uhlmann 2, indicativa del estado. Indica un paro en la máquina sin errores asociados.
<i>LuzRojaIntU2</i>	Señal proveniente de la columna luminosa de la Uhlmann 2, indicativa del estado. Indica un paro en la máquina debido a un error.
<i>LuzVerdeContinuaU2</i>	Señal proveniente de la columna luminosa de la Uhlmann 2, indicativa del estado. Indica que la máquina se encuentra en funcionamiento.
<i>LuzVerdeIntU2</i>	Señal proveniente de la columna luminosa de la Uhlmann 2, indicativa del estado. Indica que la máquina se encuentra en espera de la reactivación de la máquina postconectada.

<i>LuzAmarillaU2</i>	Señal proveniente de la columna luminosa de la Uhlmann 2, indicativa del estado. Indica un paro en la máquina debido a problemas de alimentación.
<i>Cantidad Blisters producidos U1</i>	Señal proveniente de la Uhlmann 1, indica la cantidad de blisters producidas por la máquina.
<i>Cantidad Blisters en buen estado U1</i>	Señal proveniente de la Uhlmann 1, indica la cantidad de blisters producidas sin defectos por la máquina.
<i>Cantidad Blisters producidos U2</i>	Señal proveniente de la Uhlmann 2, indica la cantidad de blisters producidas por la máquina.
<i>Cantidad Blisters en buen estado U2</i>	Señal proveniente de la Uhlmann 2, indica la cantidad de blisters producidas sin defectos por la máquina.
Señales Generadas en la Plataforma	
<i>Paro por Material U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por espera U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro no planificado U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por ajustes U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por cambio formato U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por mantenimiento U1</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por Material U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por espera U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro no planificado U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por ajustes U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por cambio formato U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
<i>Paro por mantenimiento U2</i>	Justificación del motivo de paro, el sistema la pregunta al operario.
Señales en base de datos	
<i>Usuario</i>	El usuario que utiliza actualmente la plataforma y registra algún cambio. Tienen roles asignados.
<i>Rol</i>	Función del usuario dentro de la empresa, tiene permisos a ciertas actividades asociados
<i>Nombre de máquina</i>	Nombre de máquina que se añade al sistema
<i>Área</i>	Área dentro de Gutis a la que pertenece la máquina
<i>IdealC</i>	Ciclo Ideal de producción de una máquina. Valor brindado por el fabricante que determina el tiempo ideal en que se tarda en producir una unidad de producto.

[Elaboración Propia, 2018]

Con base en las fuentes de información señaladas en la Tabla 5.1, se procede a realizar los planteamientos de diseño del sistema. Se pretende construir una línea del tiempo que pueda ser monitoreada en tiempo real para cada máquina y que refleje su estado. De este modo se define el tiempo activo total como todo el tiempo que la máquina se encuentra funcionando correctamente y el tiempo detenida total se define como la suma de todos los paros ya sean debidos a mantenimiento, a cambios de formatos en la maquinaria, por ajustes realizados por el operario, por espera a la reanudación de una máquina postconectada, por errores en el control de alimentación de la máquina o por un paro no planificado.

Estas 6 categorías fueron seleccionadas en base a los motivos de paros que podrían suceder en las Uhlmann según consultas al departamento de Ingeniería en Gutis. Por lo tanto la plataforma deberá brindar alguna herramienta similar a la que se observa en la Figura 5.3, nótese que de este modo se logrará un monitoreo no solo del estado actual de la máquina en tiempo real sino que a la vez se expandirá la cantidad de opciones de motivos de paro brindando información 100% real de que es lo que sucede con las máquinas y como estos repercuten en el tiempo total activa de la máquina y por ello su rendimiento. Posibilitando entonces la toma de decisiones para corregir un tipo de paro reiterativo en una máquina, disminuyendo tiempos muertos y elevando la eficiencia global de la planta.

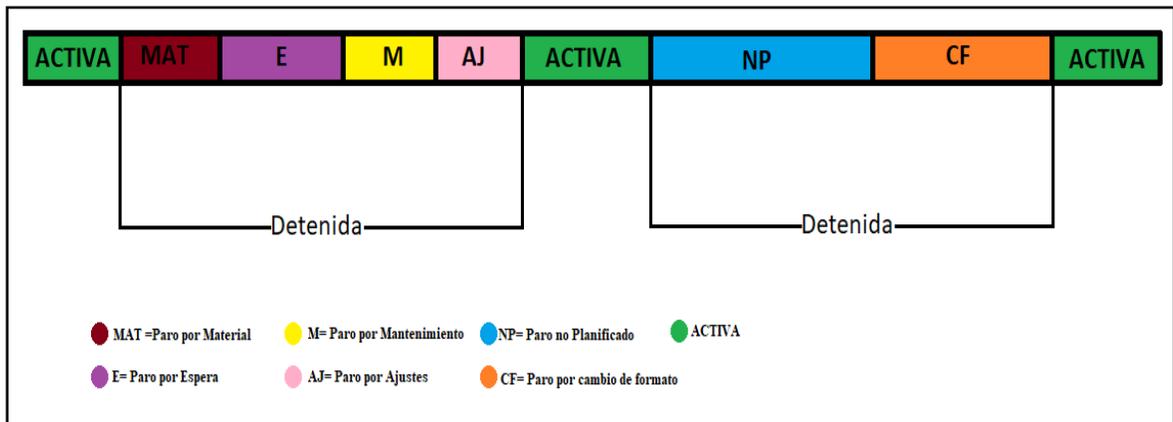


Figura 5.3 Concepto general de monitoreo para las máquinas Uhlmann 1 y Uhlmann 2.

[PIXLR (Elaboración Propia, 2018)]

Por otro lado, se definen una serie de variables que deberán provenir de una base de datos, la cual deberá ser accedida por la plataforma y modificable desde esta. Esto con la intención de poder almacenar un histórico de la información recopilada por el sistema y brindar la capacidad

de poder realizar análisis comparativos en el tiempo. Además esta base de datos cumple una función esencial para una herramienta tecnológica dentro de la industria farmacéutica (que a su vez se encuentra sujeta a las regulaciones de la FDA) y esta es que permite registrar los cambios que se realicen en la plataforma, registrando el momento exacto en que se realiza una modificación en la base de datos y el usuario responsable. Garantizando la confiabilidad del sistema y el respaldo de sus registros, facilitando su validación a través de la normativa CFR-21 parte 11.

De igual modo esta base de datos pretende otorgar a la plataforma de mayores funcionalidades como el registro de nuevas máquinas y su información que se pueda utilizar en la creación de indicadores de eficiencia en la planta (tal como el ciclo ideal de cada máquina). Inclusive posteriormente se hará uso de esta base de datos para el registro a través de la plataforma de los tiempos de producción según sea la máquina seleccionada.

5.2 Selección de indicadores de desempeño

Para lograr el diseño de una plataforma funcional, además de conocer los problemas actuales en la empresa y de haber determinado las fuentes de información necesarias para conseguir resultados satisfactorios, se debe tener claro que se persigue al final de este proyecto. De manera concisa, lo que se busca es brindar funcionalidades de monitoreo remoto, pero también igual de importante es la generación de reportes estadísticos basados en la información monitoreada de las máquinas.

En esta sección se plasman los indicadores propuestos para el proyecto que luego han de ser programados dentro de la plataforma que se seleccione. Además de la medición de los tiempos activos totales y tiempos totales en los que la máquina se encuentra detenida, resulta provechoso cuantificar la cantidad de paros que se registran durante un día.

Más aún se propone cuantificar el número de paros según el motivo que ocasiono el paro (alguno de los 6 tipos discutidos en la sección anterior). Con esto a futuro se podrá distinguir entre patrones reiterativos en alguna máquina o inclusive diagnosticar falta de capacitación de un operario respecto a otro, hasta podrá utilizarse para planificar los mantenimientos preventivos en las máquinas. Otra posibilidad que la consecución de este indicador podría ofrecer a Gutis en un futuro, es la capacidad de tomar decisiones más inteligentes en la formulación de las ordenes productivas de un día, ya que por ejemplo en un caso hipotético en el que se presenten 7 paros por cambios de formatos puede significar o un problema con los formatos utilizados o que el orden en

que se realizaron los lotes productivos potencio un mayor número de cambios de formato y por lo tanto un incremento en el tiempo no productivo en la máquina.

Otro indicador seleccionado es el OEE (*Overall Equipment Effectiveness* por sus siglas en inglés). Se seleccionó principalmente por dos motivos, el primero de ellos debido a que en el sistema propuesto se cuenta con toda la información para realizar el cálculo de este indicador. El segundo motivo es que el OEE es el estándar de oro utilizado mundialmente para medir la productividad en la industria. Este define el porcentaje de tiempo de manufactura que realmente es productivo. En otras palabras, en qué medida se fue capaz de producir únicamente productos en buen estado, al ritmo más rápido y sin tiempo perdido en paros.

Como se observa entonces el OEE toma en consideración tres factores principales: La disponibilidad del equipo, el desempeño del equipo (producir lo más rápido posible) y la calidad obtenida (porcentaje de producto en buen estado). En todo proceso productivo se tendrán ciertas pérdidas, sin embargo la intención de medir el OEE a través de la plataforma desarrollada consiste en maximizar el tiempo completamente productivo.

Algunas definiciones que deben tenerse en consideración para el cálculo de este indicador se enlistan a continuación, estas también se pueden entender de forma gráfica a partir de la Figura 5.4.

- Tiempo Total: Representa absolutamente todo el tiempo disponible en un período establecido. Por ejemplo el tiempo total de un día son 24 horas completas.
- Tiempo de producción planificado: Básicamente consiste en todo el tiempo que debería estar produciéndose al 100%. Consiste en restar las pérdidas de tiempo planificadas (como el cierre de la planta, paros por almuerzo, etc.).
- Tiempo de ejecución: Se toma en consideración las pérdidas de disponibilidad del equipo, en otras palabras los paros no planificados incluyendo los paros por ajustes o cambios de formato (en el caso del sistema de este proyecto, se toma en consideración las pérdidas en tiempo por los 6 tipos de paros propuestos anteriormente). Consiste entonces de la resta del tiempo perdido debido a alguno de estos motivos al tiempo de producción planificado.
- Tiempo de ejecución neto: Toma en consideración la disminución en el desempeño de un equipo, o sea todo aquello que fuerce al equipo a producir a un ritmo más lento que el ideal señalado por el fabricante.

- Tiempo completamente productivo: Finalmente se toma en consideración las pérdidas en la calidad de los productos durante el tiempo que la máquina estuvo activa.



Figura 5.4 Factores del OEE.

[OEE]

De manera matemática se tiene que el OEE se puede calcular como el factor de los tres factores que lo componen. De este modo la disponibilidad se calcula como se muestra en la Ecuación 5-1.

$$A = \frac{\text{Tiempo de ejecución}}{\text{Tiempo de producción planificado}}$$

Ecuación 5-1

El factor de desempeño se calcula según la Ecuación 5-2. Donde el ciclo ideal corresponde al tiempo mínimo que tarda en producirse una sola unidad de producto. Nótese que si se produce todo el tiempo de ejecución al ritmo del ciclo ideal, entonces este factor es igual a 1.

$$P = \frac{\text{Ciclo Ideal} \times \text{Total Producido}}{\text{Tiempo de ejecución}}$$

Ecuación 5-2

El factor de calidad se calcula a partir de la Ecuación 5-3 y es básicamente una relación entre el total de productos sin defectos y el total producido.

$$Q = \frac{\textit{Total en buen estado}}{\textit{Total Producido}}$$

Ecuación 5-3

Por último el indicador OEE se calcula como el producto de estos tres factores, y aunque es posible simplificar la expresión para el cálculo directo del OEE como se muestra en la Ecuación 5-4, se recomienda el cálculo de cada factor por separado. Esto debido a que cada factor revela información valiosa por sí sola sobre el proceso productivo, la cual en un cálculo directo del OEE se desconocería y hasta podría resultar engañoso.

Por ejemplo una compañía puede experimentar un aumento ligero en su OEE debido a un aumento en su factor de disponibilidad pero sufrir una disminución en su factor de calidad. En general si se calculase únicamente el OEE parecería ser una situación positiva pero dependiendo del caso la mayoría de compañías no promoverían esta acción a costas de una disminución en su calidad. Incluso podría ser el caso que debido al aumento en la disponibilidad del equipo pero a la caída en calidad se esté presentando un problema en la máquina que resulte luego en un paro y esto se refleja antes de que suceda en el aumento de productos defectuosos. Por este motivo en la plataforma se obtendrá el OEE a partir del producto de los tres factores y no a través del cálculo simplificado de la Ecuación 5-4.

$$OEE = \frac{(\textit{Total en buen estado} \times \textit{Ciclo Ideal})}{\textit{Tiempo de producción planificado}}$$

Ecuación 5-4

5.3 Diseño de la Base de datos que alimenta al sistema

Para este proyecto se utilizó MySQL de Oracle como base de datos. Esta cuenta con una versión 100 % funcional gratuita que se rige por la GLP, en donde inclusive se permite el uso de esta base en negocios o aplicaciones lucrativas, siempre y cuando no se venda una aplicación o plataforma con esta versión de MySQL empotrada (y se hace la aclaración que una plataforma comercial que tenga una conexión a esta versión de MySQL no es lo mismo que una plataforma comercial que utilice MySQL de manera embebida, y por lo tanto la primera es completamente legal).

Una parte fundamental del desarrollo del proyecto consiste en el diseño de la base de datos que respalde la plataforma desarrollada y que alimente el sistema brindándole a este de todas sus funcionalidades. Entre estas funcionalidades se destacan el fácil almacenamiento de datos y generación de históricos, la facilidad de realizar consultas a través de SQL (el cual es un lenguaje exclusivamente diseñado para el manejo de las bases de datos), la posibilidad de realizar cálculos con datos y establecer relaciones entre distintas entidades y por último potencializar la escalabilidad de la plataforma desarrollada. Permitiendo agregar nuevas máquinas o construir nuevos indicadores de manera casi que inmediata sin la necesidad de implementar nuevo código en la plataforma.

Para el diseño de la base de datos se hace primero un análisis de que es lo que se va a requerir de la base de datos y que información se deberá almacenar en esta. Dado que el problema gira entorno al monitoreo de las máquinas se pensó conveniente comenzar con una tabla dentro de la base de datos que defina las máquinas que se han integrado al sistema y sobre las cuales se realizan el resto de operaciones. Para definir una máquina en este sistema se seleccionan los siguientes atributos:

- ID: Es la identificación única de cada máquina y se genera de manera automática a lo interno de la base de datos (utilizando auto-incrementos). Se selecciona este atributo como llave o clave primaria (*primary key* o PK), ya que con base a esta es posible distinguir entre todas las filas de la tabla (es un atributo que permite distinguir un elemento dentro de la base de la tabla en la base de datos).
- Machine: Hace referencia al nombre de la máquina a lo interno de Gutis (por ejemplo Uhlmann 1 o Uhlmann 2), también es un valor único. Es un llave candidata (*candidate key* o CK), ya que podría ser utilizada para distinguir entre las filas de la tabla (distinguir un elemento), sin embargo se optó por utilizar ID como PK (criterio personal de diseño que no tiene repercusión en el comportamiento general del sistema).
- Area: Se indica a cual área, dentro de las 6 del sector productivo, pertenece la máquina. Se opta que el operario no pueda introducir manualmente el valor desde la plataforma sino que tenga que seleccionar de una lista que contiene todas las áreas disponibles en Gutis. Por ello este valor hace referencia al valor de otra tabla dentro de la base de datos denominada Area, y por lo tanto es una llave foránea (*foreign key* FK) que apunta o hace referencia a la columna Name de la tabla Area.

- Description: Espacio para breve descripción de la máquina, marca, modelo, año, etc.
- IdealCycle: Atributo dedicado para indicar el valor de ciclo ideal de la máquina dado por el fabricante.

Como se hizo mención además se creó una tabla denominada Area la cual no se puede modificar desde la plataforma desarrollada (solamente se puede leer) en donde se incluyen las 6 áreas de producción en la empresa actualmente. Al igual que una descripción de cada área y un ID que la identifica en la tabla por lo que se asigna como PK. Nótese que existe una relación de entidades uno a muchos por parte de la tabla Area a la tabla Machine. Ya que es un área puede tener múltiples máquinas asociadas, mientras que una máquina solo puede estar asociada a un área.

Se crea además una tabla denominada MachineStops en donde se planean almacenar todos los motivos de paros asociados a una máquina durante un día. Para eso se define dentro de sus atributos los 6 tipos de paro propuestos, el día al que corresponden estos registros, un ID que nuevamente se utiliza para identificar cada registro (que por lo tanto se define como PK) y se genera de manera automática, y por último el nombre de la máquina a la que corresponden estos registros. El nombre de máquina se relaciona con la tabla creada previamente Machine por lo que se define como un FK que hace referencia a la columna de Machine. Nótese que la relación existente entre Machine y MachineStops es una a muchos. Ya que una máquina puede presentar múltiples registros de paros pero un registro de paros solo puede estar vinculado (relacionado) a una sola máquina.

De manera similar se crea una tabla para el registro diario de las estadísticas monitoreadas. En esta se incluyen el tiempo total activa (RunTime), el tiempo total detenida (StopTime), la cantidad total de blísteres producidos (TotalCount), la cantidad total de blísteres en buen estado producidos (GoodCount). Además muy importante es que dado que solo debería de existir un registro diario, y para poder identificar dentro de los registros que se den no resulta posible identificar una fila a partir del nombre de la máquina (habrán muchos registros coincidentes); por lo que se opta por utilizar una combinación entre el nombre de la máquina y el día en que se da el registro para identificar cada elemento de la tabla. De modo que se utilizan dos PK, o lo que se conoce como una llave compuesta (*Composed Key*).

Además existe una relación de entidades uno a muchas entre la tabla Machine y esta tabla. Ya que una máquina puede tener múltiples registros estadísticos, pero cada registro de la tabla DailyStats solo puede asociarse a una máquina.

Finalmente se construye una tabla en la base de datos donde se almacenen los turnos de producción o jornadas laborales asociadas a una máquina. De esta tabla es de donde se procederá a través de una consulta en la plataforma calcular el tiempo de producción planificado para una máquina. Esta tabla contiene atributos correspondientes al arranque y finalización programadas de los turnos (estos valores deberán ser ingresados por el usuario desde la plataforma).

Además el sistema deberá extraer de manera automática en la plataforma el usuario que realizó el registro y el momento exacto en que se dio este (con la intención de resguardar la integridad de los datos). También se almacena en esta tabla el cálculo de la duración del turno ingresado. Se utiliza un ID de turno para identificar cada registro en la base de datos.

El nombre de la máquina que le corresponde el turno registrado es almacenado en la tabla. Este nombre es un FK que se relaciona con la columna Machine de la tabla Machine. Existe de este modo una relación de entidades uno a muchas entre la tabla Machine y esta tabla. Ya que una máquina puede tener múltiples turnos asignados, pero cada turno solo puede estar asociado a una sola máquina.

El diseño de esta base de datos se ilustra en el diagrama de relación de entidades que se muestra en la Figura 5.5. La creación de todas estas tablas se hizo a través de MySQL Workbench 8.0.12 de Oracle.

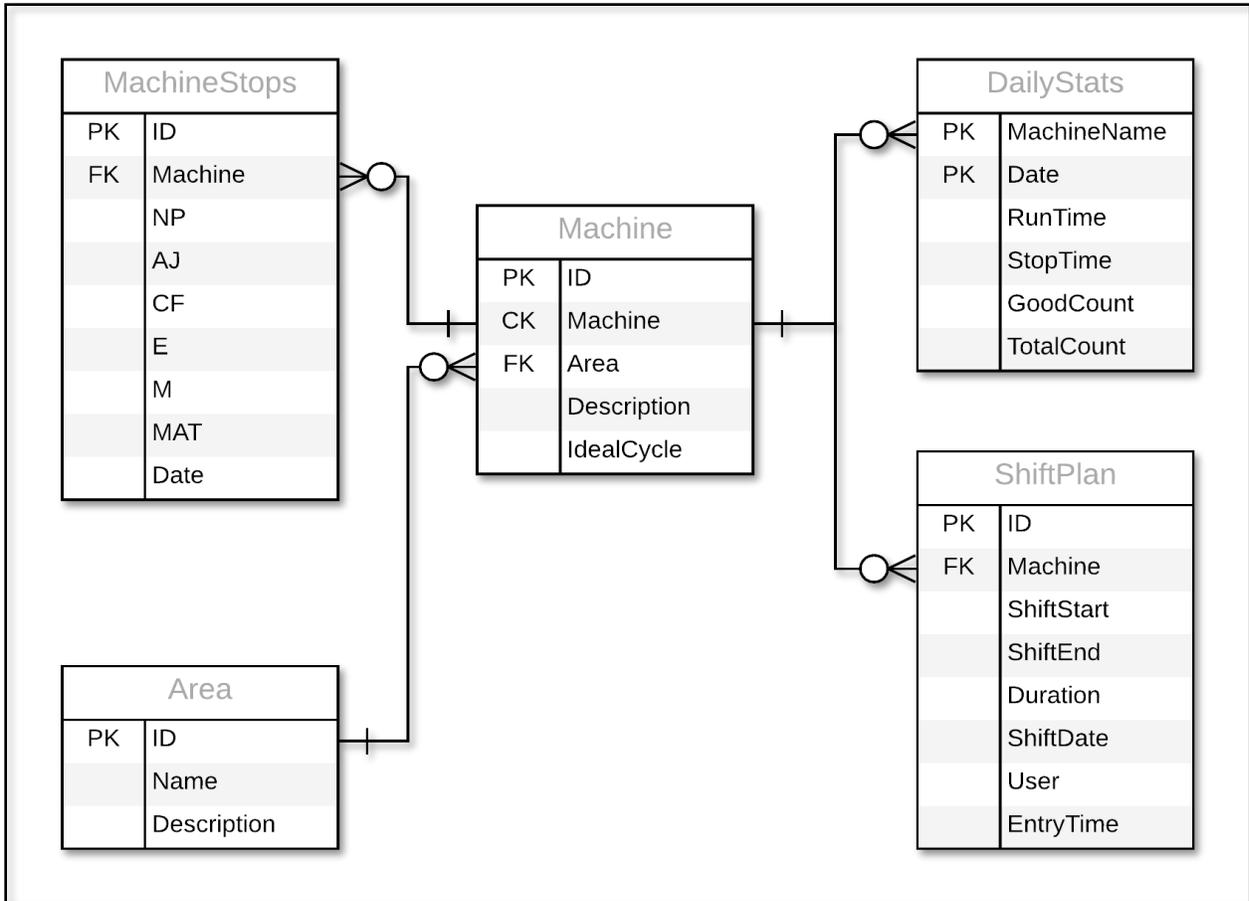


Figura 5.5 Diagrama de Relación de Entidades (ERG) para el diseño de base de datos.

[FLUIDCHART (Elaboración Propia, 2018)]

5.4 Selección de la plataforma tecnológica

Para la selección de una plataforma tecnológica en donde se pueda realizar el proyecto se analizaron dos opciones principales: Ignition de *Inductive Automation* y Wonderware InTouch. Debe aclararse que al tratarse de un proyecto piloto a lo interno de la compañía, este servirá también para evaluar la viabilidad y capacidades de las distintas herramientas de *software* disponibles hoy en día. Por lo que dependiendo de la selección que realice en este proyecto, al momento en que se desee realizar una implementación similar a esta en la empresa se incluirá siempre a competir la herramienta seleccionada contra otras posibles ofertas y/o integradores de softwares distintos.

5.4.1 Wonderware InTouch

Es un software para implementación de SCADA desarrollado en Windows que utiliza el lenguaje de programación Visual Basic Script (VB Script). Su principal ventaja consiste en su validada implementación en la Industria y sus años presentes en el mercado, al igual que su capacidad para conectarse con equipos y soluciones de distintas marcas presentes en el mercado al igual que conectividad a bases de datos.

No obstante sus desventajas son bastantes. Para empezar su estructura para cobrar por licencias es compleja y limitada, se cobra por número de variables, conexiones, número de ventanas e inclusive la cantidad de clientes en la red. De manera que se restringe bastante la flexibilidad del proyecto que se vaya a realizar y la escalabilidad de este, inclusive añadir una conexión extra a una base de datos requiere de un cobro adicional. Otra desventaja relacionada con el costo es que en comparación con otros competidores sus licencias suelen ser más costosas. Otra desventaja es los requerimientos de hardware y software que se requieren para utilizar la plataforma así como su compleja instalación. Además no viene equipado de fábrica con OPC-UA, y su programación en VBScript resulta hoy en la actualidad una desventaja debido a su complejidad y rezago respecto a lenguajes de programación más recientes y amigables tales como Python. Sin funcionalidades de IoT o para utilizar clientes móviles ni capacidad incluida para conectar algún dispositivo como un microcontrolador a través del protocolo MQTT a no ser que se migre a InTouch Edge HMI y System Platform que requieren de un coste adicional.

5.4.2 Ignition

Ignition es una plataforma desarrollada en Java con funcionalidades de SCADA, IoT y MES que permite conectar de manera sencilla datos de una enorme variedad de fuentes, diseñar una aplicación industrial y de manera inmediata lanzar por web (*web-deploy*) clientes en cualquier parte del mundo (únicamente creando un acceso remoto al Gateway en donde se encuentre el servidor web de Ignition instalado). Además su estructura de cobro por licencia consiste en un único pago por servidor que incluye un número **ilimitado** de conexiones, Tags, clientes y diseñadores (permitiendo con la misma licencia que múltiples desarrolladores trabajen de manera simultánea desde ordenadores distintos sobre el mismo proyecto, disminuyendo el tiempo de desarrollo). Otra ventaja considerable de Ignition es que ofrece la descarga gratuita del software con absolutamente todas sus funcionalidades a excepción de que al ser una versión de prueba se

debe reiniciar el periodo de prueba cada 2 horas desde el Gateway, lo que resulta principalmente conveniente para la elaboración de un proyecto piloto como en este caso.

Su principal desventaja podría ser su poco tiempo en el mercado, no obstante en la actualidad compañías como Coca-Cola, Johnson&Johnson (del área farmacéutica), Kelloggs, Walt Disney, Fiat Chrysler Automobiles entre otras mega-compañías han realizado implementaciones de proyectos en Ignition.

Por todos estos motivos se opta por seleccionar Ignition, no obstante en busca de justificar su elección aún más y demostrar su capacidad validada para solventar problemas en industrias farmacéuticas se presenta el siguiente ejemplo. A inicios de Setiembre se realizó la conferencia anual de usuarios de Ignition, en donde se presentó el caso exitoso que realizó la compañía especializada en servicios de automatización y sistemas de control Wunderlich-Malec para la compañía farmacéutica de origen suizo Bachem en su nueva planta en California Estados Unidos.

De manera similar a este proyecto la compañía realizó inicialmente un proyecto piloto haciendo uso de los simuladores de PLC de ignition y de un ambiente de simulación que les permitió realizar las pruebas de validación y aceptación antes de ser implementado en la planta. Lograron de este modo desarrollar una solución lo suficientemente flexible para las necesidades propias de la farmacéutica pero a la vez lo suficientemente confiable como para satisfacer las necesidades del estándar 21CFR parte 11 de la FDA para sistemas electrónicos y de software (Inductive Automation, 2018).

5.5 Arquitectura del sistema

El diseño del sistema propuesto consiste en una única instalación de Ignition por lo que adopta la arquitectura básica de la plataforma. Basada en un servidor con múltiples conexiones, a modo que los clientes que se integren a la red se comuniquen directamente con el servidor para obtener las lecturas o valores de los tags vinculados a un dispositivo y los registros históricos de la base de datos. Esta configuración a su vez permite que si un proyecto se actualiza o modifica, no se requiere actualizar de manera manual cada cliente que se encuentra conectado, más bien al realizar los cambios desde el servidor estos son transferidos de manera automática. Lo que otorga a la arquitectura sencillez y flexibilidad.

Como se mencionó, la arquitectura del sistema consiste de un único servidor (por lo cual se requiere una única licencia para todo el sistema), en el cual se ejecuta un servicio en segundo plano en el sistema operativo de este conocido como *Gateway* (puerto de acceso). Dependiendo de la seguridad que se utilice cualquier computador dentro de la misma red puede acceder al *Gateway* a través de la dirección IP del computador donde se encuentra instalado Ignition, facilitando enormemente la conexión (lanzamiento) de un nuevo cliente de ejecución (cliente que utilice la plataforma) o un cliente de desarrollo (que modifique o cree un proyecto).

Los dispositivos de planta que se desean conectar a la red (como un PLC o un microcontrolador) se conectan de manera directa al servidor ya sea a través de drivers OPC-UA incluidos por Ignition que facilitan enormemente la conexión con una gran cantidad de marcas como Siemens, Allen-Bradley y Omron vía Ethernet. También soporta conexiones con dispositivos que utilicen los protocolos Modbus TCP o Modbus RTU. Mientras que su versión *Edge* ofrece el *plug-in* (programa adicional) para conectar a través del protocolo MQTT y soporte de Amazon AWS (ambos protocolos de comunicación para el IoT) a una enorme variedad de dispositivos electrónicos.

Finalmente la arquitectura de este sistema también presenta la conexión a una base de datos (para el caso de este proyecto piloto MySQL 8.0) la cual puede encontrarse, dentro de la red, en un computador distinto al que contiene al servidor. Nuevamente se aclara que para la realización de este proyecto piloto tanto el servidor de Ignition como la base de datos se encuentran instaladas en el mismo computador (computador con Windows 10 de 64 bits como sistema operativo y procesador Intel CORE i7). La arquitectura utilizada en este proyecto presenta la estructura que se muestra en la Figura 5.6.

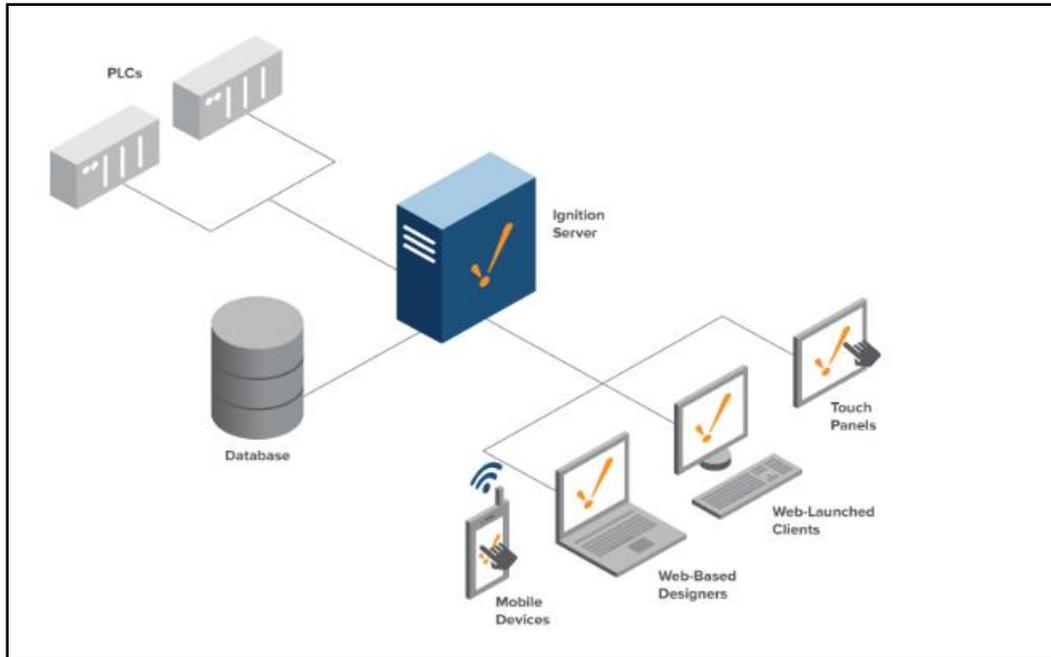


Figura 5.6 Arquitectura Estándar del sistema diseñado utilizando Ignition.

[Inductive Automation, 2018]

Otro aspecto que se debe mencionar de la arquitectura seleccionada, es que presenta la capacidad para servir como base a nuevas arquitecturas más complejas. Tales como contar con un único servidor pero con múltiples redes actuando como un puente entre estas, e inclusive restringir ciertos accesos o permisos basados no solamente en el rol del usuario sino también dependiendo de la localización de la red. O también permitir un acceso remoto desde cualquier parte del mundo haciendo uso de una implementación VPN. Estas capacidades de escalabilidad de la red se tuvieron en mente al momento de seleccionar tanto la plataforma por utilizar como la arquitectura del sistema.

Al tratarse de un proyecto piloto, que por ende se desarrolló en un computador personal y no se ha implementado en un servidor del departamento de TI de la empresa, la arquitectura utilizada cumple con la mostrada en la Figura 5.6 pero con algunas aclaraciones. En primera instancia tanto el servidor como la base de datos se encuentran instaladas en el mismo computador, además los dispositivos conectados para el desarrollo del proyecto consisten de simuladores de PLC que ofrece Ignition desde su *Gateway*, por último se verificó en la sección de resultados la capacidad de desplegar clientes web desde la misma red e inclusive dispositivos móviles utilizando

el puerto:8088 y la dirección IP asignada por el router al computador donde se encuentra instalado Ignition ejecutando el servicio.

5.6 Programación en Ignition.

Desde el diseñador de Ignition, el cual se puede ejecutar desde cualquier acceso al *Gateway* de Ignition se procedió a la creación de un nuevo proyecto denominado “Test” este es el nombre del proyecto final que contiene la solución planteada. Si no se ha realizado ninguna modificación desde el servidor web a las credenciales de acceso para el usuario administrador o no se ha creado otro usuario con permisos para realizar tareas de diseñador, se debe ingresar el nombre de usuario “admin” y la contraseña “password” para poder acceder al proyecto y comenzar con su programación.

Para facilitar la comprensión de la programación realizada en la plataforma se procederá a utilizar la siguiente estrategia: en primera instancia se definirán algunos conceptos esenciales propios de la plataforma que permitieron el desarrollo del proyecto, posteriormente se expondrán las variables utilizadas para la elaboración del proyecto con una breve descripción de su propósito, al igual que las ventanas creadas para la interacción con los clientes de ejecución y finalmente se explicará la lógica de programación utilizada.

Se evitará ahondar en el código creado así como la configuración de todos los componentes dentro de la plataforma, no obstante esta información estará disponible, para el lector que le interese, a través del archivo test_TEC.proj. Este archivo se encuentra dentro de la carpeta que contiene toda la documentación de este trabajo final de graduación, y con el podrá acceder de manera completa a la vista de diseñador del proyecto y revisar, si así lo deseara, cada línea de código y componente. Se debe aclarar que para poder abrir el archivo test_TEC.proj el interesado debe contar con la instalación de Ignition en su computador y crear un nuevo proyecto desde el diseñador en donde posteriormente podrá seleccionar File/Import y la dirección donde haya descargado el archivo acá proporcionado tal como se muestra en la Figura 5.7.

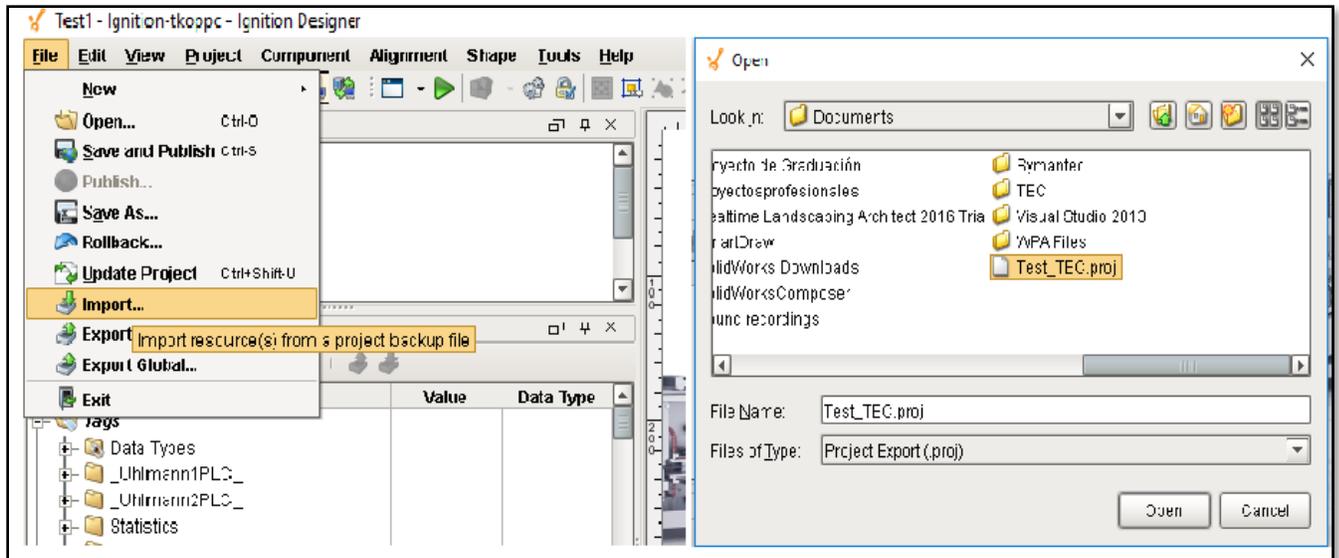


Figura 5.7 Importación del archivo que contiene el proyecto desarrollado a otro servidor de Ignition.

[IGNITION (Elaboración Propia, 2018)]

5.6.1 Conceptos de Ignition

Ignition, como todo software actual de alto nivel, utiliza una mezcla de lenguajes de programación y les asigna sus reglas para realizar el desarrollo del programa. Además ofrece una serie de componentes previamente configurados que agilizan enormemente el desarrollo de un proyecto, así como un grupo de módulos que ofrecen capacidades que no hay que programar sino más bien configurar a las necesidades del proyecto de cada desarrollador. Entre estos módulos destacan los módulos de alarma y reporte que permiten el rápido desarrollo de reportes profesionales con adquisición de datos y capacidades para utilizar comandos estadísticos que permitan obtener información deseada. Inclusive estos módulos ofrecen la capacidad de generar los reportes de manera periódica actualizando los datos con la información correspondiente y descargar los reportes en un directorio específico del computador o inclusive enviarlos de manera automática utilizando un servidor SMTP vía email a un correo en específico o inclusive a todos los usuarios pertenecientes a un rol que tengan un horario activo.

Esto último da paso a otra funcionalidad particularmente útil en Ignition que es las ventanas previamente configuradas, Inductive Automation (empresa desarrolladora de Ignition) cuenta con sus propios desarrollos que pone a disposición de todos los que utilicen el software, tal es el caso de las ventanas para manejo de usuarios desde un cliente, de este modo es posible crear

rápidamente un usuario nuevo con su información personal, asignarle un rol y hasta un horario de trabajo. De lo contrario esto se tendría que realizar desde el servidor web (desde el *Gateway*) mismo lugar donde hay que realizar toda la configuración necesaria para la creación de roles y sus permisos.

En este proyecto se han creado tres roles distintos: Administrator, Mantenimiento y Operario y a cada uno se le han asignado ciertas restricciones, por ejemplo un operario no tiene acceso a realizar cambios sobre las opciones administrativas ni visualización de estadísticas. Pero un Administrador tampoco debería poder seleccionar los motivos de paro de las máquinas. Estas restricciones se hacen por medio de la configuración de los ajustes de seguridad que ofrece Ignition para cada componente del proyecto, logrando de manera sencilla hasta que un rol no tenga acceso a dar click sobre un botón.

También se hace uso de la ventana previamente configurada de Ignition para el manejo de horarios, a través de la cual se puede crear un sinnúmero de horarios que hasta pueden automáticamente considerar días feriados agregados al sistema. Cuando se le asigna a un usuario un horario, esta información se utiliza internamente por los módulos de reportes y alarmas mencionados anteriormente para enviar alarmas y/o reportes únicamente a los usuarios que tengan horarios activos (en otras palabras cuando se supone que estén en la planta dentro de su horario laboral).

Estas breves funcionalidades de Ignition mencionadas hasta el momento requieren únicamente de configuración pero no de programación detallada por lo que se acelera el desarrollo del proyecto.

Como se mencionó, Ignition utiliza una combinación de lenguajes de programación regulados por sus propias reglas en su ambiente de programación. Por lo general en Ignition se distinguen 3 maneras de generar código (al menos en el presente proyecto se hace uso únicamente de estas).

El primero de ellos es a través de la sección de Scripts dentro del proyecto (ver Figura 5.8), acá se distinguen entre *Client Event Scripts* y *Gateway Event Scripts* (el primero de ellos son códigos que se ejecutan en el cliente más no en el servidor, mientras que el segundo son códigos que se ejecutan directamente en el servidor y por ende en todos los clientes que se conecten a este de forma simultánea).

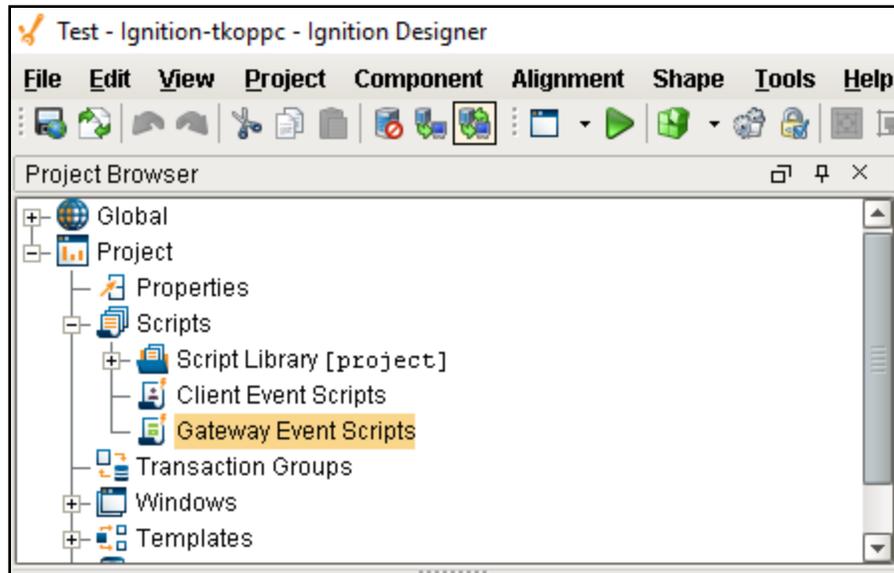


Figura 5.8 Scripts en Ignition.

[IGNITION (Elaboración Propia, 2018)]

Se distinguen 5 tipos de *Event Scripts* (estos se programan utilizando Python como lenguaje de programación, aunque se pueden utilizar ciertas funciones propias de Java):

- **Startup:** Se ejecuta cada vez que se inicie el cliente o el *Gateway*. De este modo acá se pueden programar acciones necesarias antes del inicio del sistema de forma dinámica.
- **Shutdown:** Se ejecuta cada vez que se cierre un cliente o detenga el *Gateway*. Esto permite realizar acciones necesarias como actualizar una base de datos o apagar de manera correcta los dispositivos que puedan estar conectados al sistema antes de cerrarse. Nótese que esto no protege contra pérdidas abruptas de energía que desconecten el cliente o servidor.
- **Timer:** Se ejecutan de manera periódica utilizando un ciclo de reloj interno fijo o a una tasa de retardo dada. De este modo, se ejecuta cada cierto tiempo establecido que pase. Se puede incluir más de uno.
- **Tag Change:** Se ejecuta cada vez que el valor o calidad de un *Tag* o un grupo de *Tags* cambien.
- **Message:** Se ejecutan en el cliente o *Gateway* dependiendo del seleccionado, pero que pueden ser invocados por un mensaje de otro proyecto o inclusive otro *Gateway*.

La segunda manera de generar código en Ignition consiste en la creación de *Tags* (similar al concepto de variables para otras plataformas). Aunque en la siguiente sección se expandirá

en las variables creadas y su función; en este punto se definirá de manera general el modo de interactuar con estas.

Ignition define los tipos de *Tags* que se muestran en la Tabla 5.2. Dependiendo del tipo de *Tag* se puede ejecutar una serie de código que permita interactuar con las variables y contribuir al desarrollo de la solución. En todos los tipos de *Tags* se cuenta con una sección denominada Tag Events en donde se puede escribir las funciones del programa relacionadas con la variable seleccionada. Estos códigos se ejecutan cada vez que el valor del Tag/variable cambie, un ejemplo esto se observa en la Figura 5.9.

Tabla 5.2 Tipos de *Tags* en Ignition.

Tipo	Descripción
<i>OPC Tag</i>	Variable que debe provenir de un dispositivo conectado al sistema. Requiere de un servidor OPC y la dirección al ítem que se desea.
<i>Memory Tag</i>	Variable creada propiamente en la plataforma para su uso interno, similar a una variable local en otros lenguajes de programación.
<i>Query Tag</i>	Variable interna similar a un <i>Memory Tag</i> pero que soporta consultas SQL de la Base de Datos o Bases de Datos que estén conectadas al Servidor de Ignition.
<i>Expression Tag</i>	Variable Interna similar a un <i>Memory Tag</i> pero que soporta lo que se conoce en Ignition como <i>expression language</i> , el cual permite escribir expresiones lógicas o matemáticas con valores provenientes de otros <i>Tags</i> o componentes agregados.
<i>Derived Tag</i>	Variable derivada de otra variable/Tag, que hace referencia al valor de esta última y es capaz de sobrescribir su propio valor original. Similar a un puntero en otros lenguajes de programación.
<i>System Tags</i>	Tags propias del sistema, que se encuentra bajo la carpeta <i>System</i> , y que almacena información propia de las conexiones, redes, dirección, y más de los clientes y el Gateway

[Elaboración Propia, 2018]

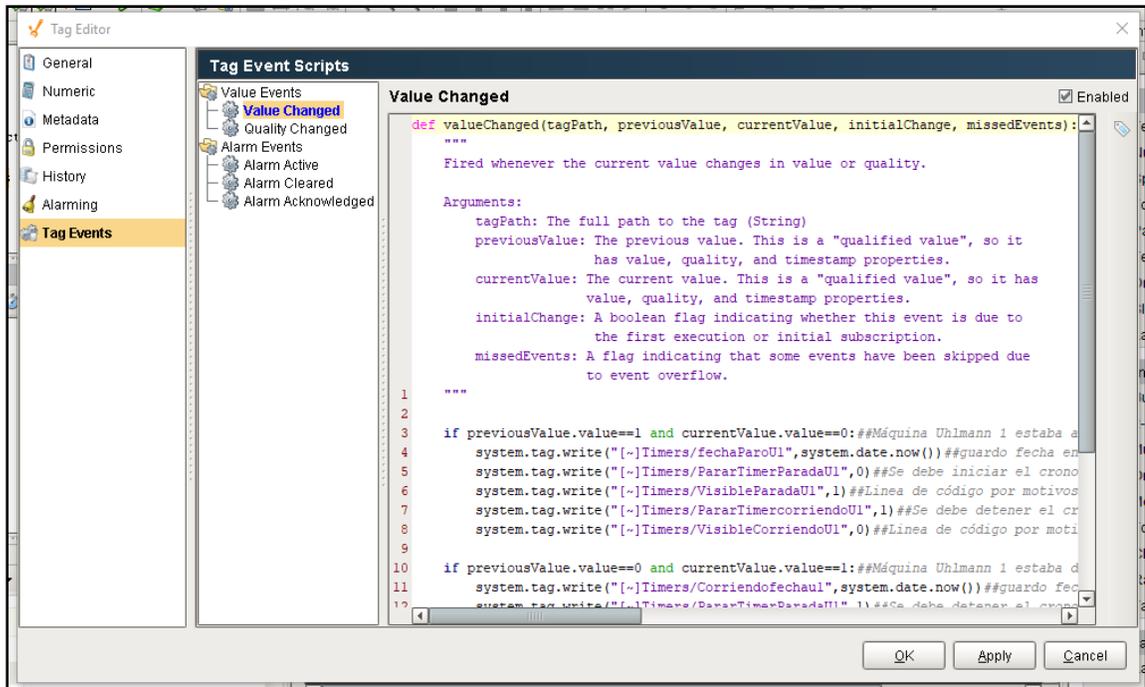


Figura 5.9 Ejemplo de código para Eventos en el Tag en Ignition.

[IGNITION (Elaboración Propia, 2018)]

Otros tipos de *Tags* como los *Expression Tags* soportan el lenguaje de expresión de Ignition, el cuál es un lenguaje de programación propio de Ignition que cuenta con una enorme cantidad de funciones que permiten realizar cálculos matemáticos, lógicos, utilizar funciones propias de JavaScript, funciones de tiempo (para el manejo de variables tipo *date* o *datetime*), funciones para leer estados de alarma, manipular datos tipo *string* entre otras más, para obtener un valor y asignarlo a la variable seleccionada. Un ejemplo de esto se ve en la Figura 5.10.

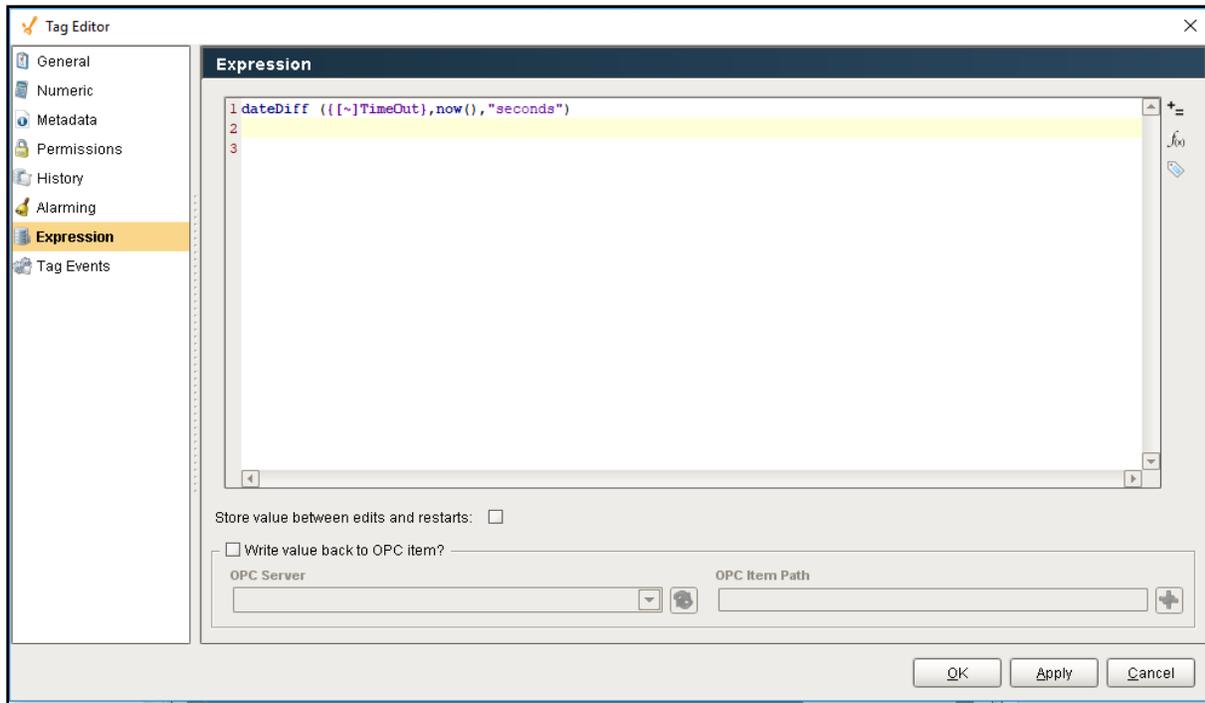


Figura 5.10 Ejemplo de código para expresiones en Tags de Ignition.

[IGNITION (Elaboración Propia, 2018)]

Las variables tipo *Query Tags* soportan códigos SQL para leer valores de una base de datos conectada y asociar algún resultado de la consulta a la variable seleccionada. Aunque en el presente proyecto se utiliza bastante la base de datos (MySQL) no hubo necesidad de utilizar variables de este tipo para escribir código SQL y acceder a la base de datos conectada.

Por último, el otro mecanismo de generar código dentro de Ignition es a partir de lo que se conoce como componentes en Ignition. Estos son elementos pre configurados por Ignition o incluso por usuarios de Ignition que deciden convertirlos públicos en los blogs de Ignition (Inductive Automation cuenta con su propia especie de red social conformada tanto por desarrolladores propios de la compañía como por integradores independientes de todo el mundo, en donde se generan temas de discusión y resuelven problemas). En el caso del presente proyecto por ejemplo se hace uso del archivo `FormComponentsPalette.proj` elaborado y compartido en el 2014 por Nick Mudge (un integrador independiente de Ignition), el cual incluye una serie de componentes que agilizan el tiempo necesario para crear entradas dadas por los usuarios desde la plataforma y que alimenten una base de datos (Mudge N,2014).

Cada componente tienen una serie de propiedades específicas, a partir del editor de estas (*Property Editor*) se puede generar el código necesario para realizar actividades dentro de la plataforma. Por ejemplo es posible cambiar el valor de una pantalla numérica según el valor de un *Tag* (a través de un *property binding* con un *Tag*), o ligando su valor a una expresión (utilizando el lenguaje de expresión de Ignition explicado anteriormente), o ligando su valor a una consulta de consulta SQL de una base de datos. Un ejemplo de esto se muestra en la Figura 5.11.

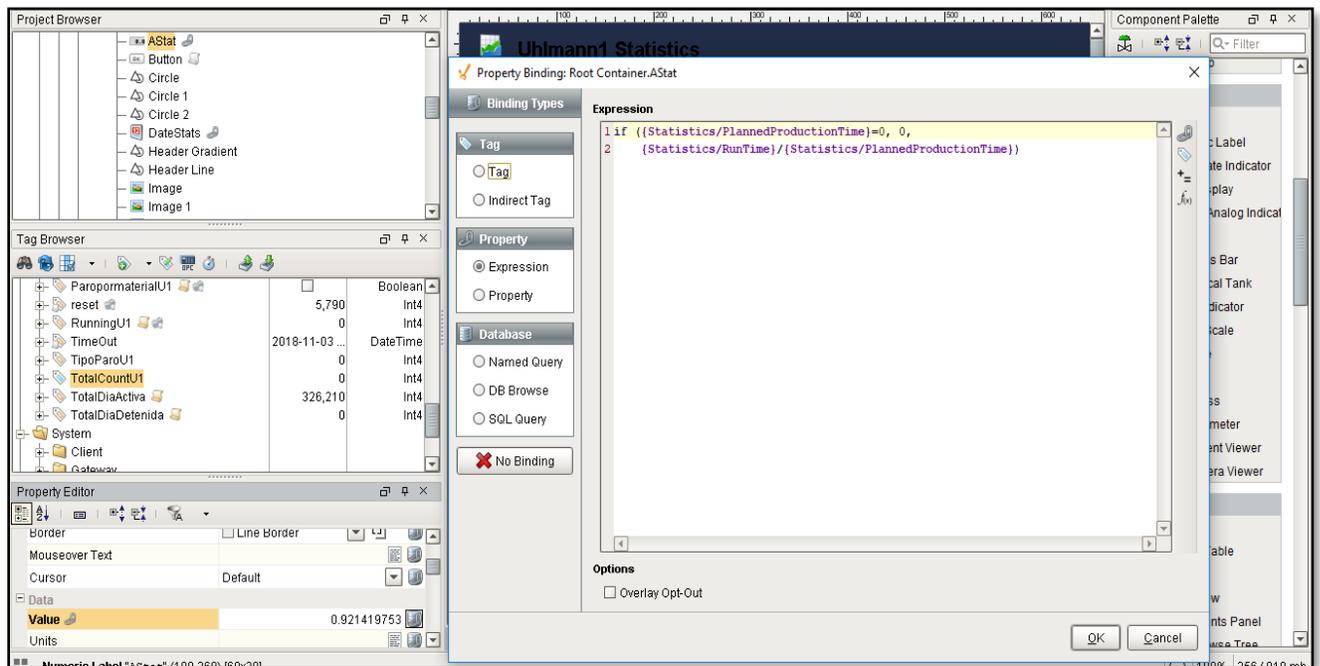


Figura 5.11 Ejemplo de ligar el valor de una propiedad de un componente en Ignition a través de una expresión.

[IGNITION (Elaboración Propia, 2018)]

Cada componente además soporta una serie de acciones que se pueden configurar haciendo click derecho sobre el componente que se desea modificar/controlar. Como se observa en la Figura 5.12, estas consisten principalmente de las siguientes opciones:

- *Customizers*: A partir de acá se pueden crear nuevas propiedades (de cualquier tipo de datos) para el componente seleccionado.
- *Scripting*: Esto permite asociar un evento o código que se ejecute en el momento en que se realice una acción sobre el componente. En el presente proyecto todos los componentes que incluyen código propio se ejecutan a partir de cuándo se presiona con un click (o en el

caso de una pantalla táctil cuando se pulse) sobre el componente. Un ejemplo de esto se muestra en la Figura 5.13.

- *Security*: Esta opción permite regular los permisos de los distintos roles dentro de la plataforma. Es aquí donde se puede evitar que un usuario de un rol dado pueda realizar un cambio en un componente que no debería, como por ejemplo presionar un botón o seleccionar un elemento de una lista.
- *Translations*: Esta opción permite crear traducciones del texto de un componente, de modo de que si se desea cambiar de idioma en la plataforma se actualicen los textos de los componentes según el idioma seleccionado.

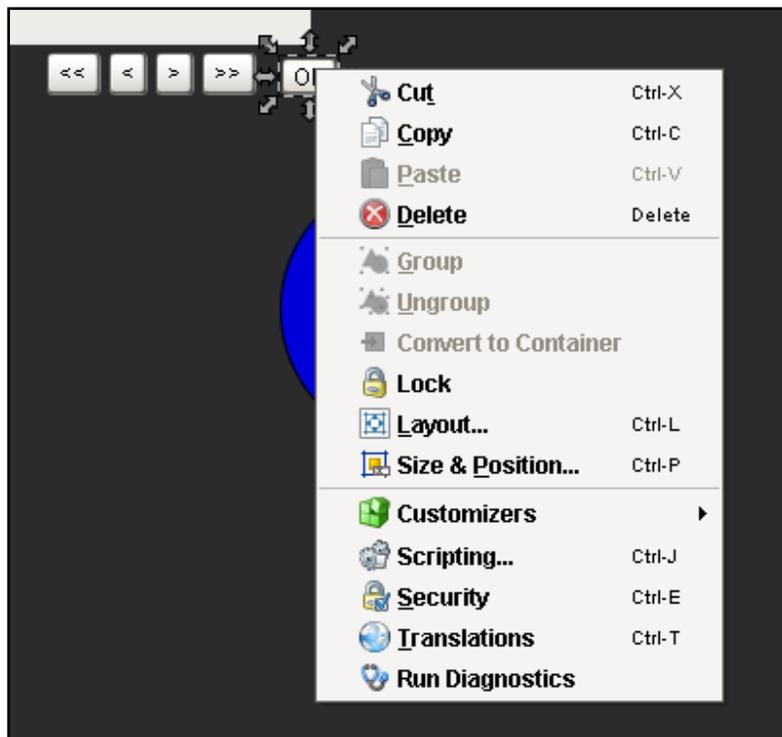


Figura 5.12 Opciones disponibles para los componentes en Ignition.

[IGNITION (Elaboración Propia, 2018)]

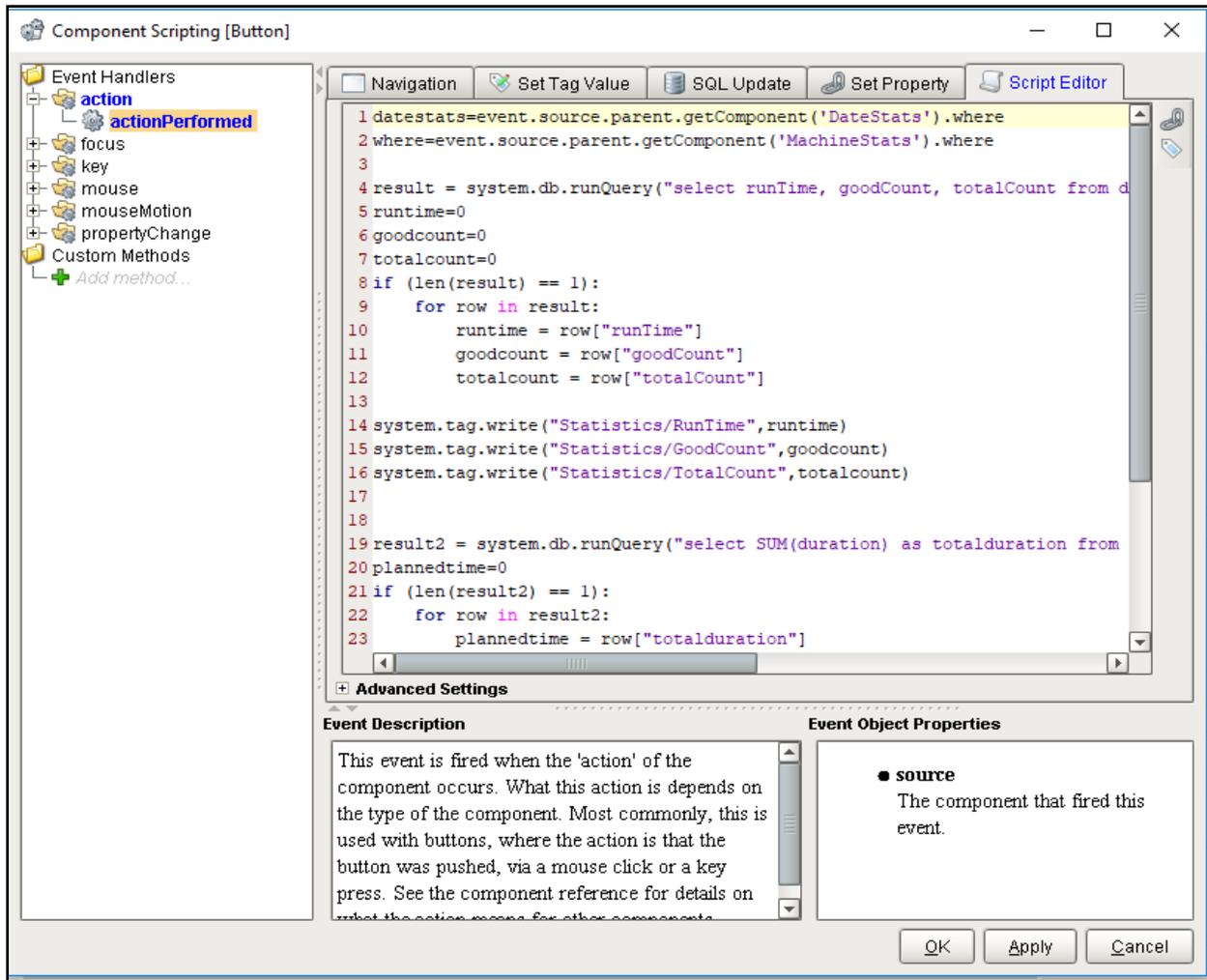


Figura 5.13 Ejemplo de programación para componente cuando se realiza una acción sobre este.

[IGNITION (Elaboración Propia, 2018)]

5.6.2 Variables Utilizadas

Para el desarrollo del programa se crearon un total de 80 Tags/variables. Estas se clasificaron en distintos folders para facilitar su distinción. Se agruparon según:

5.6.2.1 Variables de la Uhlmann 1

Acá se agrupan todas las variables que tienen que ver con la Uhlmann 1 o que se utilizaron para realizar actividades relacionadas a esta máquina. Las variables se resumen en la Tabla 5.3.

Tabla 5.3 Descripción de las variables de la Uhlmann1

Nombre de la Variable	Tipo de Tag	Proveedor	Tipo de dato	Descripción
<i>LuzRojaContinuaU1</i>	OPC Tag	Uhlmann1 PLC	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 1 para la luz roja continua, lo que indica que la máquina se encuentra detenida sin errores. Se utiliza en el programa para conocer el estado de la Uhlmann 1
<i>LuzRojaIntU1</i>	OPC Tag	Uhlmann1 PLC	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 1 para la luz roja intermitente, lo que indica que la máquina se encuentra detenida con errores. Se utiliza en el programa para conocer el estado de la Uhlmann 1
<i>LuzVerdeContinuaU1</i>	OPC Tag	PLC Simulator	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 1 para la luz verde continua, lo que indica que la máquina se encuentra funcionando. Se utiliza en el programa para conocer el estado de la Uhlmann 1
<i>LuzVerdeIntU1</i>	OPC Tag	PLC Simulator	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 1 para la luz verde intermitente, lo que indica que la máquina se encuentra detenida en espera de una máquina postconectada. Se utiliza en el programa para conocer el estado de la Uhlmann 1
<i>LuzAmarillaU1</i>	OPC Tag	PLC LuzAmarilla	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 1 para la luz amarilla continua, lo que indica que la máquina se encuentra detenida con advertencia de material. Se utiliza en el programa para conocer el estado de la Uhlmann 1
<i>TotalCountU1</i>	OPC Tag	Uhlmann1 PLC	Entero	Esta variable recibe el valor de la cantidad de blísteres producidos por la Uhlmann 1, proveniente de la máquina. Se utiliza para realizar los cálculos de indicadores de desempeño del OEE para la Uhlmann 1 y se almacena al final de cada día en la tabla <i>dailystats</i> en MySQL.
<i>GoodCountU1</i>	OPC Tag	Uhlmann1 PLC	Entero	Esta variable recibe el valor de la cantidad de blísteres en buen estado producidos por la Uhlmann 1, proveniente de la máquina. Se utiliza para realizar los cálculos de indicadores de desempeño del OEE para la Uhlmann 1 y

				se almacena al final de cada día en la tabla <i>dailystats</i> en MySQL.
<i>RunningUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para determinar si la máquina se encuentra activa o detenida. Se utiliza para la generación del Status chart que muestre el estado de la Uhlmann 1 en tiempo real.
<i>TipoParoUI</i>	Memory Tag	-	Entero	Esta variable se utiliza para determinar el tipo de paro que se ha registrado en la Uhlmann 1. Cuando vale cero significa que la máquina esta activa, si vale 1 indica un paro no planificado, 2 indica un paro por ajustes, 3 es un paro por cambio de formato, 4 se debe a un paro por espera, 5 indica paro por mantenimiento y por último un valor de 6 indica un paro por advertencia de material.
<i>Paroporespera</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por espera. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.
<i>paronoplanificadoUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro no planificado. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.
<i>ParoporaajustesUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por ajustes. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.
<i>ParoporcambioformatoUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por cambio de formato. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.
<i>ParopormantenimientoUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por trabajos de mantenimiento. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.
<i>ParopormaterialUI</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por advertencia en el control de alimentación del material. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 1.

<i>CantParosnoplanificadosU1</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros no planificados en la Uhlmann 1 que han sucedido durante el día.
<i>CantParosporajustesU1</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por ajustes en la Uhlmann 1 que han sucedido durante el día.
<i>CantParosporcambioformatoU1</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por cambios de material en la Uhlmann 1 que han sucedido durante el día.
<i>CantParosporespera</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por esperas en la Uhlmann 1 que han sucedido durante el día.
<i>CantParospormantenimientoU1</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por mantenimiento en la Uhlmann 1 que han sucedido durante el día.
<i>CantParospormaterialU1</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por advertencias de material en la Uhlmann 1 que han sucedido durante el día.
<i>cerrar</i>	Memory Tag	-	Booleano	Esta variable se utiliza para mantener cerrada o abierta la ventana de justificación de motivos de paro para la Uhlmann 1
<i>EstadoUhlmann1</i>	Memory Tag	-	Entero	Estado proveniente de la columna luminosa de la Uhlmann 1
<i>EstadoU1anterior</i>	Memory Tag	-	Entero	Es una memoria del Estado de la Uhlmann 1, se utiliza para determinar si recientemente hubo un cambio en el estado de la máquina que requiera justificación o no.

[Elaboración Propia, 2018]

5.6.2.2 Variables de cronometro de la Uhlmann1

Acá se agrupan todas las variables que tienen que ver con la medición de los tiempos en la Uhlmann 1 o que se utilizaron para realizar actividades relacionadas a este fin. Las variables se resumen en la Tabla 5.4.

Tabla 5.4 Descripción de las variables del cronómetro de la Uhlmann1.

Nombre de la Variable	Tipo	Descripción
<i>Corriendofechau1</i>	DateTime	Guarda el momento exacto en el que se produce una activación de la máquina (reanudación).
<i>fechaParoU1</i>	DateTime	Guarda el momento exacto en el que se produce un paro en la máquina.

<i>PararTimercorriendoU1</i>	Booleano	Se utiliza cuando la máquina estaba activa y se detuvo, para almacenar el valor del tiempo que llevaba activa antes de la detención y que se debe sumar al acumulado total.
<i>PararTimerParadaU1</i>	Booleano	Se utiliza cuando la máquina estaba detenida y se reanuda, para almacenar el valor del tiempo que llevaba detenida antes de la activación y que se debe sumar al acumulado total.
<i>TotalCorriendoU1</i>	Entero	Valor acumulado del tiempo que la máquina lleva activa.
<i>TotalParadaU1</i>	Entero	Valor acumulado del tiempo que la máquina lleva detenida.
<i>VisibleCorriendoU1</i>	Booleano	Motivos estéticos en la plataforma
<i>VisibleParadaU1</i>	Booleano	Motivos estéticos en la plataforma

[Elaboración Propia, 2018]

5.6.2.3 Variables de la Uhlmann 2

Las variables utilizadas para actividades de programación relacionadas con la Uhlmann 2 son idénticas a las de la Uhlmann 1 debido a que son prácticamente el mismo modelo y en este proyecto se les va a monitorear igual. La Tabla 5.5 enlista las variables de la Uhlmann 2, como puede observarse estas siguen total correspondencia respecto a las de la Uhlmann 1.

Tabla 5.5 Descripción de las variables de la Uhlmann2.

Nombre de la Variable	Tipo de Tag	Proveedor	Tipo de dato	Descripción
<i>LuzRojaContinuaU2</i>	OPC Tag	Uhlmann2P LC	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 2 para la luz roja continua, lo que indica que la máquina se encuentra detenida sin errores. Se utiliza en el programa para conocer el estado de la Uhlmann 2
<i>LuzRojaIntU2</i>	OPC Tag	Uhlmann2P LC	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 2 para la luz roja intermitente, lo que indica que la máquina se encuentra detenida con errores. Se utiliza en el programa para conocer el estado de la Uhlmann 2
<i>LuzVerdeContinuaU2</i>	OPC Tag	Uhlmann2_2	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 2 para la luz verde continua, lo que indica que la máquina se encuentra funcionando. Se utiliza en el programa para conocer el estado de la Uhlmann 2
<i>LuzVerdeIntU2</i>	OPC Tag	Uhlmann2_2	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 2 para la luz

				verde intermitente, lo que indica que la máquina se encuentra detenida en espera de una máquina postconectada. Se utiliza en el programa para conocer el estado de la Uhlmann 2
<i>LuzAmarillaU2</i>	OPC Tag	PLC LuzAmarilla	Booleano	Esta variable recibe el valor de la señal de la columna luminosa de la Uhlmann 2 para la luz amarilla continua, lo que indica que la máquina se encuentra detenida con advertencia de material. Se utiliza en el programa para conocer el estado de la Uhlmann 2
<i>TotalCountU2</i>	OPC Tag	Uhlmann2P LC	Entero	Esta variable recibe el valor de la cantidad de blísteres producidos por la Uhlmann 2, proveniente de la máquina. Se utiliza para realizar los cálculos de indicadores de desempeño del OEE para la Uhlmann 2 y se almacena al final de cada día en la tabla <i>dailystats</i> en MySQL.
<i>GoodCountU2</i>	OPC Tag	Uhlmann2P LC	Entero	Esta variable recibe el valor de la cantidad de blísteres en buen estado producidos por la Uhlmann 2, proveniente de la máquina. Se utiliza para realizar los cálculos de indicadores de desempeño del OEE para la Uhlmann 2 y se almacena al final de cada día en la tabla <i>dailystats</i> en MySQL.
<i>RunningU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para determinar si la máquina se encuentra activa o detenida. Se utiliza para la generación del Status chart que muestre el estado de la Uhlmann 2 en tiempo real.
<i>TipoParoU2</i>	Memory Tag	-	Entero	Esta variable se utiliza para determinar el tipo de paro que se ha registrado en la Uhlmann 2. Cuando vale cero significa que la máquina esta activa, si vale 1 indica un paro no planificado, 2 indica un paro por ajustes, 3 es un paro por cambio de formato, 4 se debe a un paro por espera, 5 indica paro por mantenimiento y por último un valor de 6 indica un paro por advertencia de material.
<i>ParoporesperaU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por espera. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.
<i>paronoplanificadoU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro no planificado. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.
<i>ParoporajustesU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por ajustes. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.

<i>ParoporcambioformatoU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por cambio de formato. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.
<i>ParopormantenimientoU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por trabajos de mantenimiento. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.
<i>ParopormaterialU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para indicar que se ha presentado un paro por advertencia en el control de alimentación del material. Además se usa en configuración del status chart de la ventana de gráficos de motivos de paros para la Uhlmann 2.
<i>CantParosoplanificadosU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros no planificados en la Uhlmann 2 que han sucedido durante el día.
<i>CantParosoparaajustesU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por ajustes en la Uhlmann 2 que han sucedido durante el día.
<i>CantParosoporcambioformatoU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por cambios de material en la Uhlmann 2 que han sucedido durante el día.
<i>CantParosoporesperaU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por esperas en la Uhlmann 2 que han sucedido durante el día.
<i>CantParosopormantenimientoU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por mantenimiento en la Uhlmann 2 que han sucedido durante el día.
<i>CantParosopormaterialU2</i>	Memory Tag	-	Entero	Esta variable contabiliza la cantidad de paros por advertencias de material en la Uhlmann 2 que han sucedido durante el día.
<i>cerrarU2</i>	Memory Tag	-	Booleano	Esta variable se utiliza para mantener cerrada o abierta la ventana de justificación de motivos de paro para la Uhlmann2
<i>EstadoUhlmann2</i>	Memory Tag	-	Entero	Estado proveniente de la columna luminosa de la Uhlmann2
<i>EstadoU2anterior</i>	Memory Tag	-	Entero	Es una memoria del Estado de la Uhlmann 2, se utiliza para determinar si recientemente hubo un cambio en el estado de la máquina que requiera justificación o no.

[Elaboración Propia, 2018]

5.6.2.4 Variables de cronometro de la Uhlmann 2

Al igual que en el caso para variables de la Uhlmann 2, las variables de cronómetro de la Uhlmann2 siguen total correspondencia con sus homólogas de la Uhlmann 1. En la Tabla 5.6 muestra están variables.

Tabla 5.6 Descripción de las variables del cronómetro de la Uhlmann1.

Nombre de la Variable	Tipo de dato	Descripción
Corriendofechau2	DateTime	Guarda el momento exacto en el que se produce una activación de la máquina (reanudación).
fechaParoU2	DateTime	Guarda el momento exacto en el que se produce un paro en la máquina.
PararTimercorriendoU2	Booleano	Se utiliza cuando la máquina estaba activa y se detuvo, para almacenar el valor del tiempo que llevaba activa antes de la detención y que se debe sumar al acumulado total.
PararTimerParadaU2	Booleano	Se utiliza cuando la máquina estaba detenida y se reanuda, para almacenar el valor del tiempo que llevaba detenida antes de la activación y que se debe sumar al acumulado total.
TotalCorriendoU2	Entero	Valor acumulado del tiempo que la máquina lleva activa.
TotalParadaU2	Entero	Valor acumulado del tiempo que la máquina lleva detenida.
VisibleCorriendoU2	Booleano	Motivos estéticos en la plataforma
VisibleParadaU2	Booleano	Motivos estéticos en la plataforma

[Elaboración Propia, 2018]

5.6.2.5 Variables relacionadas a estadísticas

Las variables relacionadas a estadísticas son todas las necesarias para poder calcular el OEE en las máquinas del proyecto piloto. En resumen estas variables son las que se muestran en la Tabla 5.7.

Tabla 5.7 Descripción de las variables utilizadas en el software para el desarrollo de indicadores de desempeño.

Nombre de la variable	Tipo de dato	Descripción
<i>GoodCount</i>	Entero	En ella se almacenará (al momento de efectuar los cálculos de indicadores) el valor de la cantidad de blísteres en buen estado producidos, proveniente de la base de datos.
<i>IdealCycle</i>	Flotante	En ella se almacena el valor del ciclo ideal según indicaciones del fabricante, proveniente de la base de datos.
<i>PlannedProductionTime</i>	Entero	En esta variable se almacena la suma de todos los turnos relacionados a una máquina durante un día.
<i>RunTime</i>	Entero	En esta variable se almacena la totalidad del tiempo activo de la máquina durante el día.
<i>TotalCount</i>	Entero	En esta variable se procede a almacenar la totalidad de blísteres sin importar que hayan sido rechazados o no por la máquina.

[Elaboración Propia, 2018]

5.6.3 Ventanas Creadas

Para el diseño de las ventanas se tuvo en consideración la manera en la que se desea que los usuarios tengan acceso a la información e interactúen con la plataforma. Si bien es cierto que la mayoría de los detalles por considerar para la creación de las ventanas son meramente estéticos, un diseño acertado de estas puede contribuir a la eficiencia general del sistema. Por el otro lado, un mal diseño de las ventanas perjudica directamente la manera en la que el usuario utiliza la herramienta y puede ser responsable de crear confusiones y promover un mal uso del proyecto.

Por ello se destina una sección de este proyecto para visualizar el proceso detrás de la creación de las pantallas que constituyen la solución propuesta. El primer criterio que se utilizó fue el manejo del espacio dentro de cada pantalla, con esto se pretende que una pantalla cuente con toda la información necesaria que debe desplegar pero que no se encuentre sobrecargada y dificulte comprender o distinguir lo que se busca de cada ventana por el usuario. Es aquí donde se puede optar por la creación de una nueva ventana principal o bien una ventana emergente (también conocidas como ventanas de diálogo o *popup windows*).

Esta opción se utiliza en dos ocasiones para cada máquina (4 en total). En primer lugar, con la intención de tener una única pantalla para el monitoreo de cada máquina en el sistema, que muestre las opciones de monitoreo esenciales pero que no resulte sobrecargada o comprometa su

estética se opta por agregar un botón de más información en la ventana principal de cada máquina. A partir de este botón, cuando el usuario lo presione, se despliega una ventana emergente con más información acerca del estado de la máquina que se esté monitoreando. Otras pantallas emergentes que se decidieron agregar fueron las ventanas de justificación de paro para cada Uhlmann. Estas deberán desplegarse únicamente cuando la máquina se encuentre detenida con la intención de averiguar el motivo detrás del paro.

Con este criterio debe tenerse cuidado, ya que excederse en la creación de muchas ventanas también puede resultar perjudicial (y eso que Ignition no cobra por la cantidad de ventanas a creadas a diferencia de otros *softwares*). Se debe encontrar un balance entre no sobresaturar de información una ventana pero tampoco crear demasiadas ventanas que al final compliquen su uso o confundan al usuario sobre dónde puede encontrar la información que desea. Por ello por ejemplo en lugar de crear múltiples ventanas para desplegar las estadísticas de cada máquina (una ventana por cada máquina que se incluya dentro del sistema), se opta más bien por crear una única ventana en donde el usuario conozca o intuya que puede buscar información estadística de todas las máquinas en el sistema seleccionando de una lista de opciones.

Esto da paso al siguiente criterio utilizado, las ventanas y el sistema en general deben ser lo más intuitivos posibles. El usuario debe ser capaz de familiarizarse con la herramienta relativamente rápido y la información se deberá desplegar de manera que este sepa dónde buscar dentro de la plataforma. Para esto se ha utilizado una única ventana de navegación que le permita al usuario desplazarse dentro de la plataforma. Además desde esta el operario puede seleccionar el idioma del sistema, cerrar la sesión, cambiar de usuario y bloquear la pantalla.

Siguiendo con la idea de que el sistema debe ser intuitivo, se define el último criterio considerado en este proyecto para la creación de las ventanas: el sistema debe ser estructurado. Entiéndase por estructurado un sistema que sigue una estructura temática definida. En el caso de este proyecto se propone una estructura compuesta por tres secciones, las ventanas de monitoreo, las ventanas para uso administrativo y las ventanas para análisis estadístico (en el caso de este proyecto piloto se cuenta únicamente con una ventana en esta sección). Finalmente las ventanas del sistema se resumen en la Tabla 5.8.

Tabla 5.8 Descripción de las ventanas del sistema diseñado.

Nombre de la ventana	Estructura	Descripción
<i>Navegación</i>	Navegación	En esta ventana se permite al usuario desplazarse por la plataforma desarrollada. Además permite el cambio del idioma del sistema, cambiar de usuario, cerrar la sesión y/o bloquear la pantalla
<i>Monitoreo Uhlmann1</i>	Monitoreo	En esta ventana se cuenta con la información principal correspondiente al monitoreo en tiempo real de la Uhlmann 1
<i>Monitoreo Uhlmann2</i>	Monitoreo	En esta ventana se cuenta con la información principal correspondiente al monitoreo en tiempo real de la Uhlmann 2
<i>Gráficos Motivos de Paro Uhlmann1</i>	Monitoreo/Popup	En esta ventana se cuenta con información detallada correspondiente al estado en tiempo real de la Uhlmann 1 y sus tipos de paro durante el día
<i>Gráficos Motivos de Paro Uhlmann2</i>	Monitoreo/Popup	En esta ventana se cuenta con información detallada correspondiente al estado en tiempo real de la Uhlmann 2 y sus tipos de paro durante el día
<i>Justificación Motivos de Paro Uhlmann1</i>	Monitoreo/Popup	En esta ventana se averigua el motivo que ocasionó un paro en la Uhlmann1
<i>Justificación Motivos de Paro Uhlmann2</i>	Monitoreo/Popup	En esta ventana se averigua el motivo que ocasionó un paro en la Uhlmann2
<i>Manejo de Usuarios</i>	Administración	En esta ventana se permite a un usuario de un rol con acceso (permiso), crear un nuevo usuario, actualizar la información de un usuario existente o eliminar un usuario. Al igual que asociar un usuario a un rol previamente definido y a un horario también previamente definido
<i>Manejo de Horarios</i>	Administración	En esta ventana se permite a un usuario de un rol con acceso (permiso), crear un nuevo horario de trabajo.
<i>Añadir Máquinas</i>	Administración	En esta ventana se permite a un usuario de un rol con acceso (permiso), añadir o eliminar una máquina del sistema.
<i>Tiempo de Producción Planificado</i>	Administración	En esta ventana se permite a un usuario de un rol con acceso (permiso), añadir o eliminar turnos de trabajo planificados para cualquier máquina que haya sido añadida al sistema.
<i>Estadísticas Básicas</i>	Estadísticas	En esta ventana se permite a un usuario de un rol con acceso (permiso) poder visualizar las estadísticas de rendimiento (OEE) de cualquier día para cualquier máquina integrada al sistema

[Elaboración Propia, 2018]

5.6.4 Lógica de programación

En el proyecto se realizan una gran cantidad de actividades para garantizar la eficiencia y conseguir resultados estéticos y amigables con el usuario, en esta sección se explicará la lógica detrás de la programación de las principales de estas actividades. Como ya se hizo mención anteriormente, el lector que esté interesado en revisar la totalidad de los códigos podrá hacerlo a través del archivo test_TEC.proj.

5.6.4.1 Monitoreo tiempo real Uhlmann 1 y Uhlmann 2

Uno de los principales procesos que se lleva a cabo en el programa tiene que ver con el monitoreo en tiempo real de los estados de las Uhlmann 1 y 2, así como la posibilidad registrar justificaciones a los motivos de paros que revelen información desconocida a través de las ventanas de justificación. En general el proceso de monitoreo en la Uhlmann 1 consiste de la lectura de las señales provenientes de la columna luminosa y en caso de que se determine que la máquina se encuentra detenida se procede a preguntar al operario el motivo del paro (a no ser que sea un paro por espera o un paro por advertencia de material, en los cuales no hace falta justificación puesto que se conoce directamente de las señales).

A continuación se detalla este proceso para la Uhlmann 1 (el proceso es exactamente el mismo para la Uhlmann 2 con la diferencia de que se utilizan las variables correspondientes a esta última, por lo que no se detallará para esta), además es posible observar la lógica de programación utilizada en el diagrama de flujo de la Figura 5.15.

Este proceso se lleva a cabo en un script dentro de un *timer* por lo que el proceso se repite de manera cíclica. El proceso inicia leyendo las variables de memoria que almacenan cada tipo de paro (NP, AJ, CF, E, M, MAT) y las señales provenientes de la columna luminosa de la Uhlmann 1. Dado que solo puede estar una señal del semáforo en alto a la vez se procede a escribir sobre una variable local denominada “valor” y sobre la variable que almacena el estado actual de la Uhlmann (EstadoUhlmann1) un valor de 1 si la única señal activa es la Luz Roja Continua, un valor de 2 si la única señal activa es la Luz Roja Intermitente, un valor de 3 si la única señal activa es la Luz Verde Continua, un valor de 4 si la única señal activa es la Luz Verde Intermitente y por último un valor de 5 si la única señal activa es la Luz Amarilla Continua.

Además se lee el valor del Estado Anterior que tuvo la máquina (si es la primera vez que se inicia el sistema, este valor se inicializa en 0) y se guarda en una variable local. Se guarda siempre el estado anterior con la intención de que por ejemplo, si el estado actual de la máquina indica que se encuentra detenida debido a una luz roja continua, y anteriormente también se encontraba detenida pero con una luz roja intermitente, se debe igualmente consultar el motivo por el cual la máquina se encuentra detenida. Es decir la máquina sigue estando detenida pero hubo un cambio en su estado por lo que se debe indagar nuevamente al operario acerca de este motivo. O por ejemplo la máquina estaba activa y se detuvo por lo que debe indagarse al operario por el motivo de este paro.

Debido a ello también se devuelven los estados de las variables ParopormantenimientoU1, ParoporajustesU1, ParoporcambioformatoU1 y ParonoplanificadoU1 a cero hasta que suceda un paro y se obtenga una justificación del operario. Se debe aclarar que los ParosporesperaU1 y ParopormaterialU1 se devuelven a cero siempre que la máquina se reanude o que suceda un cambio a alto en el valor de uno de los Tags que justifican el motivo de paro, esto se observa en la sección de código de la Figura 5.14. Posterior a esta consulta se debe almacenar el valor del estado actual en la variable del estado anterior para estar preparado para la próxima ejecución del código y las nuevas lecturas.

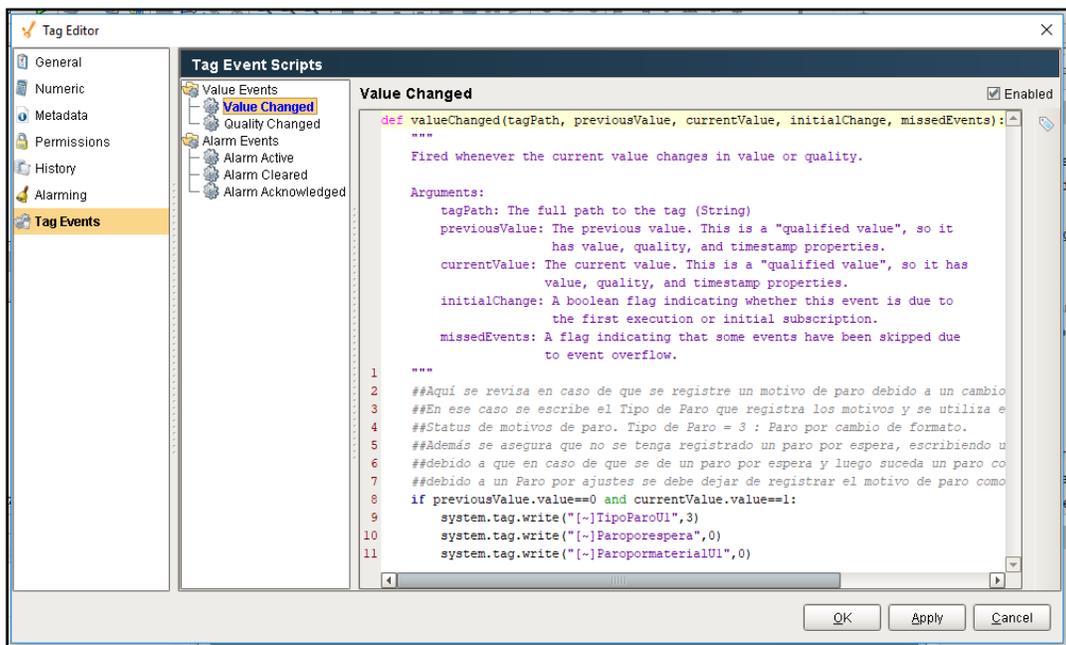


Figura 5.14 Regreso a cero para los paros por espera y paros por material cuando sucede otro tipo de paro.

Seguidamente si el “valor” (se hace referencia a la variable local) de la máquina es igual a 1,2,4 o 5 se determina que la máquina se encuentra detenida y se escribe la variable RunningU1 en 0. Una vez que se determina que la máquina se encuentra detenida se realiza una medida de presión sobre el operario en donde si no se ha reportado ningún motivo de paro (NP==0, AJ==0, M==0, MAT==0, E==0, CF==0) se empieza a contabilizar el paro como un paro no planificado (ParonoplanificadoU1=1) por otro lado si el “valor”(se hace referencia a la variable local) de la máquina es igual a 3, se determina que la máquina se encuentra activa y se escribe la variable RunningU1 en 1.

Finalmente si la máquina se encuentra detenida y no se ha cerrado la ventana de justificación de motivos de paro para la Uhlmann 1 y además el “valor” (se hace referencia a la variable local) de la máquina no es igual a 4 o 5, en otras palabras si la máquina no se encuentra detenida por espera o por advertencia de material respectivamente, se despliega la pantalla de justificación de motivos para que el operario justifique el paro.

No se requiere la ventana de justificación de motivos de paro si la máquina se encuentra detenida por espera o por material debido a que las señales de la columna luminosa ya revelan el motivo, o sea su motivo de paro ya se conoce porque es único, en cambio un paro por luz roja ya sea continua o intermitente se puede deber ya sea a mantenimiento, a un paro no planificado, a un paro por ajustes o debido a un cambio de formato.

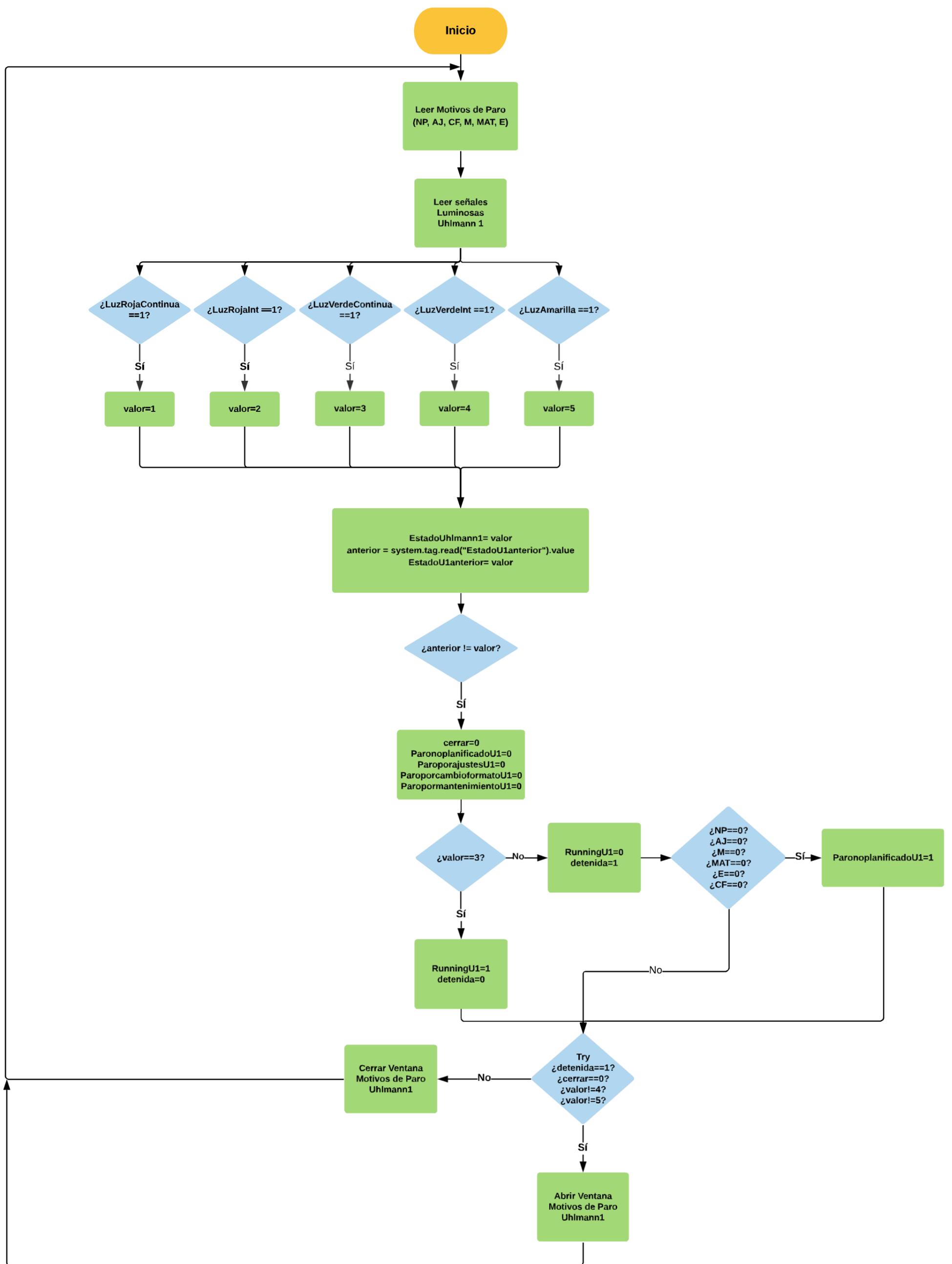


Figura 5.15 Diagrama de Flujo para la lógica de programación utilizada para el monitoreo de la Uhlmann 1.

Una vez se despliegue la pantalla de justificación de motivos de paro el operario podrá seleccionar entre 4 motivos: paro no planificado, paro por mantenimiento, paro por ajustes y paro por cambio de formatos. En el momento que el operario selecciona un motivo se ejecuta un código asociado a la opción que haya seleccionado. Básicamente lo que se realiza es leer la cantidad de paros por el motivo seleccionado registrados para la Uhlmann_1 durante el día y sumarle una unidad (aumenta la cantidad de paros por el motivo seleccionado). De igual modo se pone en alto el tipo de paro seleccionado y se escribe en alto la variable que es utilizada para cerrar automáticamente la ventana de justificación de paros una vez se seleccione una opción (cerrar=1). La Figura 5.16 muestra la sección de código que realiza este breve proceso para cuando se da la selección de Paro por Ajuste como el motivo del paro.

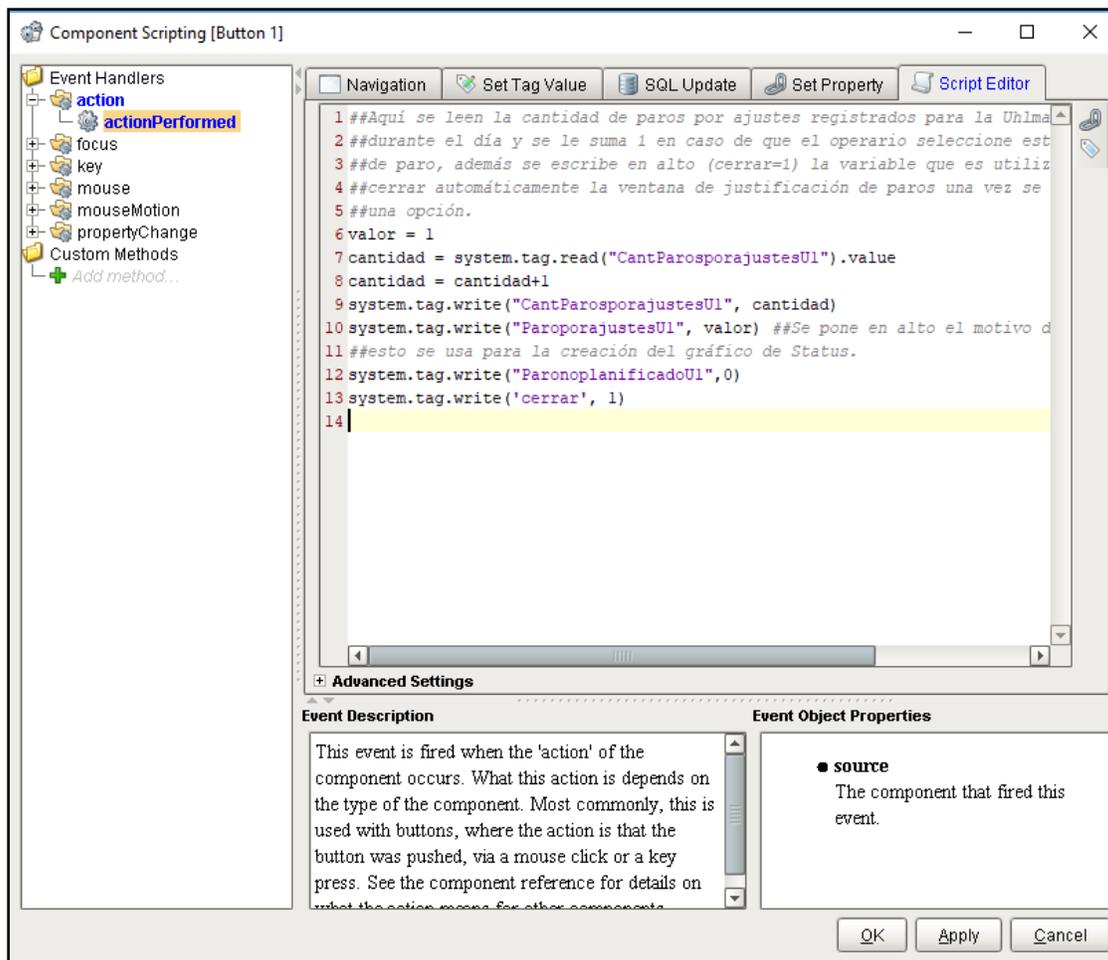


Figura 5.16 Código ejecutado cuando se realiza una justificación de paro.

[IGNITION (Elaboración Propia, 2018)]

Finalmente para visualizar lo programado de una manera gráfica y amigable desde la plataforma y teniendo en cuenta el concepto de diseño original que se pretendía lograr (recordar Figura 5.3) se optó por utilizar el componente *Status Chart* de Ignition. Para no sobrecargar la pantalla principal se decidió mostrar únicamente el estado de las Uhlmann de manera binaria en el gráfico de esta ventana, o sea máquina activa (en verde) máquina detenida (en rojo). Para ello se hizo uso de los históricos almacenados para la variable RunningU1 para la Uhlmann 1 y RunningU2 para la Uhlmann2. Mientras que en la ventana de gráficas de motivos de paro se observa a detalle y en tiempo real todas las causas de los paros de manera muy similar a lo que se tenía en mente conseguir. Para ello se hizo uso de los históricos almacenado para las 6 variables de paros creadas para cada Uhlmann y además se incluyó la variable Running para ofrecer un contexto a la información que se visualice.

5.6.4.2 Medición de los tiempos detenida y activa para la Uhlmann 1 y Uhlmann 2

Hasta el momento se ha logrado el monitoreo en tiempo real de las máquinas y el almacenamiento de la cantidad de paros para cada máquina. No obstante aún no se tiene el registro de los tiempos en que la máquina se encuentra activa y el registro de los tiempos en que se encuentre detenida. Debido a que Ignition no ofrece un componente de cronómetro que se pueda activar, reanudar y detener según el valor de una variable, se tuvo que implementar una lógica un poco extensa para lograr el resultado deseado.

El proceso realizado emplea la lógica que se muestra en el diagrama de flujo de la Figura 5.18 para la Uhlmann 1 (es exactamente igual para la Uhlmann 2, únicamente se utilizan sus variables correspondientes). A través de un *Tag Event script* para la variable RunningU1, cada vez que la máquina Uhlmann 1 estaba activa y se detuvo, se procede a guardar la fecha exacta en que se registra este paro en la variable fechaParoU1, además se debe inicializar el cronómetro que contabiliza el tiempo actual que la máquina lleva detenida, para ello se escribe la variable PararTimerParadaU1 en 0 (esta variable se utiliza de modo similar a una bandera en otros lenguajes de programación). Al igual, se debe detener el cronómetro que contabiliza el tiempo que la máquina llevaba funcionando antes de que sucediese el paro y sumar este valor al total que registra el tiempo acumulado que la máquina lleva activa, para ello se escribe en alto (1) la variable PararTimercorriendoU1 (esta variable se utiliza de modo similar a una bandera en otros lenguajes de programación).

Se realiza algo similar cada vez que la variable RunningU1 indique que la máquina se encontraba detenida y se reanudó. En este caso se procede a guardar la fecha exacta en que se registra esta activación en la variable Corriendofechau1, además se debe inicializar el cronómetro que contabiliza el tiempo actual que la máquina lleva activa, para ello se escribe la variable PararTimercorriendoU1 en 0. Al igual, se debe detener el cronómetro que contabiliza el tiempo que la máquina llevaba detenida antes de que sucediese la reanudación y sumar este valor al total que registra el tiempo acumulado de paros, para ello se escribe en alto (1) la variable PararTimerParadaU1.

De manera paralela se utiliza un componente de visualización numérico de datos, en este se asocia su propiedad *Value* con una expresión donde se calcula la diferencia en segundos entre el tiempo actual y la fecha de paro registrada, tal y como se muestra en la Figura 5.17. De este modo se construye un cronómetro artificial que iniciará en cero desde que se registre un paro y seguirá creciendo hasta que se registre otro paro y la fecha actual del sistema sea igual a la del registro del paro y por lo tanto la diferencia sea cero.

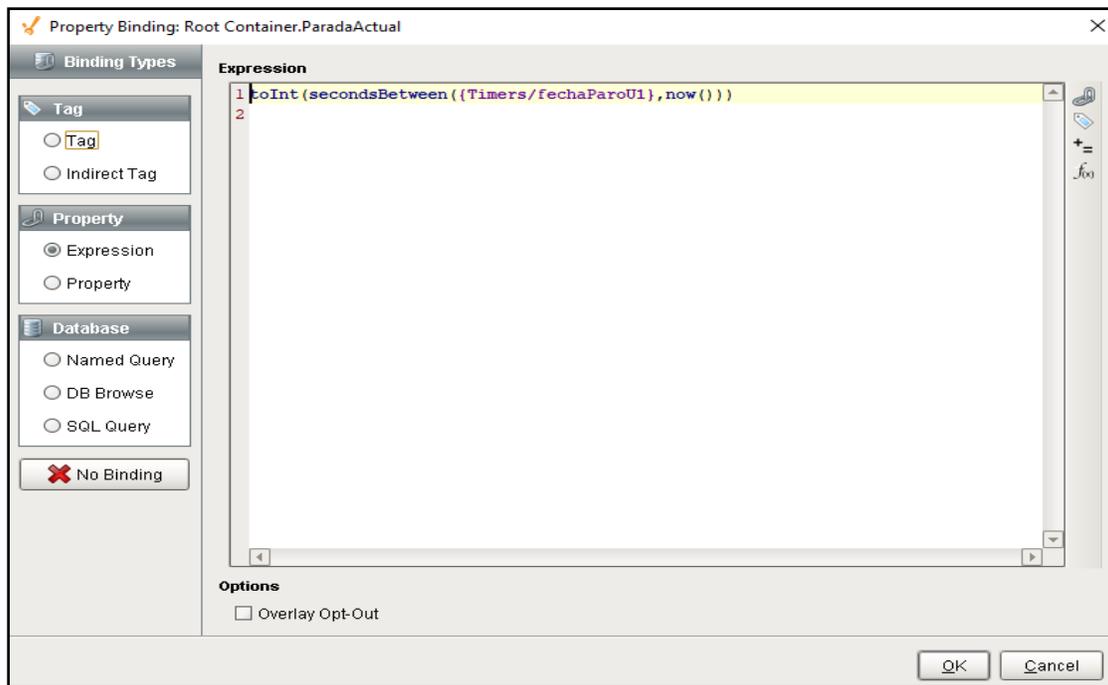


Figura 5.17 Cálculo de la diferencia en segundos ente el tiempo actual y el tiempo en que sucede el paro.

[IGNITION (Elaboración Propia, 2018)]

Lo mismo se realiza con otro componente de visualización numérica de datos, para la creación de un cronómetro artificial para registrar el tiempo actual que la máquina lleva activa (únicamente que en este caso se calcula la diferencia en segundos entre el tiempo actual y la fecha en que se registra la activación).

Finalmente en un script dentro de un *timer* (de modo que el programa se ejecute de manera cíclica) se leen las variables `PararTimerParadaU1` y `PararTimercorriendoU1`, a modo que cuando se de una reanudación (máquina pasó de estar detenida a activa y por lo tanto `PararTimerParadaU1` es igual a 1) se lea el valor que haya en ese momento en el componente de visualización que llevaba el registro del tiempo detenida y lo sume al total que registra el acumulado de tiempo detenida (`TotalParadaU1`). Finalmente se debe regresar la bandera/variable `PararTimerParadaU1` a cero, debido a que el caso ya fue atendido y ya se registró la suma (en caso contrario el sistema seguirá sumando el valor del componente al valor total acumulado).

De igual manera se realiza para cuando suceda un paro en la máquina (máquina paso de estar activa a detenerse y por lo tanto `PararTimercorriendoU1` es igual a 1). En este caso se lee el valor que hay en ese momento en el componente de visualización que lleva el registro del tiempo actual activa y lo sume al total que registra el acumulado de tiempo en que la máquina ha estado activa (`TotalCorriendoU1`). Por último se apaga la bandera (se regresa la variable a cero) `PararTimercorriendoU1`.

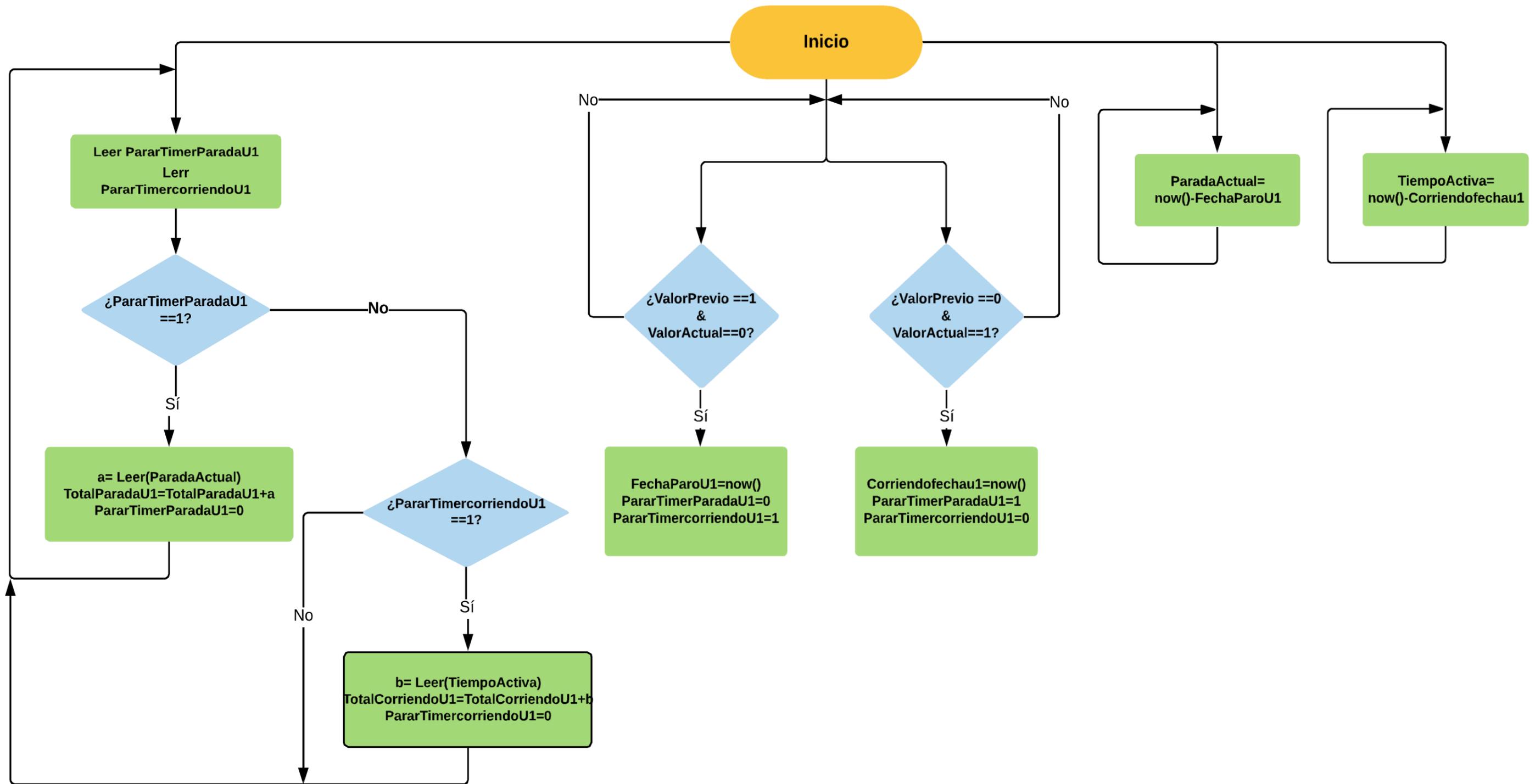


Figura 5.18 Diagrama de flujo de la lógica utilizada para la medición de los tiempos totales activa y detenida de la máquina.

[FLUIDCHART (Elaboración Propia, 2018)]

5.6.4.3 *Reset diario de variables y generación diaria de estadísticas*

Para poder realizar indicadores de desempeño de las máquinas se debe definir un espacio de tiempo sobre el cual se van a definir los cálculos. Con esto lo que se pretende decir es que para poder realizar el cálculo del OEE por ejemplo, se requiere conocer en qué espacio de tiempo se va a realizar el estudio, un OEE diario o semanal por ejemplo. Por motivos de alcance de este proyecto y contemplando su posible escalabilidad se optó por hacer un análisis diario de todas las estadísticas recopiladas por el sistema actualmente.

Esto se hace, como ya se mencionó, pensando en la escalabilidad del sistema, ya que idealmente se deberían generar automáticamente reportes diarios, semanales y mensuales de los equipos de producción, y el usuario debería poder seleccionar desde la plataforma desarrollada que estadísticas desea visualizar. Sin embargo por limitaciones de alcance de este proyecto, esto no se realizará en este piloto, pero dejando abierta la posibilidad de una futura implementación se opta por realizar un análisis estadístico diario. Ya que esta es la mínima medida, y por lo tanto a partir de los históricos almacenados en la base de datos (MySQL) para estas estadísticas se puede después construir cualquier otro reporte de manera sencilla. A diferencia de que si se seleccionase un espacio de tiempo superior, como un reporte semanal, no sería posible reconstruir un reporte diario y se estaría perdiendo información.

Con esto aclarado, se tiene entonces que se debe lograr de algún modo *resetear* las variables del sistema todos los días a la misma hora (cada 24h) y al mismo tiempo almacenar la información pertinente en la base de datos. Para ello se propone la siguiente solución que también se presenta de manera gráfica en el diagrama de la Figura 5.20.

Se utiliza la variable TimeOut, la cual es un *Expression Tag* en donde se escribe la hora exacta en que se quiere programar la acción de *reset*, el espacio para el día se asigna de manera automática según la fecha actual del *Gateway*. Posteriormente en otra variable denominada *reset*, que también es de tipo *Expression Tag*, se programa una expresión en donde se resta la diferencia en segundos entre el tiempo actual y el tiempo programado para realizar el *reset*. De igual manera en esta misma variable se utiliza un *Tag Event Script*, en donde cada vez que cambie su valor (lo cual sucede cada segundo dado que la resta programada cambia constantemente conforme avanza el tiempo) se realiza lo siguiente.

Dos segundos antes de que sea momento de realizar el reset se realiza un `system.db.runPrepUpdate` (se ingresan datos a través de una consulta o *query*) en donde se insertan los datos correspondientes a las cantidades de paros del día según cada tipo, la fecha actual y el nombre de la máquina (para la cual se contabilizan estas estadísticas). Esto se realiza tanto para la Uhlmann1 como para la Uhlmann 2.

En el momento exacto en que el valor sea cero (o sea, en el momento en que se programó el *reset*) se devuelven a cero todas las estadísticas correspondientes a las cantidades de paro según el tipo y para ambas máquinas y se pone en alto una variable denominada “comodin” que más adelante se explicará su función.

Cuando el valor cambie a 1 (1 segundo después del *reset* programado) se leen los valores totales acumulados para el tiempo activa de máquina y tiempo detenida de máquina y se almacena este valor en las variables `TotalDiaActiva` y `TotalDiaDetenida` para cada máquina. Seguidamente se devuelven a cero las variables que acumulaban los totales de tiempos (`TotalCorriendoU1`, `TotalParadaU1`, `TotalCorriendoU2` y `TotalParadaU2`). Por último se ejecuta otro `system.db.runPrepUpdate` para ingresar los valores correspondientes a nombre de máquina, día, `TotalDiaActiva`, `TotalDiaDetenida`, cantidad total de blísteres producidas (proveniente de `TotalCount`) y cantidad total de blísteres en buen estado producidas a la tabla `dailystats`. Esto se observa en la Figura 5.19

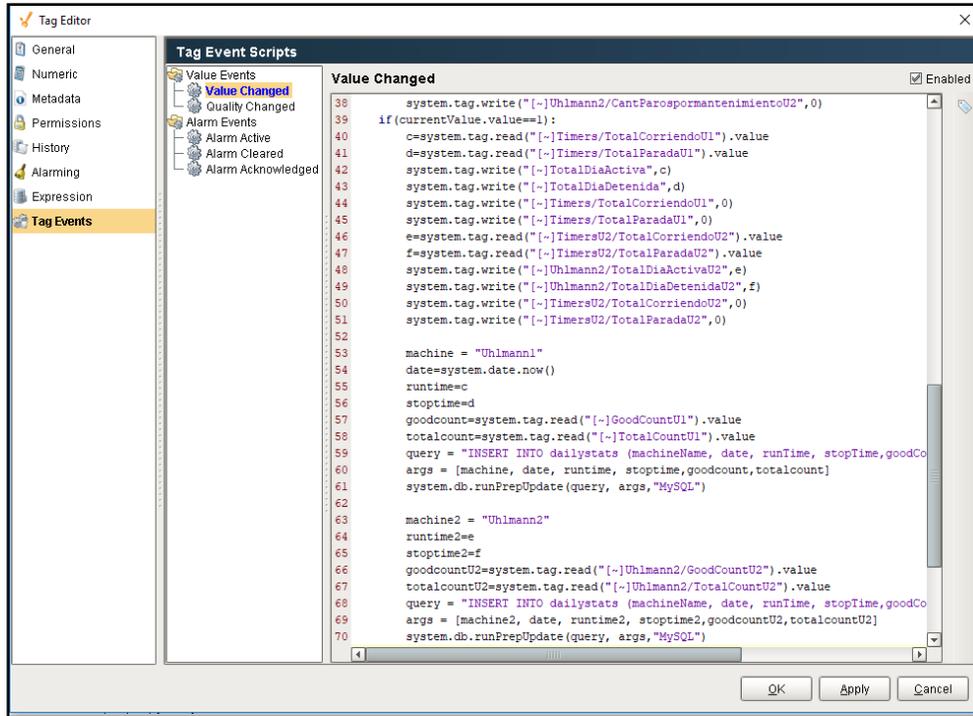


Figura 5.19 Código ejecutado 1 segundo después del momento programado para el reset.

[IGNITION (Elaboración Propia, 2018)]

La función de la variable “comodin” es simplemente alternar el valor actual para RunningU1 y RunningU2, esto lo realiza al puro inicio dentro del mismo *script* donde se realiza la lógica para la medición de tiempos activa y detenida de las máquinas. Es así como se logra entonces momentáneamente (lo que dure en ejecutarse el código del *timer script*) cambiar el estado de ambas máquinas y por lo tanto automáticamente se ejecuta lo explicado en la sección anterior y se suman el valor que haya en alguno de los cronómetros activos al valor total acumulado (antes de que este sea leído en el script del *reset*).

Con esto último se logra por ejemplo que en caso de que en el momento que suceda el *reset* la máquina se encuentra activa (desde hace 15000 segundos), antes de que se realice el ingreso a la base de datos del tiempo total activa del día, se invierte momentáneamente el estado de la máquina logrando que se sumen estos 15000 segundos al total acumulado. Para de este modo al realizarse el ingreso de los datos a la base de datos se cuenta con la información correcta.

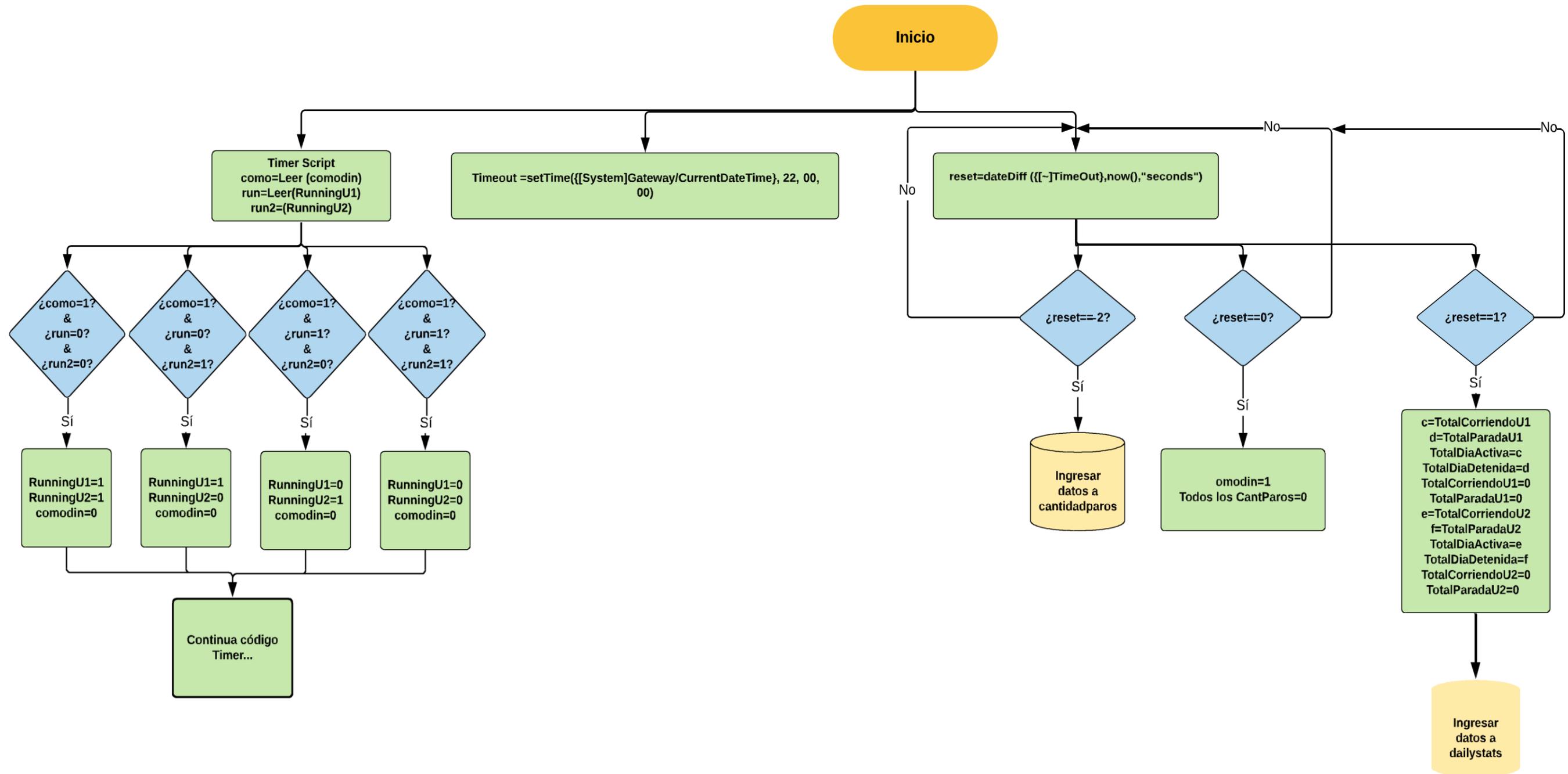


Figura 5.20 Diagrama de lógica de control implementada para el reset diario de las variables y la actualización de la base de datos.

[FLUIDCHART (Elaboración Propia, 2018)]

5.6.4.4 Tablas

Para alimentar las tablas de la base de datos se hizo uso de una interfaz gráfica en donde el usuario pueda brindar de manera sencilla la información requerida y en donde se limita al máximo que el usuario tenga que digitar los valores (esto se hace con el fin de evitar problemas y estandarizar los datos recibidos). Por este motivo en la medida de lo posible se implementaron listas desplegables para que el usuario seleccione una opción de estas o componentes de calendarios para que el usuario ingrese una fecha.

La plataforma presenta dos tablas principales, la primera de ellas permite agregar una máquina nueva a la planta, pensando en la escalabilidad del sistema. Se escoge el área donde se ubica la máquina a partir de una lista desplegable que está vinculada al conjunto de datos de la tabla “área” de MySQL, posteriormente se solicita un nombre para la máquina por añadir (el cual debe ser único para la tabla), una breve descripción de la máquina y por último el ciclo ideal brindado por el fabricante para la máquina (valor que luego será utilizado para el cálculo de indicadores). Todos estos valores son actualizados a la base de datos y se ofrece además la capacidad de eliminar una fila desde la plataforma.

Una vez añadida una máquina se habilita la posibilidad de crear turnos de trabajo para esta. Para lograr esto, se hizo uso de otra lista desplegable que se vincula mediante una consulta a todas las máquinas distintas que han sido añadidas a la tabla de máquinas en MySQL. Se ofrece además la capacidad de seleccionar desde un calendario la fecha y hora de inicio del turno o ciclo de trabajo para la máquina y la hora de finalización de este. Inmediatamente se calcula la duración del turno, utilizando una expresión para un componente de visualización numérico de datos que resta el tiempo en segundos entre el final del turno y el inicio de este. Cabe destacarse, que con el objetivo de brindar seguridad a los datos y confiabilidad al sistema, cada vez que se realice un registro sobre la tabla en la base de datos (en este caso corresponde a la tabla “*planned*”) se registra el usuario que realizó la acción y el tiempo exacto en que sucedió.

También se debe aclarar que se puede agregar más de un turno por día para cada máquina sin problema, al final el tiempo total planificado para cada máquina se calcula gracias al uso de SQL para el manejo de datos dentro de una base de datos, lo que simplifica bastante la obtención de estadísticas.

5.6.4.5 Estadísticas

A diferencia de la mayoría de las secciones explicadas anteriormente, para la obtención de los valores necesarios para el cálculo de los indicadores seleccionados, se hizo un completo uso de lenguaje SQL. Esto se debe principalmente a que en los casos anteriores se requería programar alguna solución que permitiera encontrar la información deseada. O bien se conocía la información pero no se había diseñado un sistema estable en donde se pudiese almacenar para poder ser consultado en cualquier momento y de manera sencilla. Por ello en la mayoría de este proyecto se están planteando soluciones que permitan adquirir información desconocida o realizando el monitoreo de los equipos, pero no es sino hasta el prácticamente el final del proyecto cuando se tiene la información deseada y se ha diseñado un sistema estable para su almacenamiento cuando se puede demostrar el potencial del proyecto.

Para la sección de estadísticas se cuenta con una única ventana en donde el usuario puede seleccionar la máquina que desee revisar. Este mecanismo de selección premia la escalabilidad del sistema. Ya que en caso de que se agregue una máquina más a la plataforma, al final del día productivo (en el momento que se dé la acción de *reset*) esta nueva máquina al igual que la Uhlmann 1 y la Uhlmann 2 actualizarán la tabla *dailystats* de la base de datos. Al suceder esto automáticamente se podrá seleccionar en la ventana de estadísticas la opción de visualizar estadísticas para la nueva máquina. Esto se logra vinculando la lista desplegable que realiza esta selección a los nombres de máquinas que se encuentren dentro de la tabla *dailystats* (y que por lo tanto tengan estadísticas).

Además se ofrece al usuario la posibilidad de seleccionar para cual día se desean ver las estadísticas de la máquina seleccionada. De este modo es posible recordar estadísticas anteriores y ver tendencias a lo largo del tiempo. Una vez seleccionadas tanto la fecha como el nombre de la máquina, cuando el usuario haga click en el botón de *Ok* se ejecutará un código asociado a un evento (click sobre el elemento). Este código se muestra en la Figura 5.21 y consiste básicamente de tres consultas que se realizan a la base de datos en MySQL a tres tablas distintas. La primera consulta retorna de la tabla *dailystats* los valores de tiempo de ejecución (*runTime*), cantidad de blísteres en buen estado producidos (*goodCount*) y total de blísteres producidos (*totalCount*), según los valores seleccionados para la máquina y la fecha.

La segunda retorna de la tabla *planned*, la suma de todas las columnas de duración para los valores de búsqueda seleccionados. Esta consulta devuelve el resultado en una columna

denominada 'totalduration'. Posteriormente dado que la columna siempre consiste únicamente de una celda (por el modo de selección y operación realiza) se procede a asociar el valor de esta a la variable del tiempo de producción planificado. Por último la consulta final consiste en retornar el Ciclo Ideal para la máquina seleccionada a través de la lista desplegable.

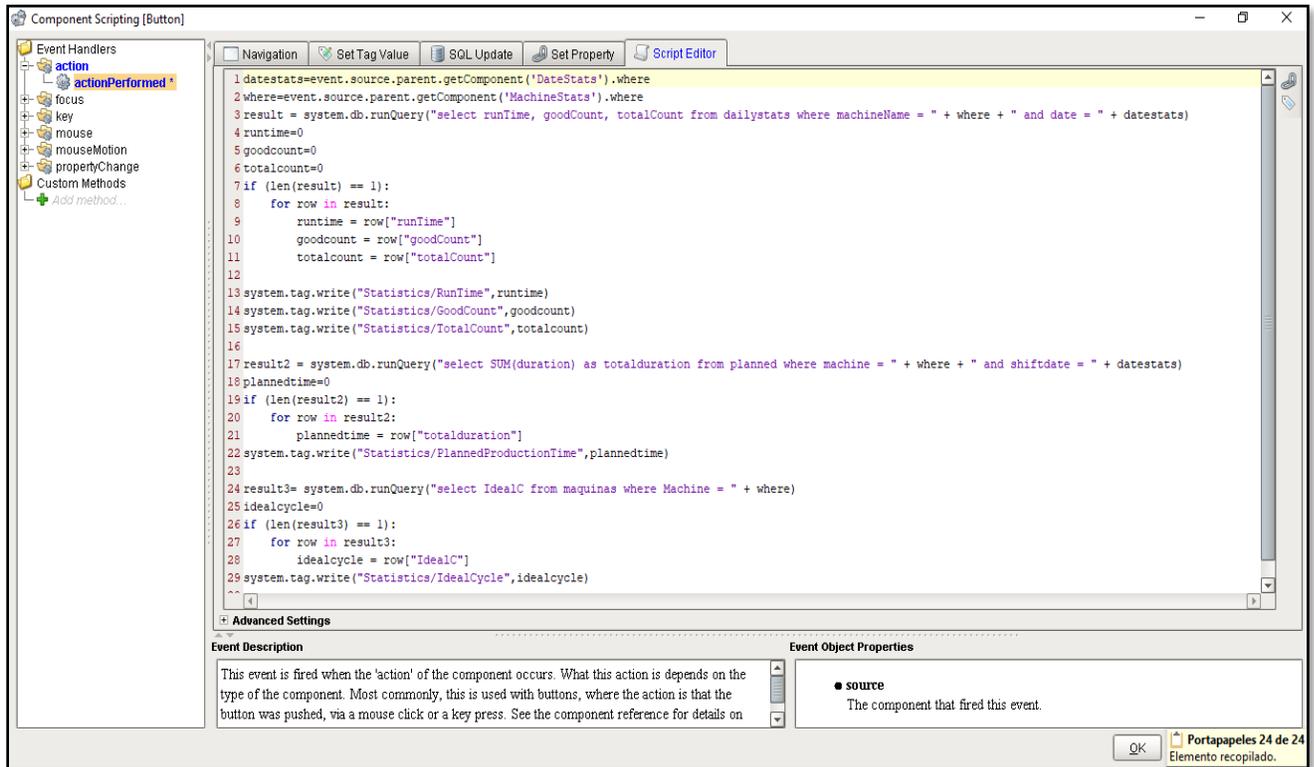


Figura 5.21 Consultas a base de datos para la obtención de los valores estadísticos de una máquina dada para una fecha dada.

Para finalizar se procede a realizar los cálculos de los tres indicadores del OEE a través de un expresión matemática en un componente de visualización numérico.

5.6.4.6 Reportes

Para la generación de Reportes se procede a utilizar el módulo propio de Ignition para creación de reportes de forma automática. Se configura además que los reportes sean enviados vía email a mi cuenta de correo personal todos los días a las 11:59:00 pm.

De manera muy similar a como se trabajó con la información en la sección anterior de estadísticas, el módulo de reportes requiere de fuentes de información provenientes de una base de

datos. Por ello y gracias al trabajo previamente realizado, se requiere solamente realizar una consulta SQL para cada tabla en donde se seleccione la información según el nombre de la máquina y el día actual. Con esto se logra generar un reporte diario para cada máquina con los principales resultados conseguidos por el sistema.

Capítulo 6 Resultados

Se recuerda al lector que en caso de contar con el archivo donde se encuentra este trabajo final de graduación podrá contar con dos insumos particularmente útiles para observar a detalle los resultados conseguidos en este proyecto. El primero de ellos es el archivo test_TEC.proj que en caso de contar Ignition instalado en el computador (incluyendo la versión gratuita) podrá importarlo e interactuar directamente con el proyecto creado. El segundo es un vídeo explicativo de lo realizado y obtenido de unos 30 min de duración que se realizó para la empresa, sin embargo ofrece una mirada al proyecto y sus resultados que difícilmente se pueden equiparar con imágenes en papel. Habiendo aclarado esto, a continuación se presentan los resultados del desarrollo de la plataforma que constituye la solución del proyecto piloto.

Como parte de la arquitectura de red seleccionada el proyecto puede visualizarse desde cualquier computador que se encuentre en la misma red donde se encuentra el servidor de Ignition instalado (en el caso de este proyecto piloto, esta ubicación sería un computador personal con sistema operativo Windows 10). Para poder lanzar un cliente desde la web se requiere únicamente conocer la IP que el computador con el servidor asume dentro de la red y desde cualquier buscador web (Safari, Google Chrome, Mozilla, etc) introducir esta dirección IP seguida por el puerto que utiliza el Gateway de Ignition (el 8088). Para averiguar la IP del computador, se puede obtener de manera sencilla desde la línea de comandos digitando el comando ipconfig.

La Figura 6.1 muestra un acceso inalámbrico al *Gateway* de Ignition desde una computadora Mac con sistema operativo MacOS. Desde este servidor es posible lanzar un proyecto como cliente de ejecución o inclusive lanzar un cliente como diseñador para trabajar sobre el proyecto.

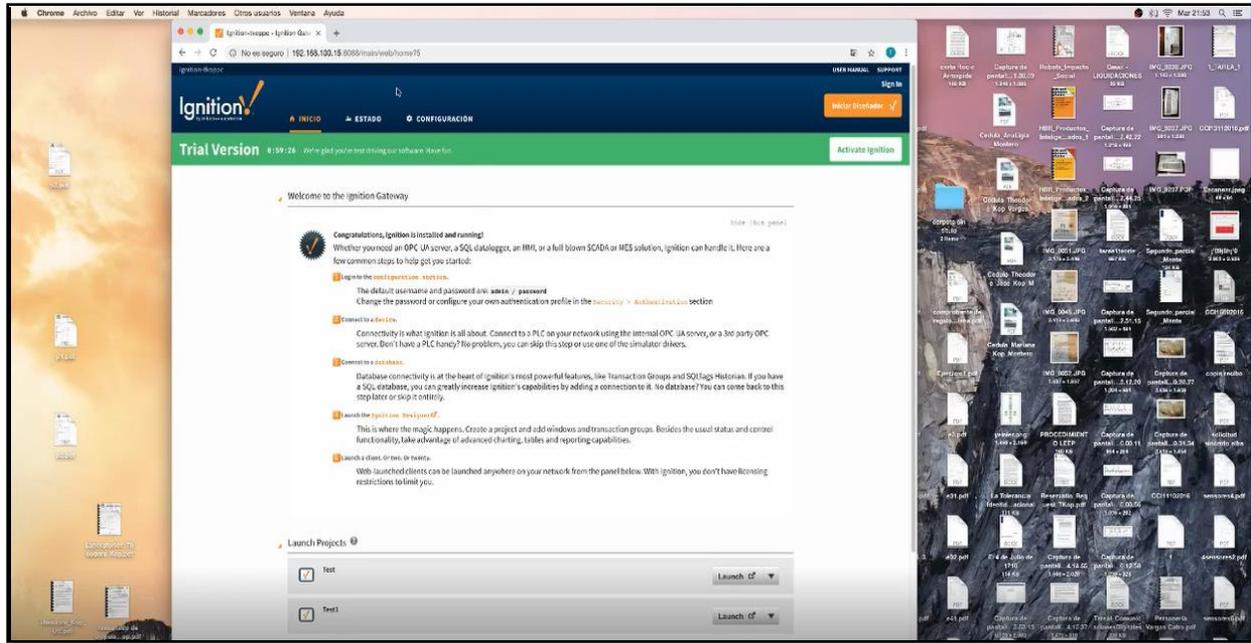


Figura 6.1 Acceso inalámbrico al Gateway de Ignition

[Elaboración Personal, 2018]

6.1 Principales ventanas

A continuación se presentan las principales ventanas configuradas en la plataforma.

6.1.1 Navegación

La Figura 6.2 muestra la ventana de Navegación final. Como se observa es una ventana bastante sencilla pero permite al usuario desplazarse por toda la plataforma e inclusive realizar un par de funciones extra. Además se cumple el principio de ser bastante intuitiva que se mencionó en el capítulo anterior. Se le otorga al usuario la posibilidad de seleccionar el Idioma del sistema entre Inglés o Español, además se indica el tiempo actual del sistema, el usuario que se encuentra conectado (esquina superior derecha), se puede bloquear la pantalla, cambiar de usuario y cerrar sesión. Por último en el centro de la pantalla de navegación se encuentra una lista desplegable que le sirve al usuario para seleccionar una sección. Las opciones de la lista se ven en la Figura 6.3, y consisten de las 3 principales estructuras que se discutieron en la descripción de esta solución.

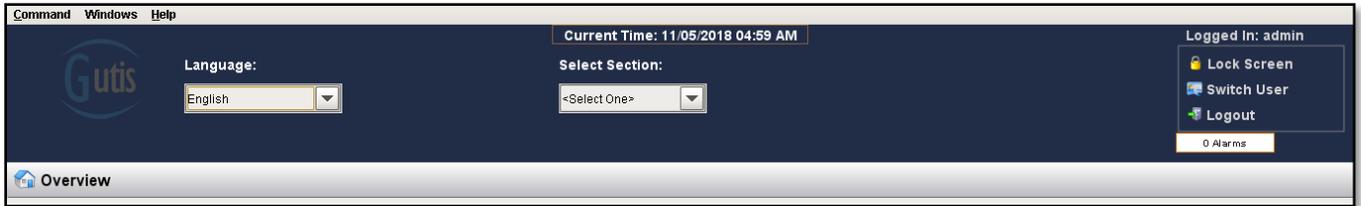


Figura 6.2 Ventana de Navegación.

[Elaboración Personal, 2018]

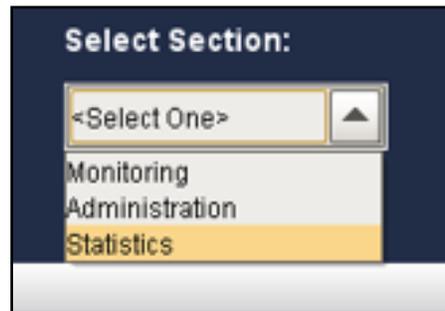


Figura 6.3 Lista de opciones en la ventana de Navegación.

[Elaboración Personal, 2018]

Si se hace click en Monitoring se observan las ventanas que se muestran en la Figura 6.4 (ignorar la ventana de Livestreams, esta es sólo para la empresa y no es parte de este trabajo final de graduación), si se hace click en Administration se observan las ventanas que se muestran en la Figura 6.5, y por último si se hace click en Statistics se observan las ventanas que se muestran en la Figura 6.6. Es así como esta ventana de navegación regula el flujo o movimiento de los usuarios de una ventana a otra.

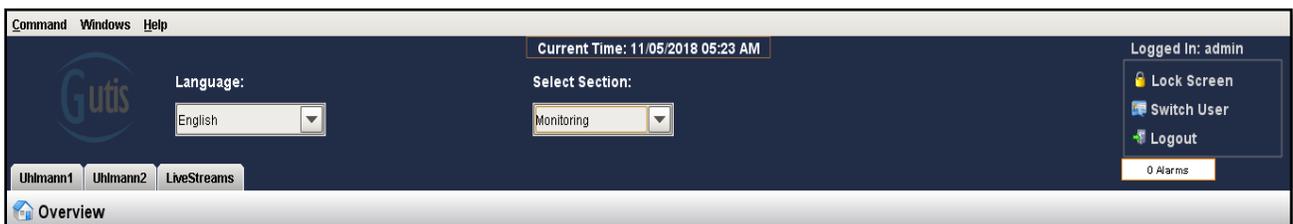


Figura 6.4 Ventanas de Monitoreo en la plataforma.

[Elaboración Personal, 2018]

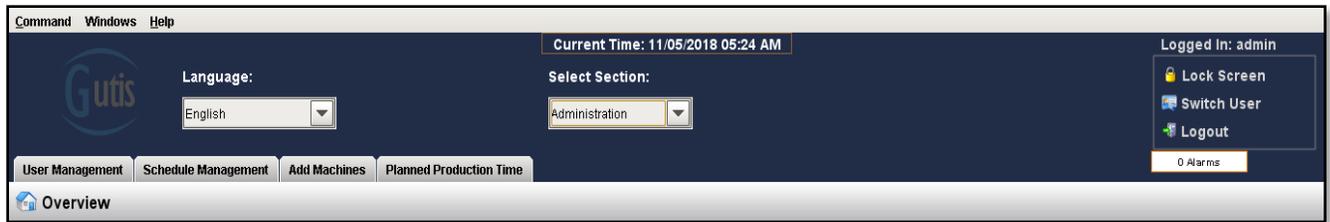


Figura 6.5 Ventanas de Administración en la plataforma.

[Elaboración Personal, 2018]

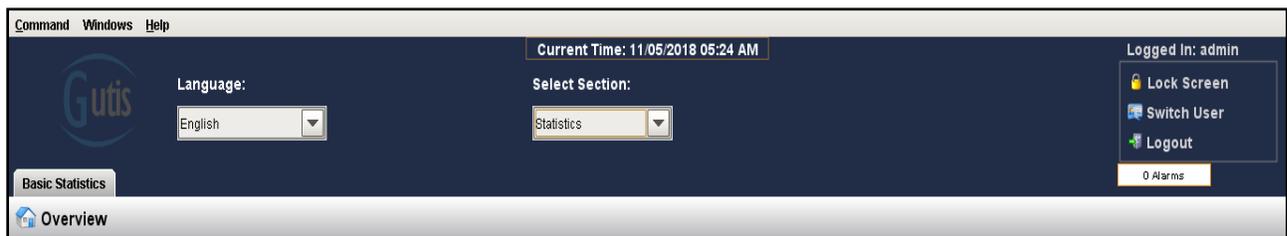


Figura 6.6 Ventana de Estadísticas Básicas en la plataforma.

[Elaboración Personal, 2018]

6.1.2 Monitoreo Uhlmann1 y Monitoreo Uhlmann 2

Dado que el monitoreo y hasta las ventanas para la Uhlmann 1 y Uhlmann 2 son iguales (con la salvedad de que cada cual usa sus variables correspondientes) se procederá a mostrar únicamente los resultados para la Uhlmann 1 en esta sección. Además se mostrarán los resultados más importantes que demuestran la plataforma realizando alguna de sus principales funciones.

La vista general de la ventana de monitoreo para la Uhlmann 1 se muestra en la Figura 6.7.

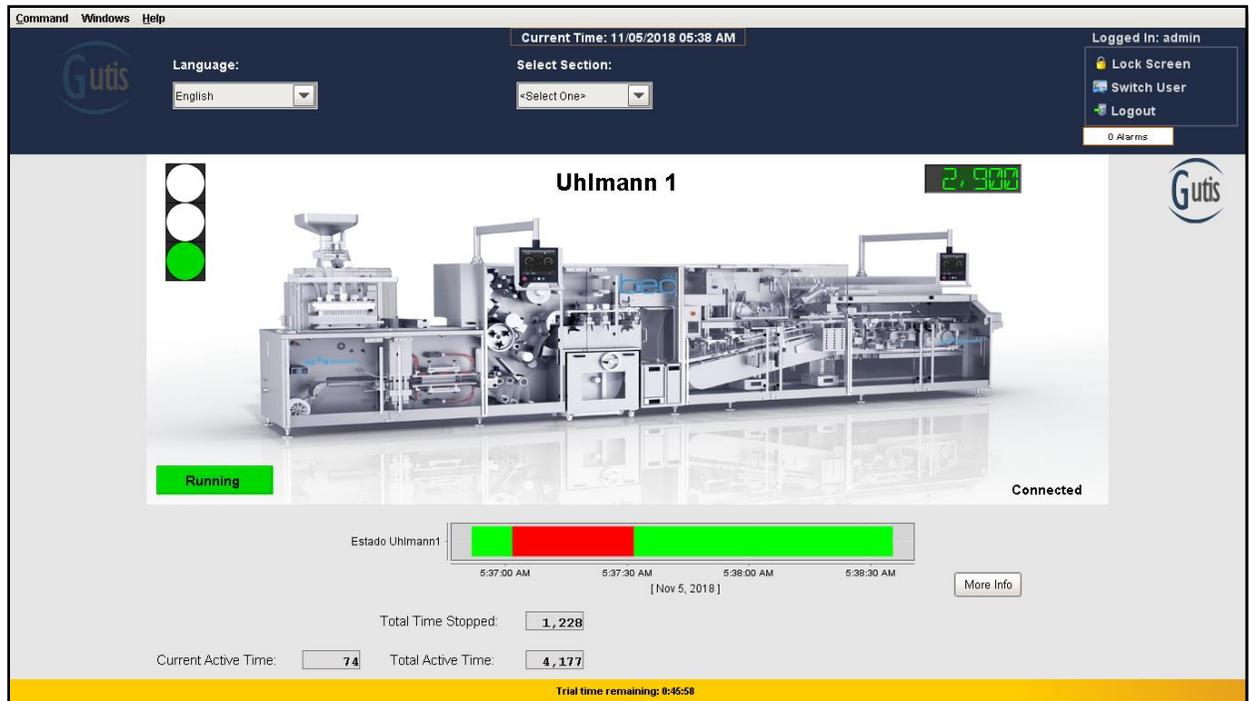


Figura 6.7 Vista General Ventana Monitoreo Uhlmann 1.

[Elaboración Personal, 2018]

Como se explicó anteriormente cuando la plataforma recibe alguna señal de la columna luminosa de que se ha ocasionado un paro el sistema es capaz de reconocerlo y consultar si así se requiere al operario el motivo de este (recordar que un paro por espera al igual que un paro por advertencia de material no requieren justificación del operario pues la señal de la columna luminosa confirma el motivo por si sola). Esta acción se muestra en la Figura 6.8 al momento en que recibió una LuzRojaContinua.

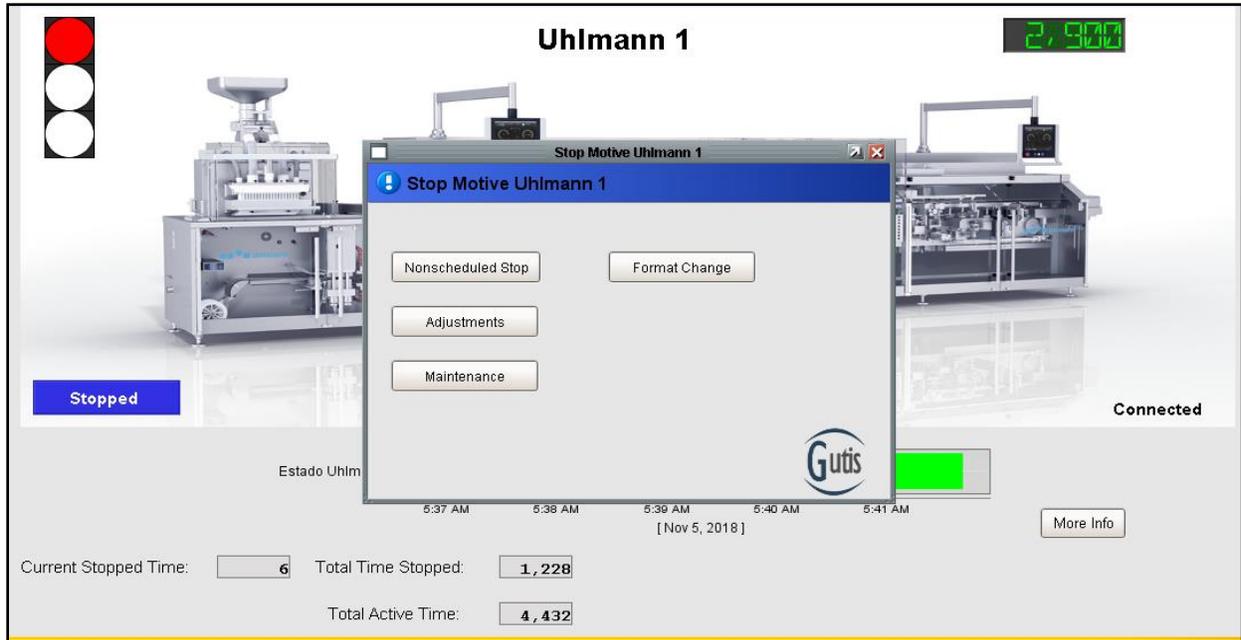


Figura 6.8 Ventana de justificación de Paro para la Uhlmann 1 al momento de detectar que la máquina se detuvo.

[Elaboración Personal, 2018]

En caso de que el paro se deba a una espera de una máquina postconectada o debido a una advertencia de control de alimentación de material, no se abre la ventana de justificación de paros pero si se registra el paro, y además cambia el estado en el gráfico de estado de la máquina.

Nótese que en la ventana se aprecian en todo momento 3 casillas numéricas, estas son precisamente las que actúan como cronómetros para reportar los tiempos que lleva actualmente activa o detenida (de acuerdo del estado de la máquina) y los tiempos totales acumulados. En la Figura 6.9 se muestra un ejemplo de esto y un acercamiento a e estos componentes.

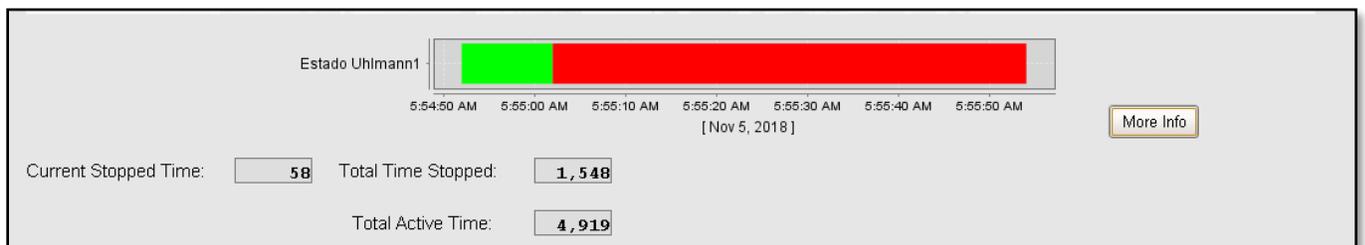


Figura 6.9 Cronómetros para registro de tiempos, gráfico de Estado de la Uhlmann1 en tiempo real y botón de más información.

[Elaboración Personal, 2018]

Nótese además de la Figura 6.9 que se observa claramente como el gráfico de Estado de Uhlmann 1 hace aproximadamente 1 min cambio de estado verde a rojo (paso de estar activa a detenida) si se observa este valor es coincidente con el tiempo que el cronómetro para el tiempo detenida actualmente indica. También si se es detallista se podrá notar que no aparece una casilla numérica para el tiempo actual activa. Esto se debe a que se programó de este modo para que en la pantalla solo aparezca el cronómetro que se encuentra activo. Es decir cuando se reanude la máquina en el ejemplo mostrado, el cronómetro para el tiempo actual activa aparecerá en la ventana mientras que desaparecerá el del tiempo actual detenida.

En la Figura 6.9 también se aprecia el botón para más información. A partir de este se puede acceder a la ventana de gráficos de motivos de paro para la Uhlmann1. Esta se aprecia en la Figura 6.10. Esta ventana ofrece una mirada detallada a los motivos detrás de los paros en la máquina. Por un lado en el gráfico superior muestra la cantidad de paros según el tipo que se han registrado durante el día. En el ejemplo de la Figura 6.10 se han registrado 3 paros no planificados, 3 por cambios de formato, 1 por espera, 1 por mantenimiento y 5 por materiales. Mientras que el gráfico de estado muestra cómo se comportó el sistema ante una secuencia en la que se dio un paro no planificado, seguido de un paro por cambio de formato, luego se puso en marcha la máquina y se volvió a detener debido a un problema de material. Como se observa el sistema es capaz de proveer esta información antes desconocida en Gutis y de desplegarla de una manera sencilla y fácil de comprender que además es respaldada por la base de datos en MySQL.

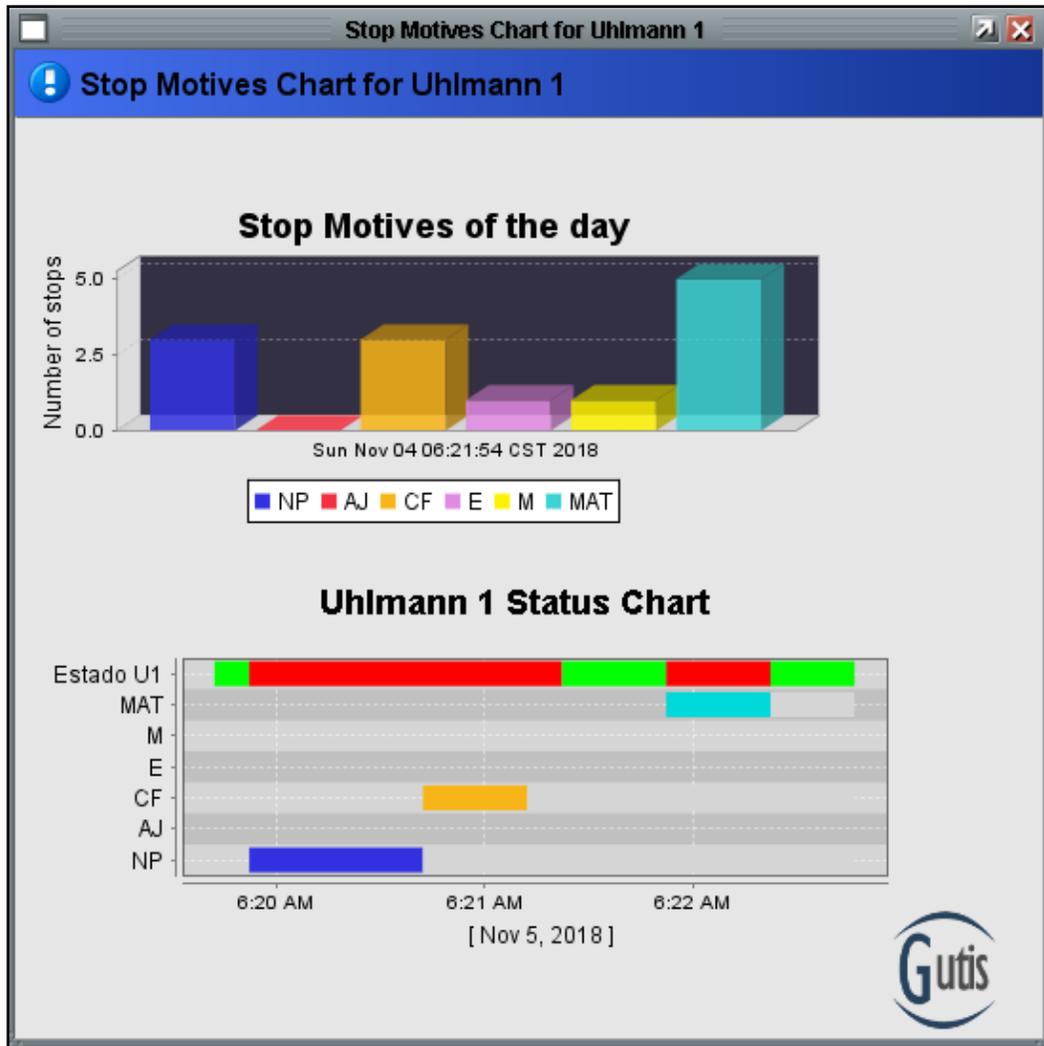


Figura 6.10 Ventana de Gráficos Motivos de Paro para la Uhlmann 1.

[Elaboración Personal, 2018]

6.1.3 Ventana de Manejo de Usuarios y Ventana de Manejo de Horarios

La ventana de manejo de usuarios se muestra en la Figura 6.11, en ella resulta bastante intuitivo lo que se permite a un usuario con acceso hacer. Tanto esta ventana como la ventana de manejo de horarios se basan en componente previamente elaborados por Inductive Automation (desarrolladores de Ignition) que comparten con todos sus usuarios, que solamente requieren de configuración para adaptarse a la plataforma de cada quien. Por ello no se profundizará en mostrar resultados de estas ventanas dado que realmente es muy poco lo que forma parte de estas que forme parte del diseño realizado. Más no obstante si se incluyen dentro del proyecto piloto debido a que

ofrecen una funcionalidad bastante conveniente. Además de que para la generación y envío automático de reportes y alarmas, Ignition hace uso de los horarios asignados a cada usuario para decidir que empleado es elegible y cual no.

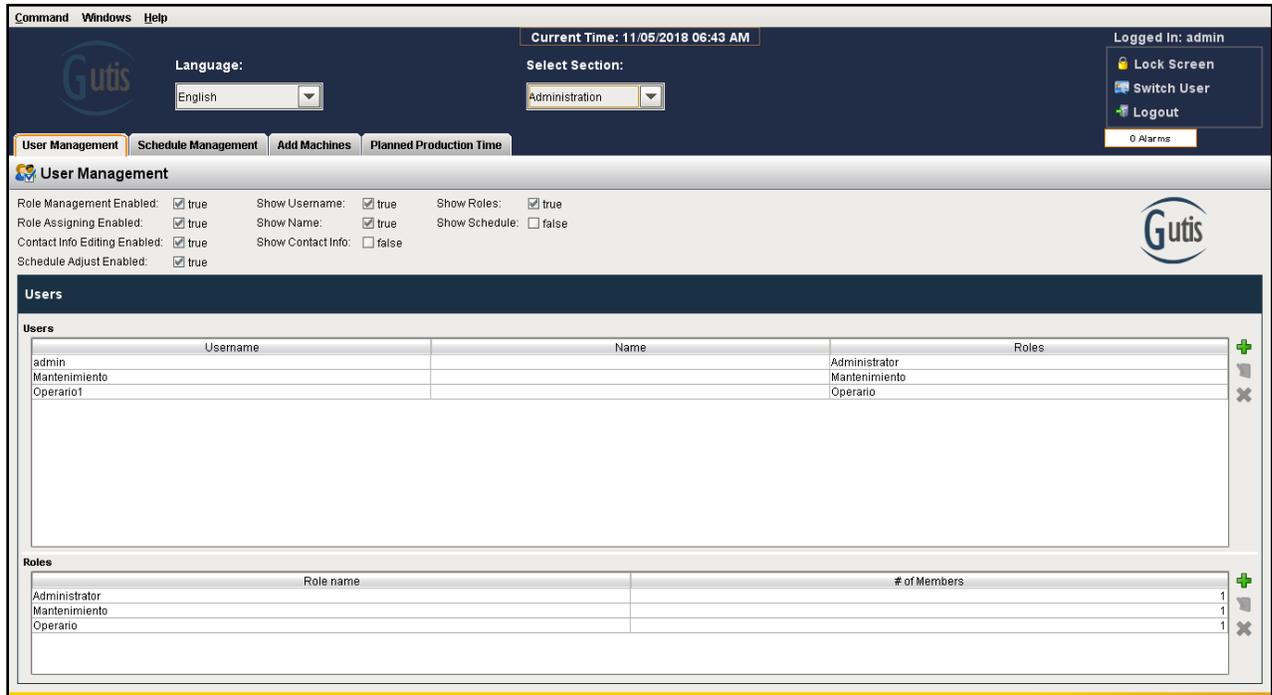


Figura 6.11 Ventana Principal para Manejo de Usuarios en la plataforma.

[Adaptación Propia, 2018]

La Figura 6.12 muestra la ventana principal para el manejo de horarios desde la plataforma.

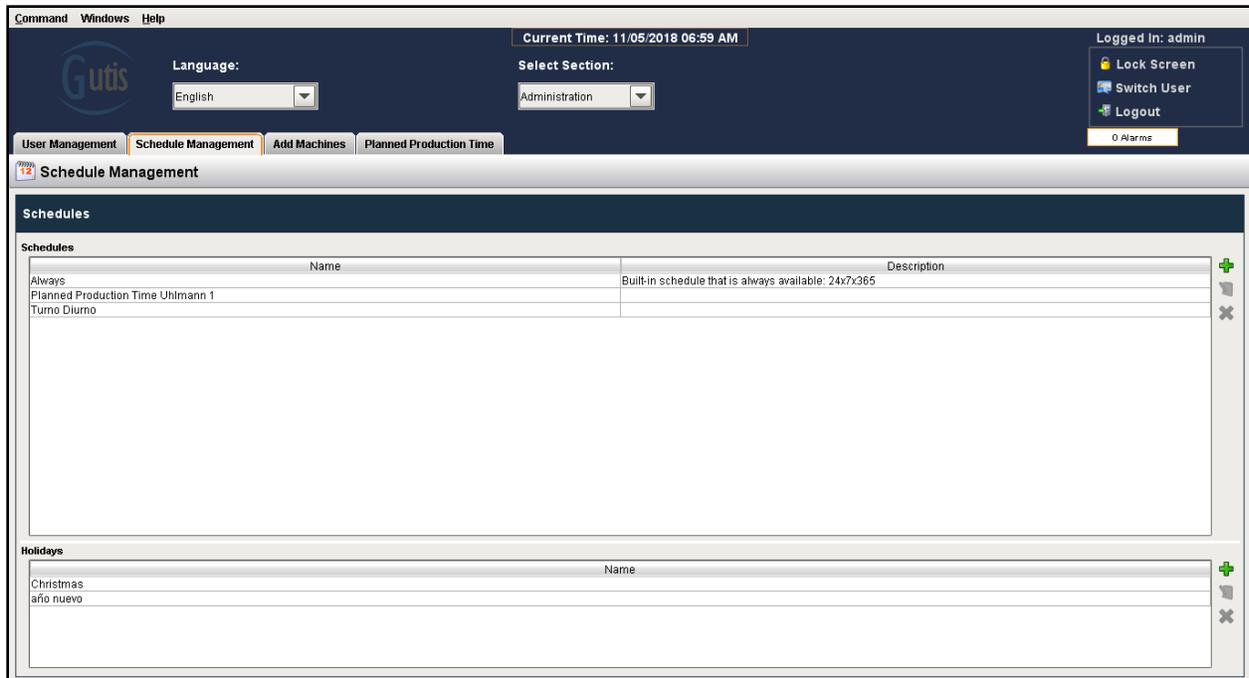


Figura 6.12 Ventana Principal para Manejo de Usuarios en la plataforma.

[Adaptación Propia, 2018]

6.1.4 Ventana para Añadir Máquinas y Ventana para Añadir Turnos de Trabajo a las máquinas

La ventana de añadir máquinas ofrece escalabilidad al proyecto, ya que permite al usuario con un rol con permiso agregar una máquina a la base de datos, y recordando el diseño que se realizó para la base de datos se tenía que la tabla de máquinas era básicamente el elemento central del diseño.

Como se observa en la Figura 6.13, la ventana consiste de una tabla que se actualiza con los valores de la base de datos, y desde la cual es posible eliminar una fila selecciona o añadir una nueva. Además también se puede apreciar que la componente de área para la máquina se realiza a través de una lista desplegable que está vinculada a la tabla de Area en la base de datos tal y como debería de ser para cumplir las relaciones de entidades propuestas en el diseño de la base de datos.

El resto de valores los puede introducir manualmente el usuario, pero nótese de la Figura 6.14 que si el usuario intenta añadir una máquina con el mismo nombre que otra en la tabla se produce un error que indica que el nombre de la máquina debe ser único. Mientras que la Figura 6.15 muestra la inserción correcta de una máquina nueva a la base de datos.

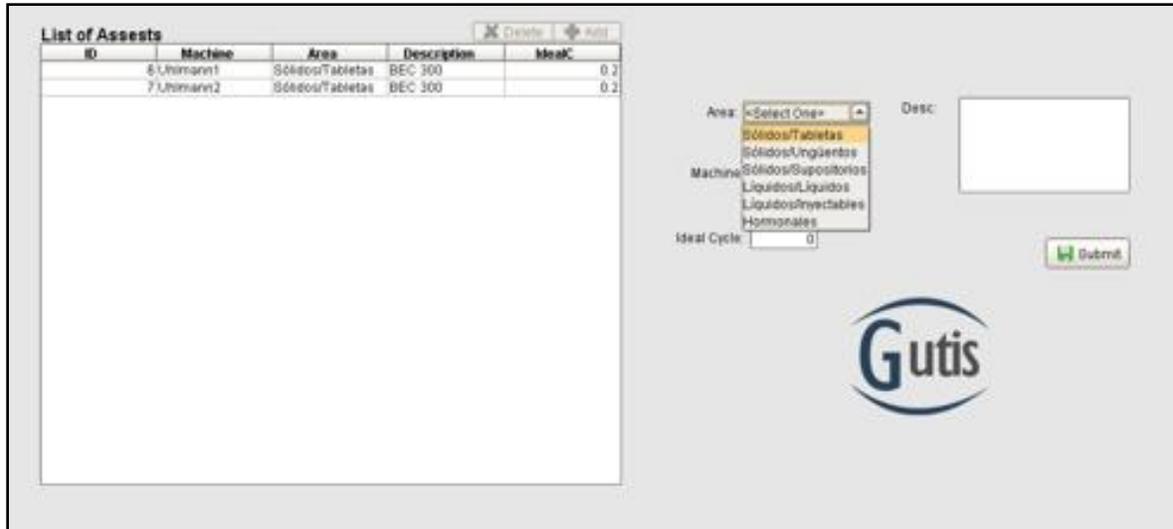


Figura 6.13 Vista General Ventana Añadir Máquinas.

[Elaboración Personal, 2018]

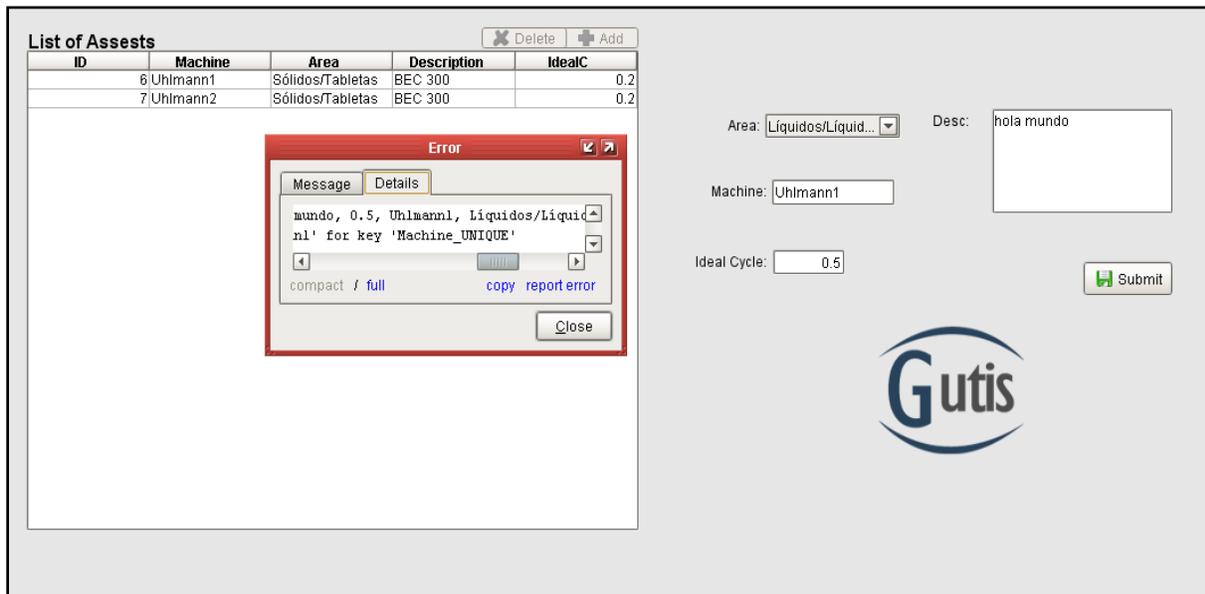


Figura 6.14 Error al tratar de introducir una máquina a la base de datos con el mismo nombre que otra dentro de la tabla.

[Elaboración Personal, 2018]

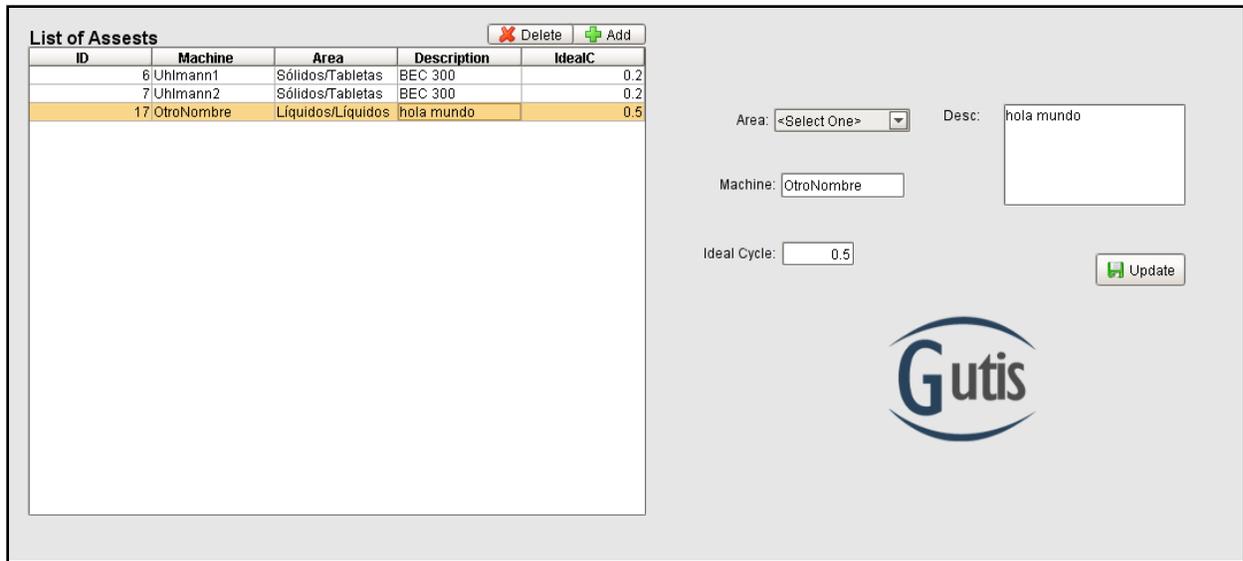


Figura 6.15 Inserción correcta de una nueva máquina a la base de datos.

[Elaboración Personal, 2018]

Por último la Figura 6.16 muestra lo que sucede cuando un usuario con un rol restringido intenta modificar la base de datos. El sistema reconoce el rol restringido del usuario y no permite realizar ninguna modificación; aún más el sistema no muestra ni los valores que hay almacenados en la base de datos y se muestra un error que le indica al usuario que no cuenta con un rol requerido.

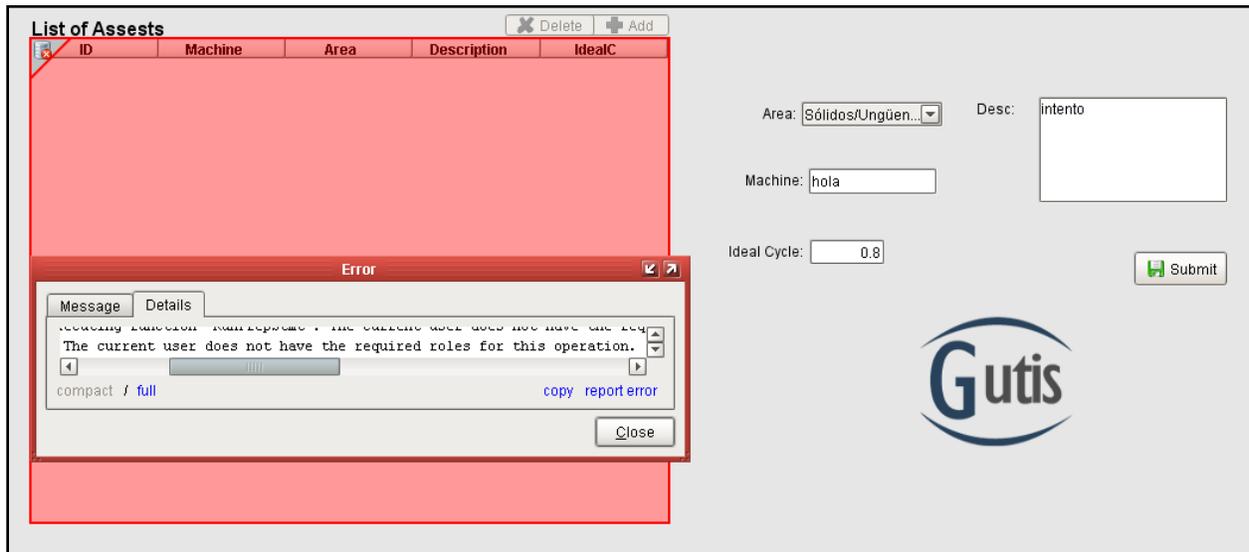


Figura 6.16 Negación de acceso a manipular la base de datos a un usuario con rol restringido.

[Elaboración Personal, 2018]

La ventana para añadir turnos de trabajo para las máquinas se muestra en la Figura 6.17, y es una herramienta que se diseñó a modo de que sea bastante flexible. Esta permite la programación de múltiples turnos de trabajo para las máquinas que han sido añadidas al sistema a través de la ventana de añadir máquinas o directamente desde la base de datos.

En esta ventana se permite inclusive permite planear turnos de próximos días para la máquina (esto se hace utilizando componentes de calendario). De este modo se busca que se registren los tiempos programados para cada máquina de una manera mucho más real (recordar que antes de esta herramienta se asumían turnos continuos de 6 de la mañana a 10 de la noche). Tal que si se planifica que cierto día habrá una actividad y el turno se verá interrumpido o si se extiende un turno laboral hasta las 12 medía noche o ante cualquier eventualidad que suceda, se pueda incluir cualquier jornada y no preocuparse por contabilizar la suma de cada turno, sino que el sistema se encarga de manera automática.

Este nuevo sistema también permite contemplar los paros por almuerzo de los operarios o los cambios de turno de estos y obtener al final día cálculos de indicadores mucho más apeados a la realidad en la empresa.

Filter: All

Planned Production Time: [Delete] [Add]

ID	Machine	Shift Start	Shift End	duration
53	Uhlmann1	Oct 26, 2018 7:00...	Oct 26, 2018 11:3...	16,200
54	Uhlmann1	Oct 26, 2018 12:3...	Oct 26, 2018 4:30...	14,400
55	Uhlmann2	Oct 26, 2018 7:00...	Oct 26, 2018 11:3...	16,200
56	Uhlmann2	Oct 26, 2018 1:00...	Oct 26, 2018 4:30...	12,600
57	Uhlmann1	Oct 27, 2018 8:00...	Oct 27, 2018 11:3...	12,600
58	Uhlmann1	Oct 27, 2018 2:00...	Oct 27, 2018 4:00...	7,200
59	Uhlmann2	Oct 27, 2018 7:30...	Oct 27, 2018 11:3...	14,400
60	Uhlmann1	Oct 29, 2018 7:00...	Oct 29, 2018 11:3...	16,200
61	Uhlmann1	Oct 29, 2018 1:00...	Oct 29, 2018 4:30...	12,600
62	Uhlmann2	Oct 29, 2018 7:00...	Oct 29, 2018 11:3...	16,200
63	Uhlmann2	Oct 29, 2018 2:00...	Oct 29, 2018 3:30...	5,400
64	Uhlmann1	Oct 30, 2018 7:00...	Oct 30, 2018 11:3...	16,200
65	Uhlmann1	Oct 30, 2018 1:00...	Oct 30, 2018 5:30...	16,200
66	Uhlmann2	Oct 30, 2018 7:00...	Oct 30, 2018 10:3...	12,600
67	Uhlmann2	Oct 30, 2018 1:00...	Oct 30, 2018 5:00...	14,400
72	Uhlmann1	Oct 31, 2018 7:30...	Oct 31, 2018 11:3...	14,400
73	Uhlmann1	Oct 31, 2018 1:30...	Oct 31, 2018 4:30...	10,800
74	Uhlmann1	Nov 1, 2018 7:00 ...	Nov 1, 2018 11:3...	16,200
75	Uhlmann1	Nov 1, 2018 1:00 ...	Nov 1, 2018 4:30 ...	12,600

Machine: Uhlmann1

Start Production Time: 10/26/2018 07:00 AM

Stop Production Time: 10/26/2018 11:30 AM

Duration: 16,200 [Update]

Gutis

Figura 6.17 Vista general de la ventana para añadir turnos de trabajo a las máquinas vinculadas a la plataforma.

[Elaboración Personal, 2018]

6.1.5 Ventana de Estadísticas

Esta ventana se programó de modo que se requiera una única ventana para la visualización de las estadísticas del OEE de todas las máquinas que han registro estadísticas diarias (que hayan registrado a la tabla dailystats en la base de datos). Para ello se hace uso de una lista desplegable que muestra precisamente todas las máquinas que han registrado estadísticas; y de un calendario que permite al usuario seleccionar para cual día desea revisar el OEE de la máquina que haya seleccionado. Es aquí cuando resulta realmente provechoso el uso de la base de datos para el manejo de datos y una base de datos que este bien interrelacionada.

De la Figura 6.18 a la Figura 6.21 se muestran algunos resultados históricos realizados a lo largo de 4 días de pruebas simuladas para la Uhlmann 1 y Uhlmann 2

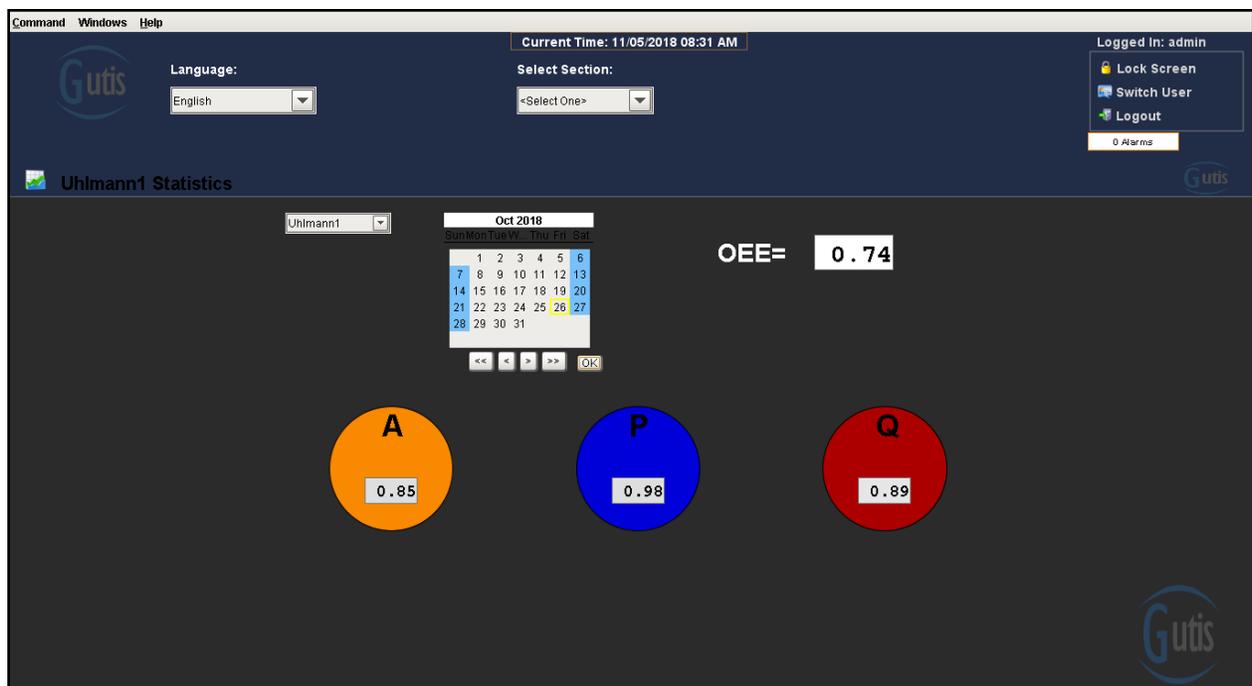


Figura 6.18 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 1 el día 26 de Octubre del 2018.

[Elaboración Personal, 2018]



Figura 6.19 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 1 el día 29 de Octubre del 2018.

[Elaboración Personal, 2018]



Figura 6.20 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 2 el día 29 de Octubre del 2018.

[Elaboración Personal, 2018]



Figura 6.21 Cálculo de OEE en la plataforma para prueba simulada de la Uhlmann 2 el día 30 de Octubre del 2018.

[Elaboración Personal, 2018]

6.2 Reporte

Parte de la funcionalidad que pretendía lograrse con esta plataforma era la generación de reportes de manera automática con información directamente monitoreada por el sistema. Gracias al módulo de reportes de Ignition, el cual permite utilizar como fuente de información los datos provenientes de una base de datos que alimente el sistema este proceso se agiliza bastante y automatiza. Ya que dado que al final de cada día productivo se hace un nuevo registro sobre la base de datos y la mayoría de tablas en la base de datos fueron construidas con un atributo para la fecha del registro (date), lo único que se hace es programar consultas SQL desde la vista de reporte con la fecha actual y de este modo se actualizan todas las variables que se utilicen dentro de este cada cambio de día.

Para el envío de reportes se configuro un servicio SMTP que utiliza el servidor de Gmail a través del puerto 465 y se creó un nuevo usuario de Gmail que es el que se encarga de hacer el envío diario del reporte. Este usuario es: gutismail007@gmail.com.

En la Figura 6.22 se muestra la serie de mensajes recibidos (enviados por este correo creado) a mi correo personal desde que se programó el servicio.

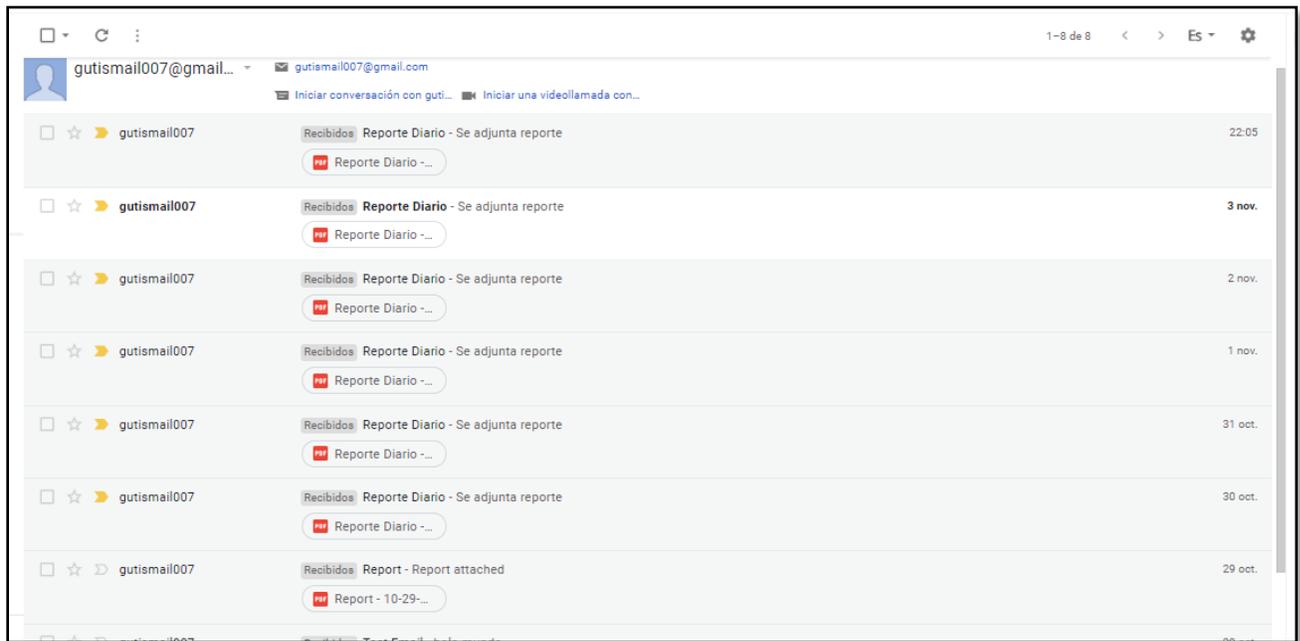


Figura 6.22 Flujo de correos recibidos a una cuenta de correo personal con reportes diarios.

Por último a continuación se adjunta un reporte diario recibido con datos de prueba simulado para que se visualice la estructura y contenido que los conforman.

REPORTE DIARIO GUTIS



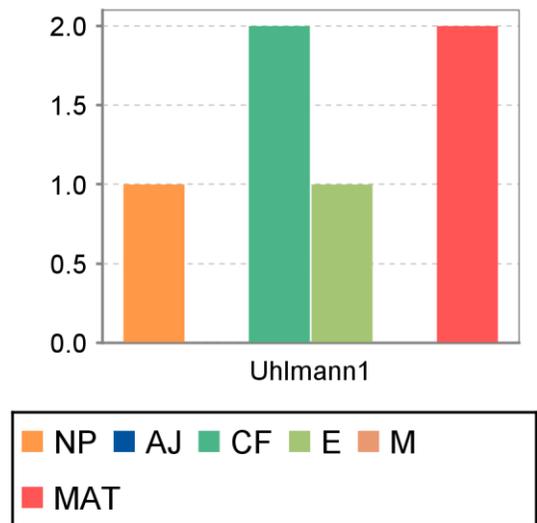
Uhlmann 1

Turnos Reportados para la Uhlmann 1

Uhlmann1	2018-10-30 07:00:00	2018-10-30 11:30:00	16200	admin
Uhlmann1	2018-10-30 13:00:00	2018-10-30 17:30:00	16200	admin

Motivos de Paro del día:

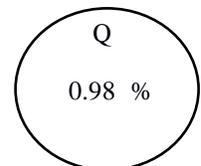
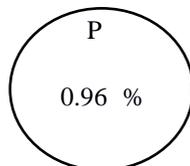
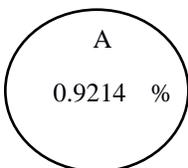
Ajustes (AJ):	0
Cambios de formato (CF):	2
Esperas (E):	1
Mantenimiento (M):	0
Paros no planificados (NP):	1
Paros por Material (MAT):	2



OEE Estadísticas:

Total de tabletas producidas:	143678
Total de tabletas de tabletas buenas:	140539
Tiempo Total Activa:	29854 s
Tiempo Total Detenida:	56546 s

OEE= 0.87%



REPORTE DIARIO GUTIS



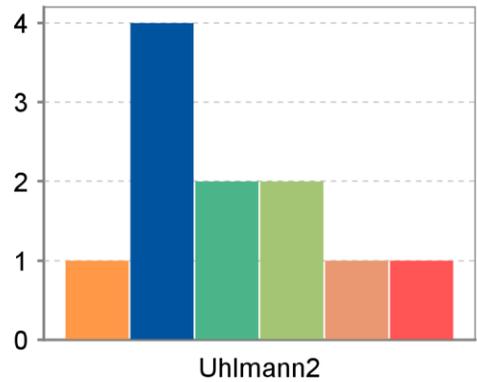
Uhlmann 2

Turnos Reportados para la Uhlmann 2

Uhlmann2	2018-10-30 07:00:00	2018-10-30 10:30:00	12600	admin
Uhlmann2	2018-10-30 13:00:00	2018-10-30 17:00:00	14400	admin

Motivos de Paro del día:

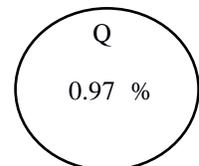
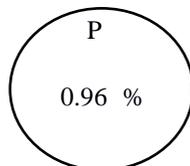
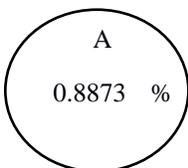
Ajustes (AJ):	4
Cambios de formato (CF):	2
Esperas (E):	2
Mantenimiento (M):	1
Paros no planificados (NP):	1
Paros por Material (MAT):	1



OEE Estadísticas:

Total de tabletas producidas:	114921
Total de tabletas de tabletas buenas:	112023
Tiempo Total Activa:	23956 s
Tiempo Total Detenida:	62400 s

OEE= 0.83%



Capítulo 7 Análisis Financiero

Para el análisis financiero de este proyecto se deben tener varias consideraciones presentes. En primer lugar se debe reconocer que al ser un proyecto que recopila información que hasta la fecha es desconocida a lo interno de Gutis no se tiene un punto de comparación confiable. En otras palabras el proyecto propuesto sugiere el primer intento confiable por conseguir, monitorear y analizar indicadores de desempeño para las máquinas seleccionadas. Lo cual por si solo es una apuesta por la que muchas compañías en la actualidad están invirtiendo (recordar información expuesta en el capítulo 3 Estado del Arte) e inclusive por la cual muchos analistas de tendencias del mercado y revisiones económicas advierten que se debe invertir si se desea competir dentro de los próximos años completamente inmersos en el contexto de la Industria 4.0. Algo que se alinea además con las políticas de mejora continua que se implementan en Gutis.

También debe reconocerse el potencial de expansión con el cual se diseñó la plataforma actual. Ya que en caso de que se llegue a implementar en la totalidad de la planta en Gutis y se le agreguen nuevas funcionalidades que aumenten su eficiencia, el impacto global del proyecto podría ser mucho mayor.

Inclusive el proyecto abre las puertas para que eventualmente se pueda implementar un sistema inteligente para el análisis de datos sobre la plataforma, y de este modo lograr que el sistema sea capaz de diagnosticar posibles fallos en máquinas, analizar tendencias dentro de la compañía y encontrar patrones, sugerir horarios convenientes para la realización de mantenimientos preventivos o hasta recomendar los horarios más convenientes para la producción de un tipo de formato o lote farmacéutico y hasta realizar por si solo configuraciones ideales en los ajustes de máquinas para replicar máximos desempeños de máquina en la totalidad de la planta.

En fin el potencial de crecimiento sobre la plataforma es enorme y su posibilidad de repercutir de manera positiva en los rendimientos financieros de la empresa también lo es. Sin embargo, regresando a la realidad actual del sistema, se tiene que por la naturaleza intrínseca del proyecto el potencial de la naturaleza se reduce a la capacidad de los responsables en la empresa de utilizarla. En otras palabras, dado a que el sistema diseñado actualmente no tiene ninguna acción de control (no afecta de manera directa en el proceso productivo), si el sistema es capaz de diagnosticar de manera correcta problemas recurrentes en las máquinas que tienen un efecto sobre los paros producidos y por ende en la disponibilidad de la máquina, la traducción de este conocimiento a una corrección de dicha situación depende exclusivamente de la aptitud del

responsable de hacer la toma de decisiones (al menos hasta que se agregue alguna funcionalidad de análisis inteligente de datos a la plataforma).

Por este motivo y habiendo hecho aclaraciones de las limitantes, se puede construir una estimación (que como toda estimación no deja de ser más que una aproximación razonada pero en base a supuestos).

7.1 Inversión Inicial Proyecto Piloto

La inversión Inicial requerida para la elaboración del presente proyecto piloto, que cumpliera con su objetivo de analizar la viabilidad de su implementación en la planta de Gutis, se muestra en la Tabla 7.1.

Tabla 7.1 Costos totales Inversión Inicial Proyecto Piloto

Descripción	Cantidad	Costo con I.V.I (SUSD)
<i>Licencia de Prueba Ignition 7.9.5 con todos los módulos habilitados (se vende por servidor)</i>	1	Gratuito
<i>MySQL Community Server 8.0.12 (GPL)</i>	1	Gratuito
<i>MySQL Workbench 8.0.12 (GPL)</i>	1	Gratuito
<i>Salario mensual TFG</i>	4	170
		Total Costos Proyecto Piloto = \$680 USD

[Elaboración Propia, 2018]

Como se puede apreciar el costo por desarrollar este proyecto piloto no representa un gasto significativo, esto se debe en parte a las licencias de software utilizadas durante el proyecto y a una maximización de los recursos desde la conceptualización de la solución, en donde se hizo énfasis en procurar que el mayor peso del proyecto se realizará sobre la plataforma desarrollada y se evitase al máximo depender de la implementación o uso de dispositivos externos para adaptar las señales o procesarlas externamente.

Cabe destacarse que al tratarse de una arquitectura de red con clientes web, se debe hacer uso de una arquitectura de red existente inalámbrica existente. En caso de no contar con una se debe incluir los costos de esta (que pueden variar considerablemente dependiendo de la

complejidad de la red que se desee diseñar y al tratarse de una red empresarial del tamaño y seguridad que se le otorgue).

7.2 Inversión Inicial Proyecto Implementado

En caso de que el proyecto se implemente en Gutis, la principal inversión por realizar es la de conseguir la licencia del *software* de Ignition. Actualmente Ignition ofrece tres paquetes distintos: El Ignition-Foundation con un coste de \$9950, la versión del *software* Ignition-Pro por un precio de \$14995, y la versión de Ignition Ultimate que tiene un precio actual de \$19095. La recomendación personal a la empresa es que se adquiriera la versión del *software* Pro. Esto debido a que según los módulos utilizados para la elaboración del proyecto piloto, este paquete es el más económico en ofrecer los módulos necesarios. Por ejemplo, la versión de Ignition-Foundation no ofrece el módulo de reportes ni tampoco el módulo para unir de manera sencilla los datos provenientes de dispositivos OPC y bases de datos SQL, módulos que si son incluidos desde la versión Pro. Representando esta la opción más económica en ofrecer las herramientas para replicar el proyecto elaborado en la totalidad de la planta y con la posibilidad de añadirse nuevas funcionalidades.

Para el caso de la base de datos, nuevamente se recomienda MySQL de Oracle, ya que su versión gratuita es completamente funcional y además su implementación en negocios lucrativos es completamente legal según lo regulado por la GLP. Por lo que la implementación de esta no representaría un costo, al menos en cuanto a licencias. Finalmente se requiere conectar los distintos PLC de las máquinas que se deseen integrar al sistema. Como se mencionó anteriormente Ignition cuenta con drivers para conexión a la mayoría de distribuidores de PLC, pero en todo caso que no se tenga alguno de dicha marca, el *software* también soporta la conexión de dispositivos OPC a través del protocolo Modbus TCP. Por lo que en toda caso este costo se resume al pago por los servicios o salario a un técnico o ingeniero que realice la conexión. Se estima un coste máximo de alrededor de los \$10 por hora para realizar el trabajo necesario para lo conexión y un total máximo de 2 horas por dispositivo que se desee conectar. De este modo se tiene que para replicar el proyecto en la Uhlmann 1 y Uhlmann 2 se requiere de una inversión inicial total que se resume en la Tabla 7.2, además se ha optado por dejar un porcentaje destinado a imprevistos de un 15% del

valor total (valor sugerido por el propio departamento de ingeniería en Gutis) e incluir el impuesto general sobre las ventas del 13%.

Tabla 7.2 Costos Totales Inversión Inicial Implementación en Gutis.

Descripción	Cantidad	Costo con I.V.I (SUSD)
<i>Licencia de Ignition 7.9.5 Pro (se vende por servidor)</i>	1	16 944.35
<i>MySQL Community Server 8.0.12 (GPL)</i>	1	Gratuito
<i>MySQL Worckbench 8.0.12 (GPL)</i>	1	Gratuito
<i>Trabajos Conexión de equipos.</i>	4	10
	Imprevistos (15%)	Subtotal= \$16 984.35 USD
		Total Costos Proyecto Piloto = \$19 532 USD

[Elaboración Propia, 2018]

7.3 Rentabilidad del Proyecto

Para determinar la rentabilidad del proyecto se proceden a realizar las siguientes suposiciones y aclaraciones:

- Se toma como base que refleje el panorama actual en la empresa los cálculos manuales realizados por el Ingeniero Gustavo Rodríguez. Aunque se advierte de la baja confiabilidad de estos datos.
- En base a estos datos se reporta una disponibilidad de la Uhlmann 1 de apenas un 40% y de la Uhlmann 2 de un 60% por lo que en promedio se supondrá que la situación actual en estas máquinas ronda un promedio del 50% de disponibilidad.
- Se supone además, que dado que el propósito de este proyecto es en gran parte maximizar el rendimiento de las máquinas, la herramienta será usada al máximo para la toma de decisiones promoviendo la mejora continua en la empresa. De este modo se tiene un potencial ideal de conseguir el 100% de disponibilidad en las máquinas integradas al sistema. Lo que representa un incremento del 100% en la disponibilidad (o el doble como resulte más sencillo).
- No se puede considerar el impacto sobre los factores de desempeño y calidad que se incluyen en este proyecto debido a la inexistencia de esta información. Por lo cual se asumirá una relación lineal (que se advierte que en la realidad no lo es así, pero dadas las condiciones ya mencionadas, la aproximación resulta válida para demostrar el argumento) entre la disponibilidad de máquinas y la cantidad de blísteres producidos por día productivo. De manera que este valor podría potencialmente llegar a duplicarse.
- Debido a que la información asociada al coste de producir una tableta es un valor confidencial en la empresa (el cual no es ni siquiera divulgado al departamento de mantenimiento ni al de producción) no es posible brindar el margen de utilidad asociado a cada tableta del área de sólidos y poder realizar un cálculo para determinar el margen de ganancia adicional resultante de duplicar la producción en el área de sólidos.

Capítulo 8 Conclusiones

La falta de conocimiento de información sobre los estados de las máquinas así como de sus motivos de paros y niveles de rendimiento imposibilitan la toma de decisiones inteligentes basadas en datos. Situación que complica las metas de mejora continua fijadas por la empresa. Con la escasa información disponible, se logró de manera poco confiable determinar que la disponibilidad en las máquinas Uhlmann 1 y Uhlmann 2 ronda apenas un 50% y los Ingenieros no pueden dar causa de tan bajo rendimiento.

Se propone, con la aprobación del Ingeniero Greivin Granados (supervisor en la empresa), la introducción de 6 tipos de motivos de paros. Estos son: Paro no planificado (NP), Paro por ajustes (AJ), Paro por cambio de formato (CF), Paro por espera a máquina postconectada (E), Paro por advertencia de material (MAT), Paro por trabajo de mantenimiento (M). De los cuales únicamente los paros debido a mantenimiento eran registrados y no por un sistema para el análisis de datos.

Se determina que las señales luminosas del semáforo industrial de las Uhlmann 1 y Uhlmann 2 constituyen una buena fuente para el diagnóstico del estado de las máquinas (siguiendo la nomenclatura que el fabricante utiliza para cada columna) ya que revela de manera inmediata si la máquina se encuentra detenida, activa, en espera o detenida por advertencia de material y no requiere mucha intervención con el cableado de las máquinas. Además es posible extraer directamente de la máquina información respecto a la cantidad de blísteres producidos y cantidad de blísteres producidos en buen estado.

Se es capaz de diseñar un software de monitoreo remoto para dos equipos de producción marca Uhlmann en la farmacéutica Gutis Ltda. El sistema es capaz de monitorear en tiempo real el estado de las máquinas, conocer los motivos de paros de estas, generar un indicador OEE para cada máquina y enviar reportes a través de un correo (email) programado.

La base de datos diseñada da respaldo al sistema y permite almacenar y manejar eficientemente los datos registrados. Además esta otorga a la plataforma de capacidades para realizar históricos.

El software seleccionado para el diseño y programación de la plataforma fue Ignition en su versión gratuita de Inductive Automation.

Se propone e implementa a escala una arquitectura de red básica pero flexible, basada en un único servidor con capacidad de conectar clientes de manera inalámbrica, incluso se logra conectar clientes móviles como un teléfono celular o una tableta, una única conexión a una base de datos dentro de la misma red. Los dispositivos con las señales de interés se conectan utilizando el protocolo OPC-UA, y para el caso de este proyecto piloto se hace la conexión con un PLC simulador que ofrece Ignition.

El proyecto introduce a Gutis Ltda. en el contexto de la Industria 4.0 y presenta la escalabilidad y flexibilidad suficientes como para permitir su implementación exitosa en la totalidad de la planta productiva en Pavas e incluir nuevas funcionalidades que expandan en el manejo y análisis de datos.

Capítulo 9 Recomendaciones

- Se recomienda, en caso de implementación, agregar algún mecanismo por el cual el usuario pueda seleccionar si desea ver estadísticas diarias (como se tiene actualmente), estadísticas semanales o estadísticas mensuales de un equipo incluido en la plataforma (nueva funcionalidad).
- Se recomienda en caso de implementación con más equipos que sean distintos a los de marca Uhlmann, realizar una programación orientada a objetos (utilizando los DataTypes de Ignition) y agregar a la tabla Machine de la base de datos una nueva columna denominada Type (o el nombre que se desee) en donde se indique el tipo de máquina que se desea agregar. Posteriormente crear una serie de *templates* para cada tipo de máquina que se desee que la plataforma soporte y realizar una programación de todas las funciones asociadas a estos tipos. Finalmente cuando el usuario introduzca una nueva máquina a través de la tabla en la ventana add machines (o añadir máquinas dependiendo del idioma seleccionado) en la sección de administración de la plataforma, leer el campo del tipo (Type) de máquina que el usuario seleccionó y generar automáticamente una nueva instancia del tipo de dato correspondiente con el nombre que el usuario haya introducido para ese registro en la tabla Machine. De este modo resultaría posible agregar máquinas de manera automática al sistema y comenzar a generar estadísticas y monitoreo para esta nueva máquina a través de un único click. Lo que aumentaría aún más la escalabilidad y eficiencia del sistema.
- Se recomienda, en caso de implementación, la compra de la licencia de Ignition-Pro. Este proyecto piloto demostró la capacidad de este software para desarrollo de SCADA o sistemas similares y su potencial es muy alto. Sin ninguna duda se recomienda la compra de este software sobre otros competidores. Mientras que para el caso de la licencia, se recomienda Ignition-Pro debido a que cuenta con todos los módulos necesarios para la replicación de este proyecto piloto e inclusive otros que podrían aumentar su funcionalidad. Cabe destacar que este proyecto piloto hizo uso del módulo móvil para visualización del proyecto desde un cliente en un celular, y este módulo no se incluye por

defecto con la compra de Ignition-Pro. Deberá evaluarse la necesidad o no de incluir este módulo como un adicional o simplemente abandonar esta funcionalidad para futuras implementaciones.

- Se deberá capacitar a todos los miembros del personal que vayan a utilizar la plataforma, para ello inclusive se cuenta con el apoyo audio-visual de un vídeo que contiene una explicación del modo en que se utiliza la plataforma y los resultados obtenidos para el diseño de este proyecto piloto. En caso de realizar modificaciones, el apoyo audio-visual para documentar y capacitar el personal siempre se recomienda (criterio personal).
- El proyecto piloto y la base de datos se deberán trasladar a un servidor estable que reciba mantenimiento y cuidados a cargo del departamento de TI.
- En la medida de lo posible se recomienda conectar (enlazar) la plataforma o la base de datos con SAP para ofrecer una mayor redundancia de los sistemas en la compañía. Además ambos sistemas podrían compartirse información valiosa para las decisiones administrativas.
- En caso de una futura implementación más robusta, se recomienda utilizar un experto para el análisis de datos de manera inteligente o con conocimientos en *machinelearning*, ya que a partir de los históricos generados de los equipos conectados se podrían predecir fallos en equipos, hallar patrones, sugerir los momentos ideales para comenzar la producción de acuerdo al tipo de materia prima o factores externos que de otro modo no se podrían contemplar o inclusive realizar la configuración de ajustes ideales para aumentar al máximo la producción de una máquina.
- Se recomienda revisar cada factor del OEE de manera individual, esto con la intención de poder analizar patrones y tendencias propias del proceso que se esté monitoreando, tales como la calidad, el desempeño y la disponibilidad de las máquinas.

Capítulo 10 Referencias

Badami, V. (2016, December 21). MQTT protocol – Powering Facebook messenger to IoT devices [Web log post]. Recuperado 29 Agosto, 2018, de <https://www.hackerearth.com/blog/internet-of-things/mqtt-protocol/>

Castellanos, L. F. (2011, July 25). LA INTEGRACIÓN DE LOS SISTEMAS SCADA A NUESTROS AMBIENTES DE PRODUCCIÓN [Web log post]. Recuperado 08 Octubre, 2018, de <http://producir-eficiente.blogspot.com/2011/07/la-integracion-de-los-sistemas-scada.html>

De Vries, J. (1994). The Industrial Revolution and the Industrial Revolution. *The Journal of Economic History*, 54(2)

Derhamy, H., Eliasson, J., Delsing, J., & Priller, P. (2015). A survey of commercial frameworks for the Internet of Things. 2015 IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), 1-8.

Fonseca, C., & T. (2017, January 16). ¿Cómo capacitarnos para la Cuarta Revolución Industrial? Recuperado 16 Setiembre, 2018, de <https://es.weforum.org/agenda/2017/01/en-la-cuarta-revolucion-industrial-se-solicitan-habilidades-complejas-en-ambitos-caoticos>

Gil, P., Pomares, J., & A., C.H. (2010). *Redes y transmisión de datos*. San Vicente del Raspeig: Publicaciones de la Universidad de Alicante.

Granados, G. (2018, 16 de junio). Entrevista personal.

Gutis. (2017.). *INNOVACIÓN Y TECNOLOGÍA PARA LA VIDA*. Recuperado 23 Junio, 2018, de <http://www.gutis.com/>

Haykin, S.: *Sistemas de Comunicación*. Primera edición. Limusa Wiley, 2001.

Inductive Automation. (2018, September 18). Meeting 21CFR 11 Standards for Pharma. Recuperado 20 Octubre, 2018, de <https://icc.inductiveautomation.com/discover-gallery-detail/2018/wunderlich-malec-engineering>

Montúfar, L. S. (2006). Informática I: Un enfoque constructivista. México: Pearson Educación.

Mudge, N. (2014, December 22). Nick Mudge Ignition Software Consulting & Development. Recuperado 25 Octubre, 2018, de <http://nickmudge.info/?page=4>

P. Varga, F. Blomstedt, L. L. Ferreira, J. Eliasson, M. Johansson, and J. Delsing, "Making system of systems interoperable - the core components of the arrowhead technology framework (accepted for publication)," IEEE Internet of Things Journal, August 2015.

Perpinan, A. (2004). Administración de redes GNU/Linux. Fundación Código libre Dominicano, 2004.

PricewaterhouseCoopers. "Leveraging the Upcoming Disruptions from AI and IoT." PwC, PwC, 10 Oct. 2017, www.pwc.com/gx/en/industries/tmt/publications/ai-and-iot.html.

Rake, R. (2018, Febrero). Supervisory Control and Data Acquisition (SCADA) Market (Publicación No. IC_18544). Recuperado 12 Octubre, 2018, de Allied Market Research website: <https://www.alliedmarketresearch.com/scada-market>

Rodríguez Penín, A. (2011). Sistemas SCADA Guía Práctica. Barcelona: Marcombo.

Singh, S. K. (2011). Database systems: Concepts, design and applications. Pearson Education India.

Tanenbaum, Adrew S. y Wetherall, David J. (2012). Redes de computadoras

Technavio. (2015, Setiembre 21). Mobile Products and Acceptance of the MAC Approach Will Drive Growth in the Global SCADA Market [Web log post]. Recuperado 12 Octubre, 2018, de <https://www.technavio.com/blog/industrial-mobile-products-and-acceptance-of-the-mac-approach-will-drive-growth-in-the-global-scada-market>

Tomasi, Wayne. Sistemas de Comunicaciones Electrónicas. 4 Ed. México: Prentice-Hall, 2003.

Torres, V. (2012). 5 Minutos de Matlab. Recuperado 24 Setiembre, 2018, de <http://5minutosdematlab.blogspot.com/2011/09/codigo-para-encontrar-la-senal.html>

Wagner, H. (n.d). CAGR vs Average Annual Return: Why Your Advisor Is Quoting the Wrong Number [Web log post]. Recuperado 12 Octubre, 2018, de <https://investinganswers.com/education/how-invest/cagr-vs-average-annual-return-why-your-advisor-quoting-wrong-number-1996>

Capítulo 11 Apéndices

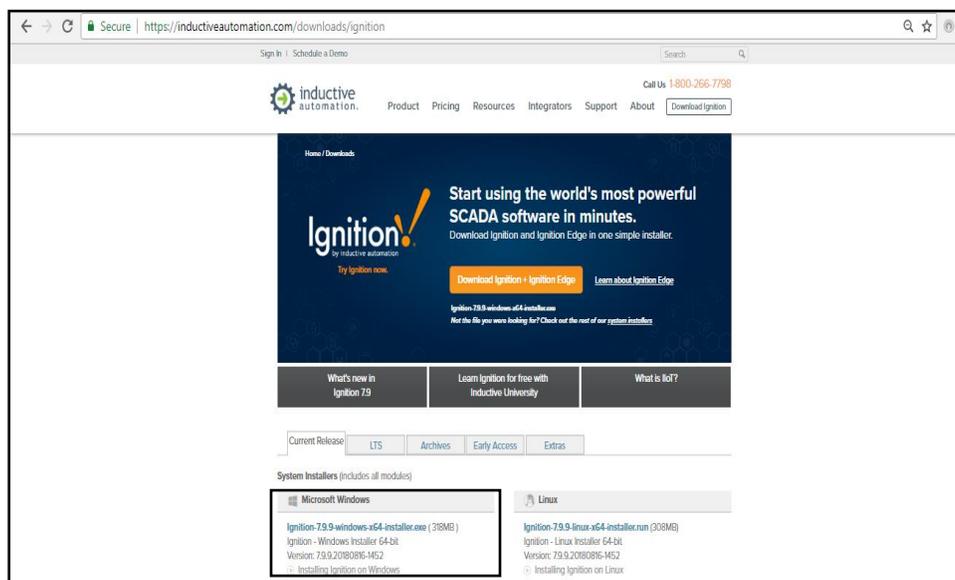
Manuales de Usuarios (Elaboración Propia, 2018)

Manual Usuario para la instalación de Ignition 7.9.5 en Windows

El presente manual tiene como intención servir de guía para la instalación de la plataforma de Ignition en un sistema operativo de Windows. Se instalará inicialmente la versión demo aunque en cualquier momento que lo desee podrá comprar alguna licencia o módulos disponibles sin tener que realizar una nueva descarga. Como nota importante se debe considerar que Ignition se encuentra escrito en Java por lo que Java 8 debe estar instalado en los sistemas que utilizarán Ignition. Si no cuenta con Java en el equipo por favor ingrese a (<https://www.java.com/en/>) e instálelo antes de continuar con la instalación de Ignition.

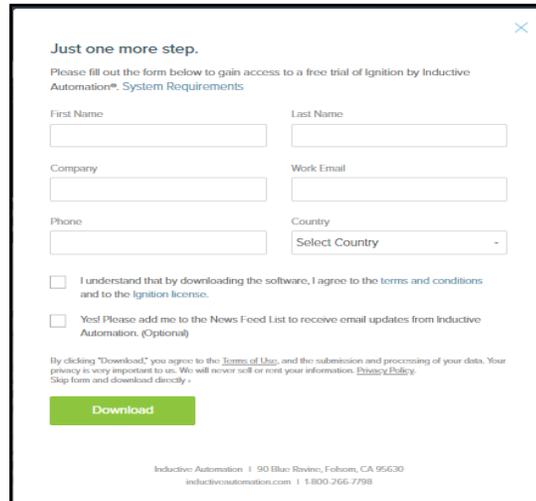
Paso 1

- Diríjase a la página oficial de Inductive Automation a la sección de descargas>ignition (<https://inductiveautomation.com/downloads/ignition>). Bajo la sección de versiones recientes (*Current Releases*), encuentre y seleccione el instalador para Windows de 64 bits (dependiendo de la arquitectura del equipo, este manual asume una arquitectura de 64 bits).



Paso 2

- Si no cuenta con un usuario de Inductive Automation, deberá crearse uno para poder continuar con la instalación. Haga *click* en *Download* al completar el formulario solicitado.



Just one more step.

Please fill out the form below to gain access to a free trial of Ignition by Inductive Automation®. System Requirements

First Name Last Name

Company Work Email

Phone Country

I understand that by downloading the software, I agree to the terms and conditions and to the Ignition license.

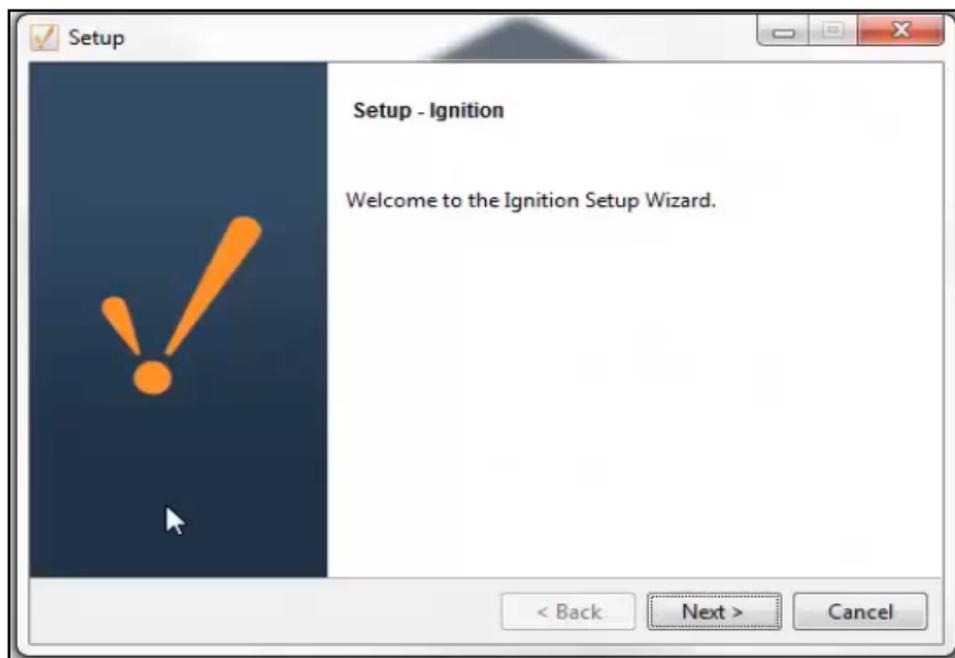
Yes! Please add me to the News Feed List to receive email updates from Inductive Automation. (Optional)

By clicking "Download," you agree to the [Terms of Use](#), and the submission and processing of your data. Your privacy is very important to us. We will never sell or rent your information. [Privacy Policy](#).
Skip form and download directly >

Inductive Automation | 90 Blue Ravine, Folsom, CA 95630
inductiveautomation.com | 1.800.266.7798

Paso 3

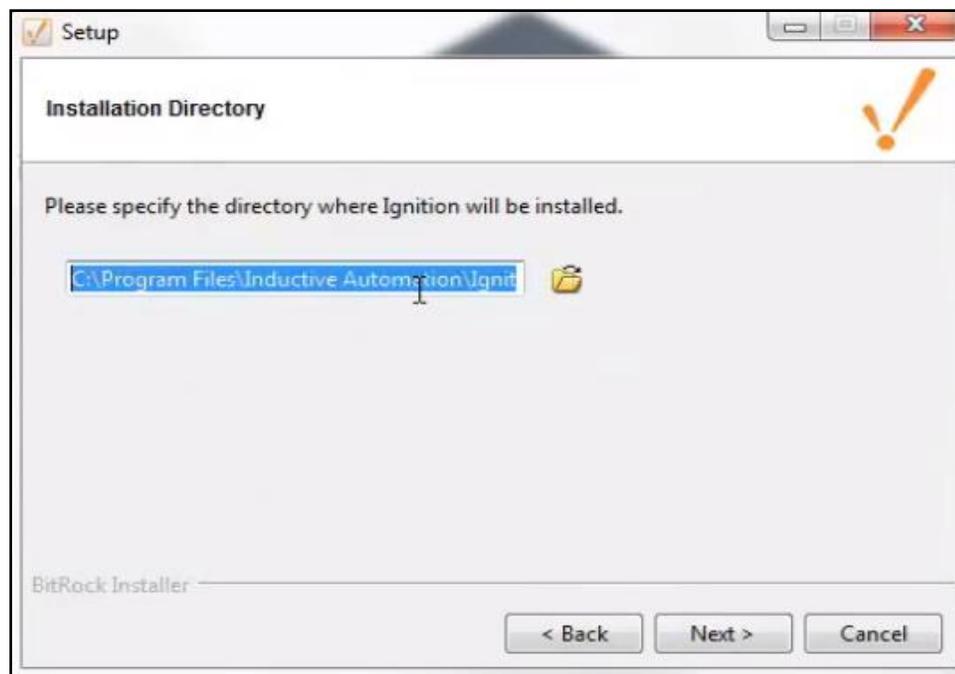
- Ejecute el instalador y haga *click* en *Next*>.
- Lea y acepte los términos y condiciones de la licencia para Ignition. Haga *click* en *Next*>.





Paso 4

- Indique el directorio en donde desea instalar Ignition. Por default se asume la siguiente dirección: **C:\Program Files\Inductive Automation\Ignition**. Haga *click* en *Next >*.



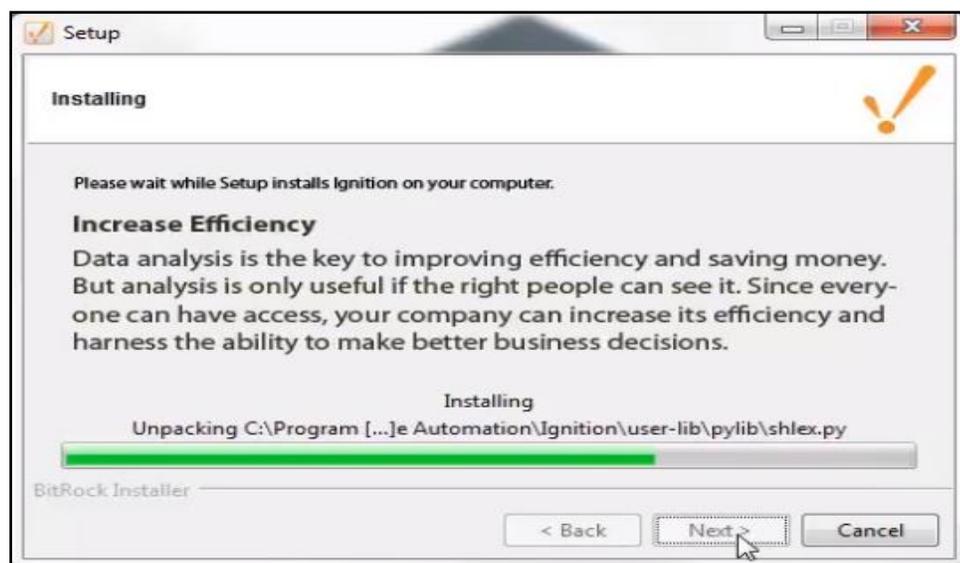
Paso 5

- Seleccione el modo de instalación. El modo típico de instalación (*Typical*) incluye Ignition con una serie de módulos predeterminados, mientras que el modo personalizado (*custom*) permite la opción de escoger cuales módulos se desean instalar específicamente. Sin importar el modo de instalación que se seleccione, Ignition siempre permite instalar o renovar nuevos módulos en cualquier momento que se desee. Para efectos de este manual se selecciona el método típico de instalación. Haga *click* en *Next>*.



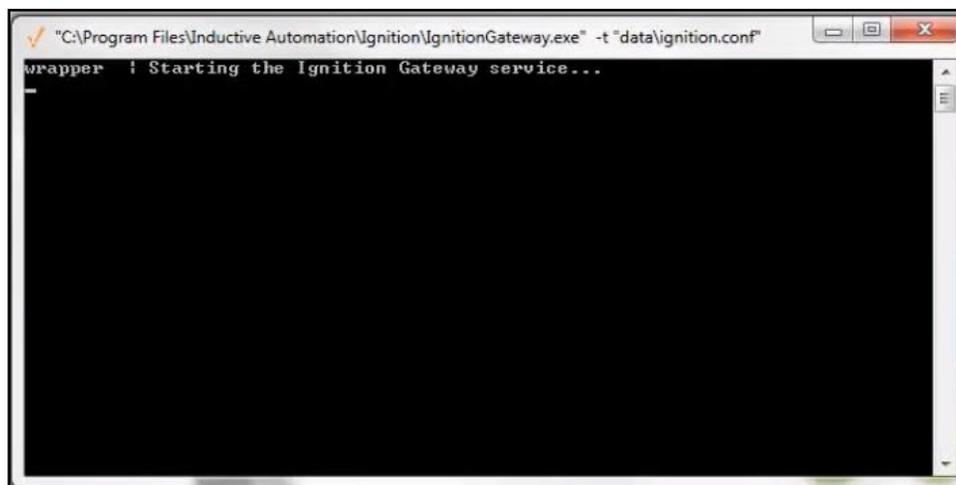
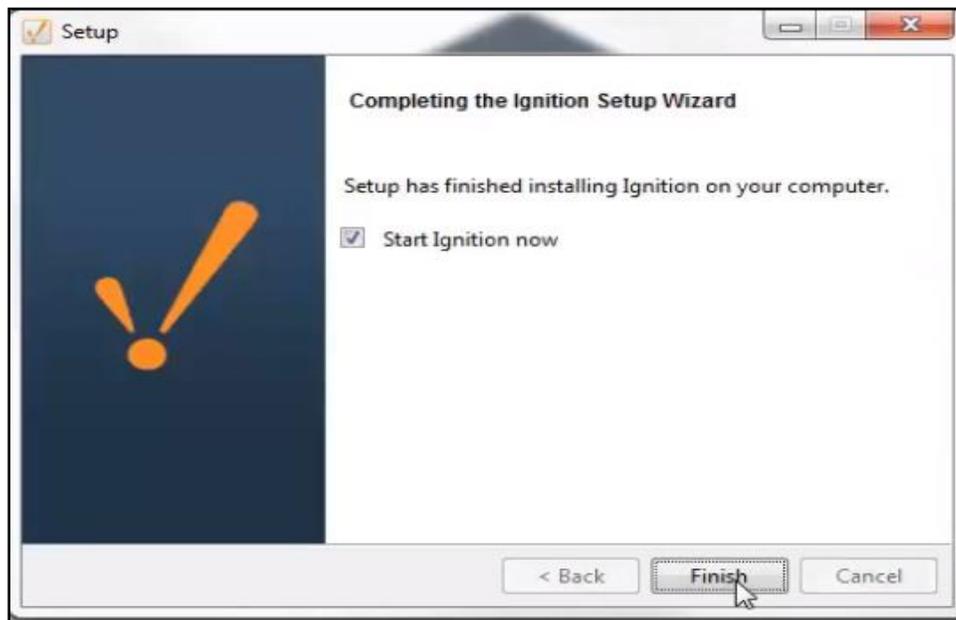
Paso 6

- La instalación deberá tardar un poco menos de 3 min. Al finalizar por favor haga *click* en *Next>*.



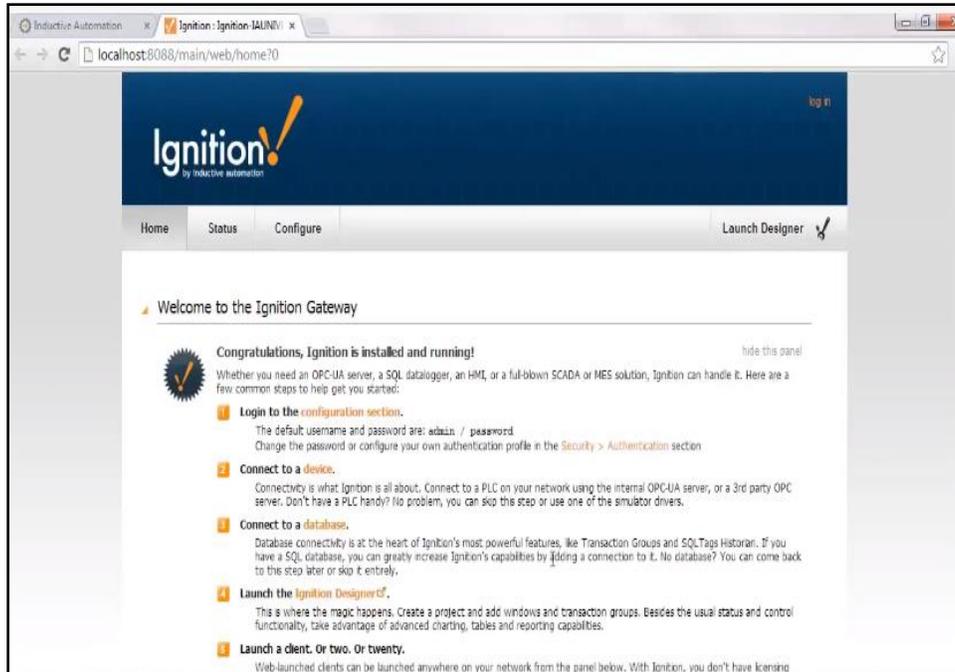
Paso 7

- Estamos por finalizar la instalación de Ignition. Marque la casilla *Start Ignition Now* y haga *click* en *Finish*.
- Se comenzará a ejecutar un programa para inicializar el *Gateway* de Ignition, este es un servicio que ejecuta todas las tareas dentro de la plataforma, y que “corre” en el servidor donde se haya realizado la descarga del software de Ignition. El *Gateway* de este modo instala un servidor web de Ignition para de este modo poder ingresar desde cualquier equipo que se encuentre en la misma red que el servidor (<http://localhost:8088>) utilizando cualquier buscador web.



Paso 8

- Verifique que Ignition se haya instalado correctamente y que esté activo.
- ¡Hemos terminado el proceso de instalación de Ignition satisfactoriamente!



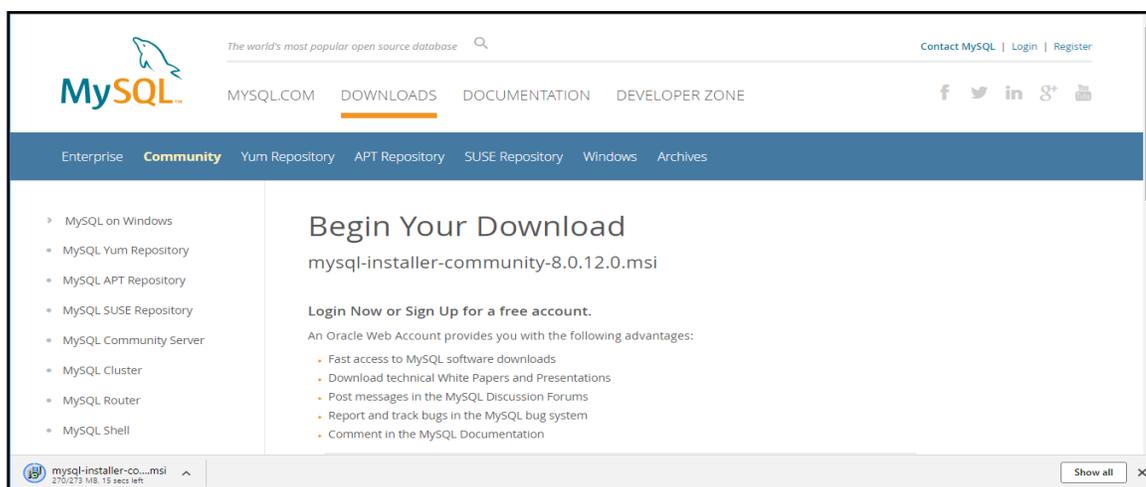
Manual Usuario para la instalación de Base de Datos MySQL

El presente manual tiene como intención servir de guía para la instalación de una base de datos MySQL que sirva para abastecer la plataforma de Ignition que se encuentra instalada en el equipo que ha sido destinado como servidor central (si aún no ha leído el manual de usuario para la instalación de Ignition puede visitar el siguiente link: [Manual Usuario Instalación Ignition](#)) que detalla paso a paso este proceso). Muchas de las funciones avanzadas de Ignition requieren de una conexión a una base de datos externa. Cabe destacarse que no existe ninguna dependencia o conflicto si se instala el software de Ignition antes de tener una base de datos que vincularle o bien si primero se instala la base de datos antes de tener el software de Ignition instalado.

Se debe aclarar también que Ignition se puede conectar a más de una base de datos y estas no necesariamente deben ser MySQL, sin embargo con fines didácticos se elabora este manual que comparte muchos de los pasos necesarios para la instalación de cualquier base de datos, y muchos otros propios de MySQL (por este motivo si se desea instalar otra base de datos distinta a MySQL se recomienda buscar la documentación propia de esta para realizar cualquier instalación).

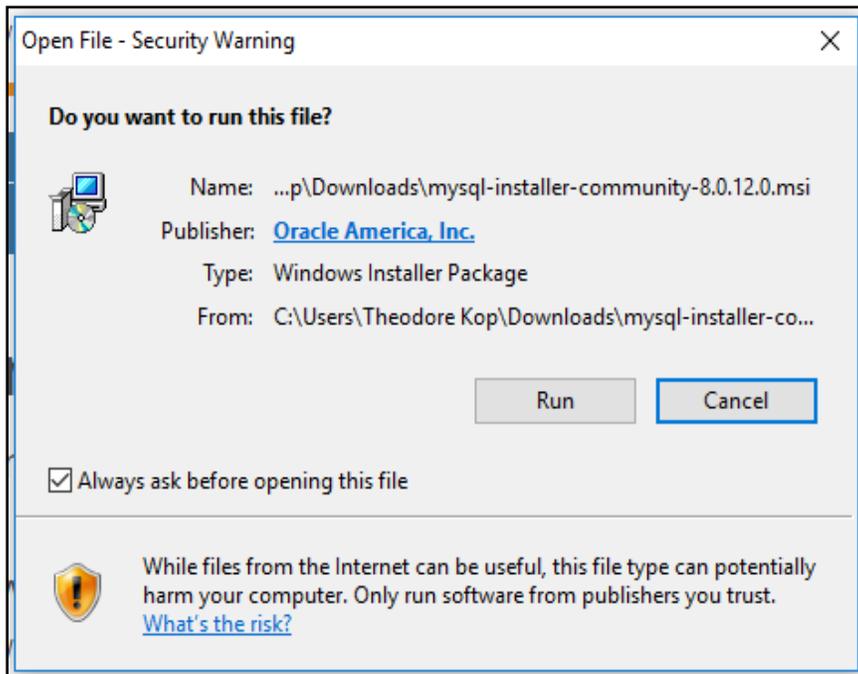
Paso 1

- Desde cualquier buscador/navegador web diríjase a la página principal de MySQL (www.mysql.com).
- Haga *click* sobre la pestaña de descargas (*downloads*) y luego sobre Comunidad (Community)
- Encuentre y descargue el instalador para MySQL y recuerde la versión que esta por instalar (cuando se realice la conexión con Ignition esta información resulta valiosa).



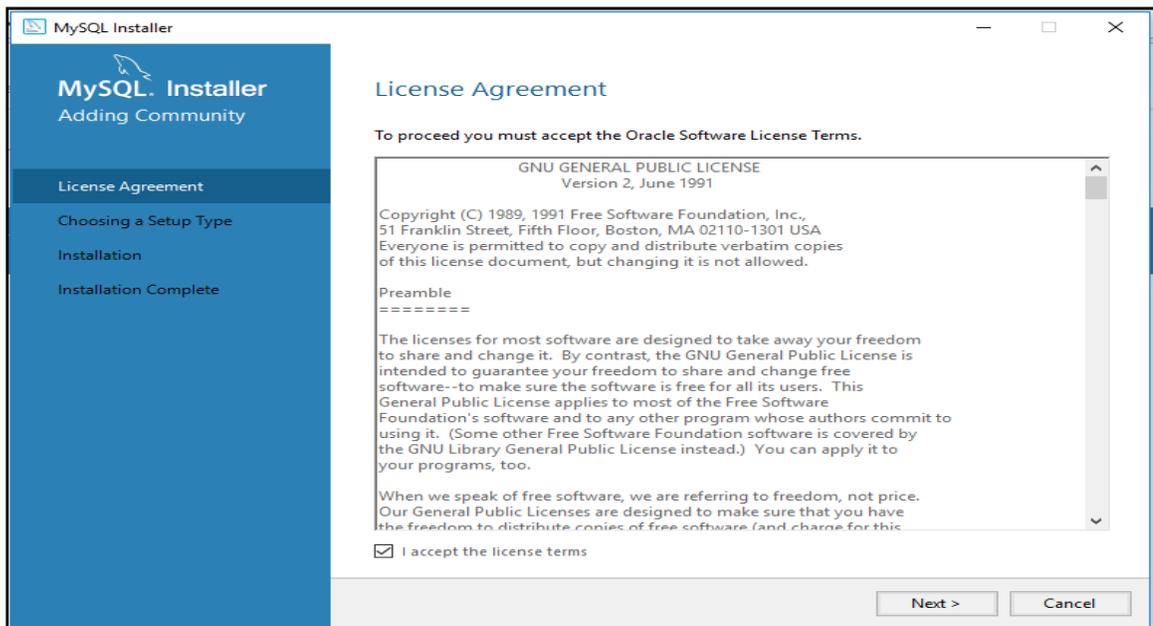
Paso 2

- Ejecute el instalador.



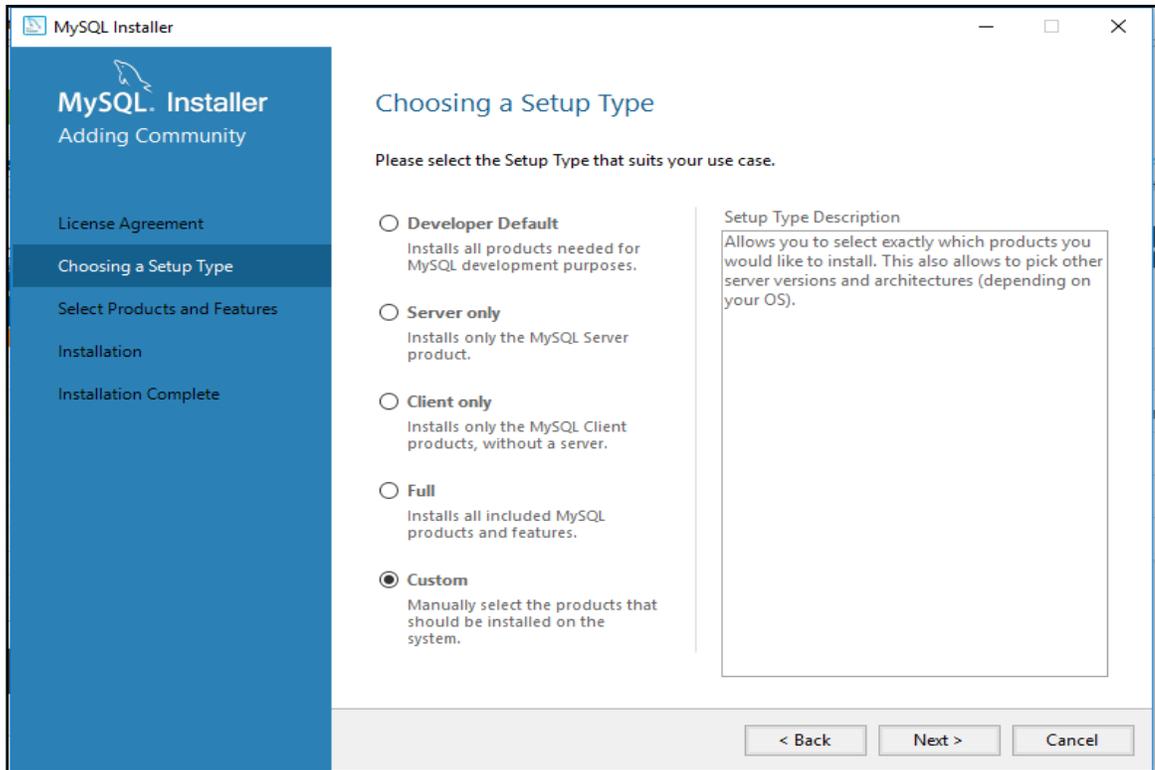
Paso 3

- Lea los términos y condiciones* de la licencia de Oracle. Para poder proseguir con la instalación debe aceptar dichos términos*.



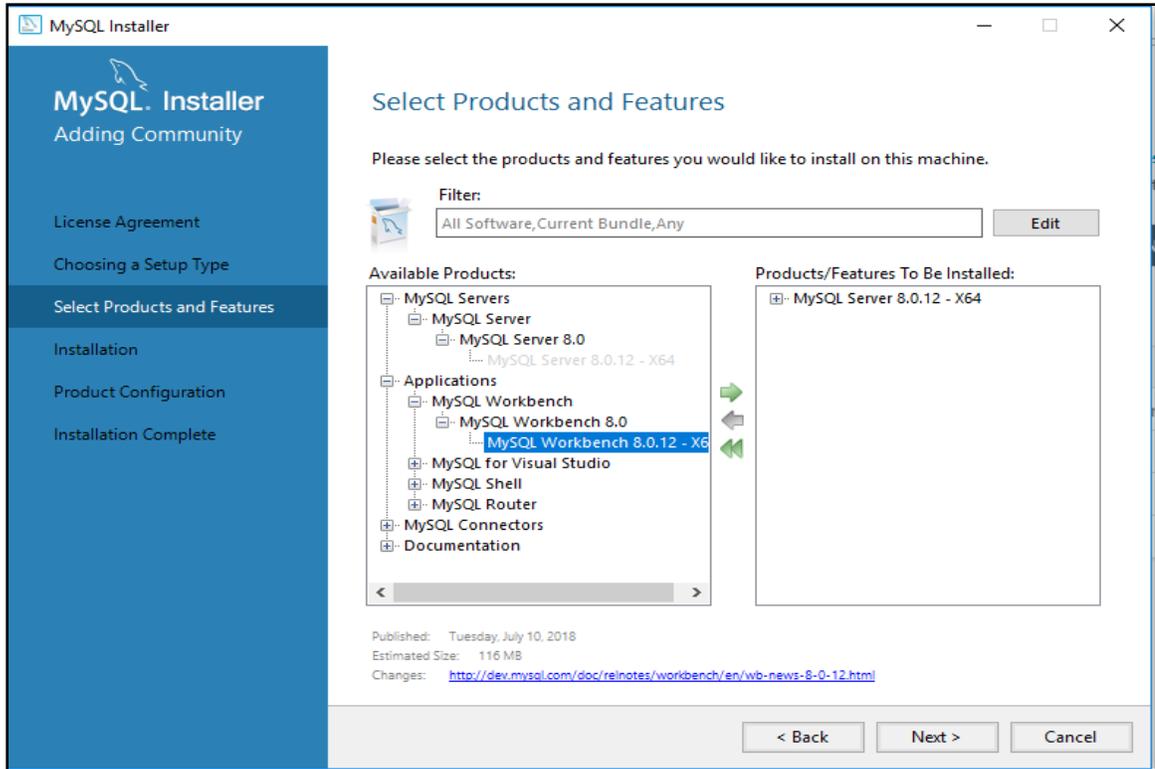
Paso 4

- Seleccione el modo de configuración “customizado” (*custom*) que le permitirá seleccionar únicamente los productos necesarios por instalar en el sistema para poder conectar la base de datos exitosamente con Ignition.



Paso 5

- Ahora seleccione únicamente MySQL Server (la versión depende del instalador que haya ejecutado) y MySQL Workbench.

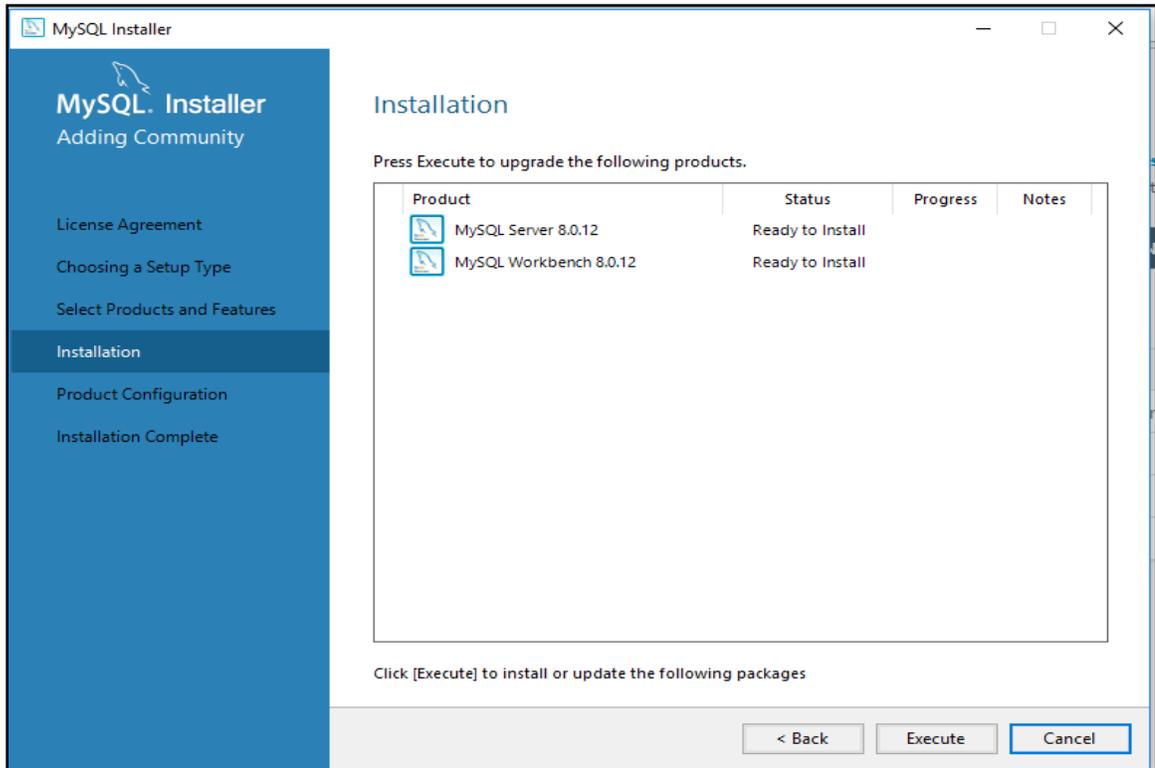


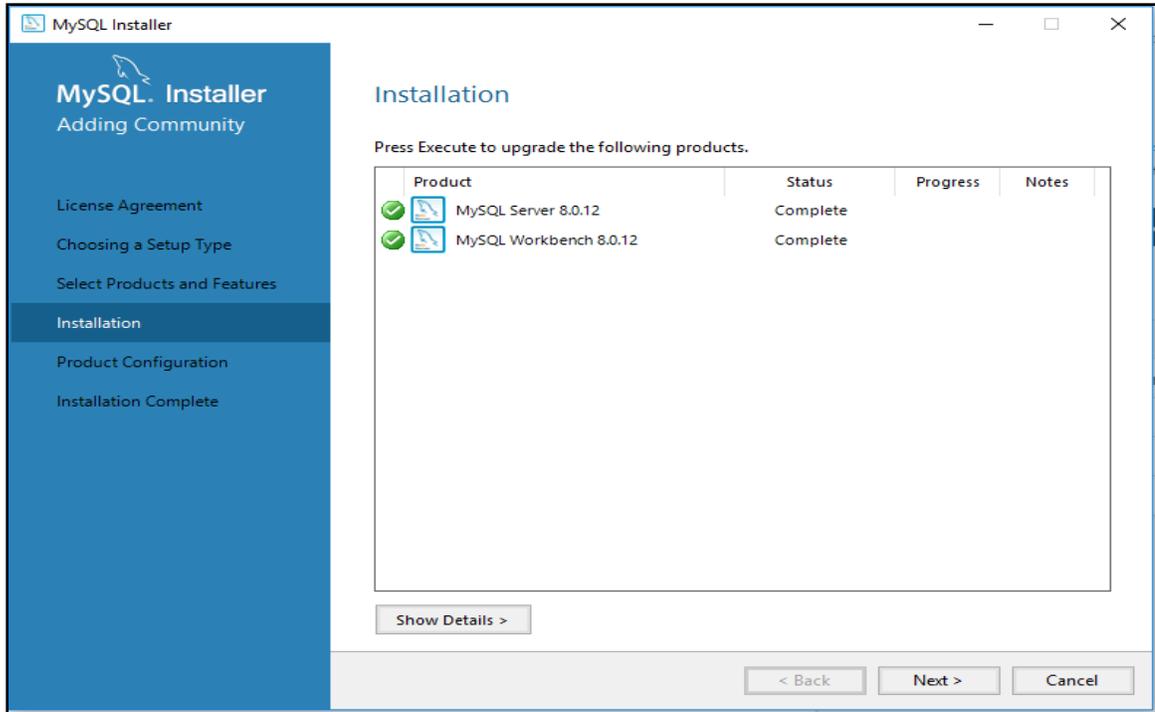
De este modo únicamente debería tener los siguientes productos seleccionados para la instalación:



Paso 6

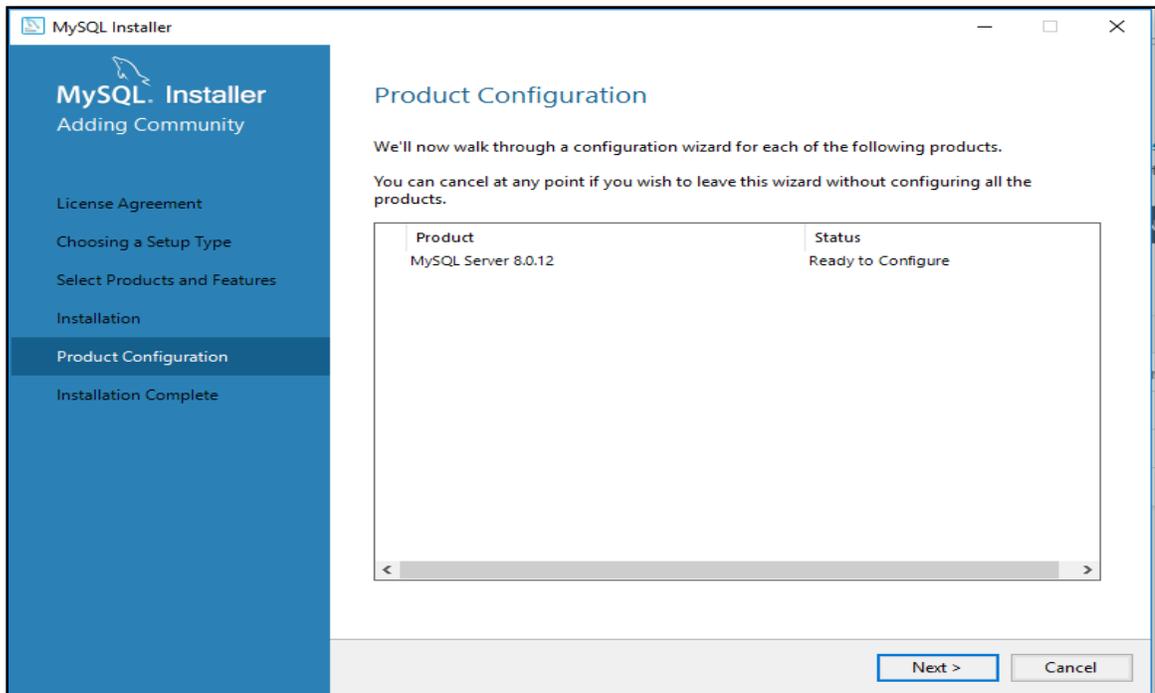
- Inicie la instalación de los productos seleccionados, para ello presione ejecutar (*Execute*).
- La instalación tardará poco tiempo. Al finalizar, el estado de ambos productos deberá indicar que el proceso se completo sin problemas y le permitirá continuar con la configuración de los productos. Haga *click* en *Next >*.

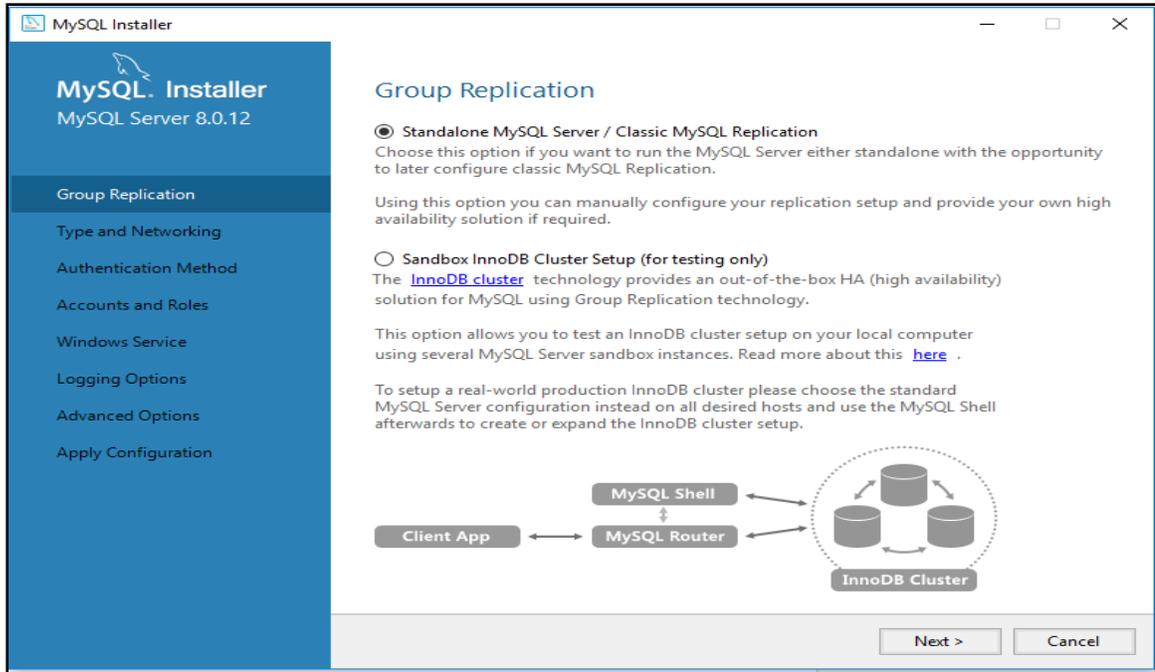




Paso 7

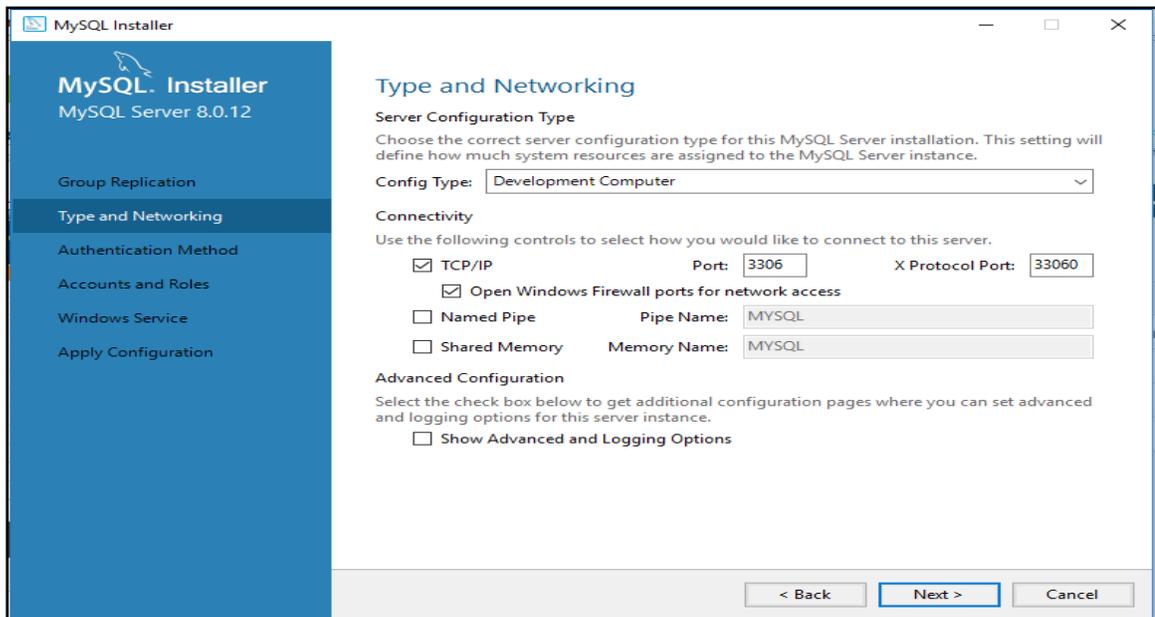
- Para la configuración del Servidor de MySQL seleccione la opción *Standalone MySQL/Classic MySQL Replication* y presione *Next>*.





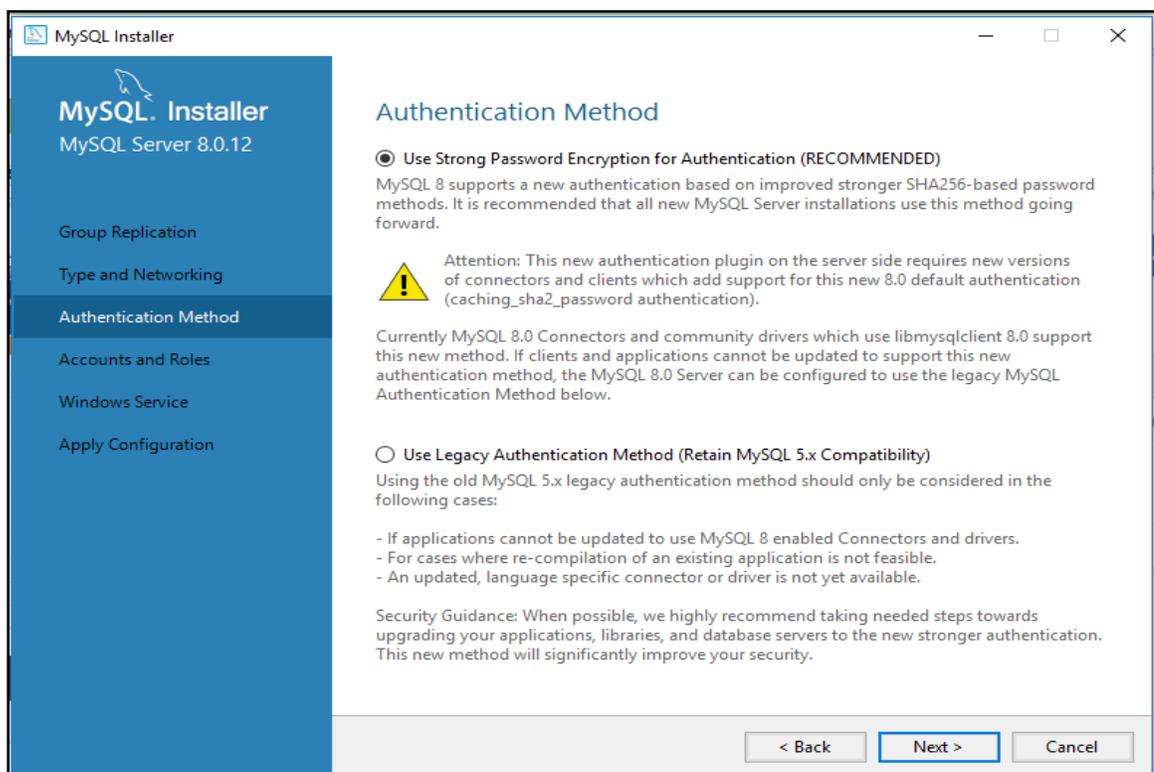
Paso 8

- Es importante que si el servidor de la base de datos se va a instalar en una computadora para desarrollo se seleccione el tipo de configuración correcta, de este modo se definirá cuantos recursos del sistema pueden asignarse al servidor.
- Para la configuración de los parámetros de conectividad seleccione el protocolo TCP/IP y el puerto 3306. De igual manera, marque la casilla “Open Windows Firewall ports for network access”. Haga *click* sobre *Next*>.



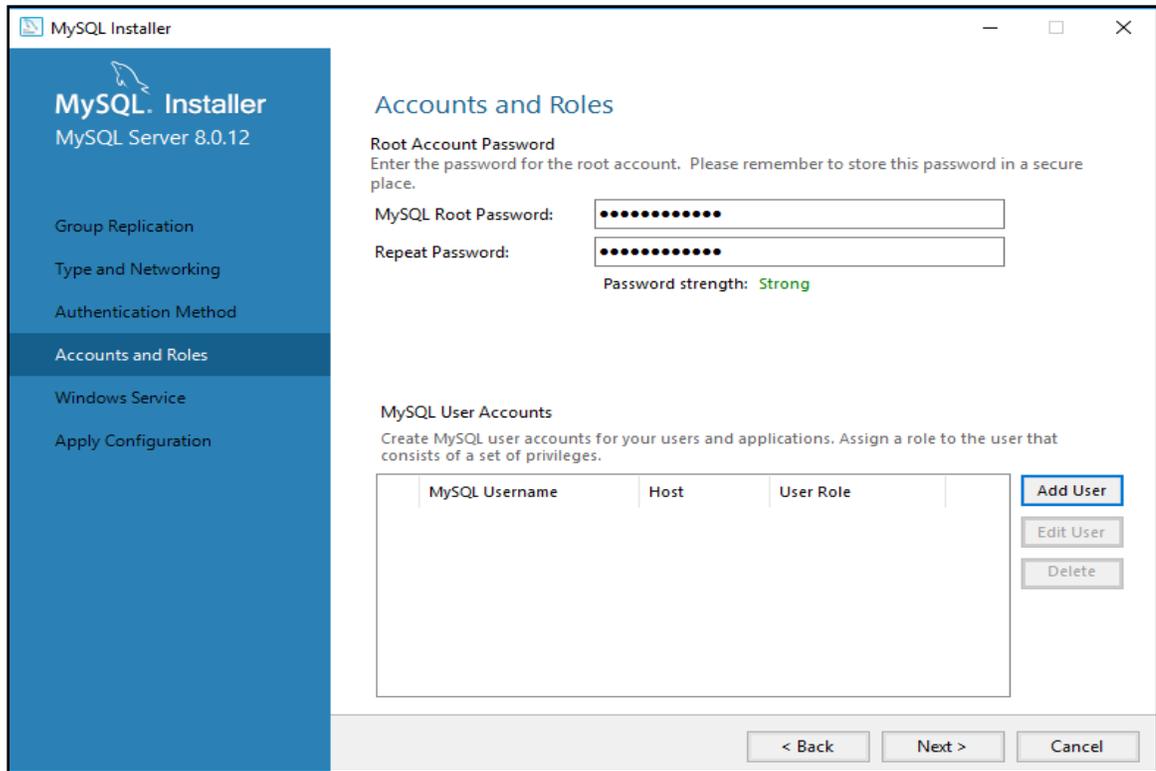
Paso 9

- Seleccione el método recomendado para la autenticación (*Strong Password Encryption for Authentication*) y presione *Next*>. Este nuevo método para versiones posteriores a MySQL 8.0 mejora considerablemente la seguridad de la base de datos. Se debe aclarar que este nuevo complemento de autenticación del lado del servidor requiere nuevas versiones de conectores y clientes que soportan este método (*caching_sha2_password* authentication). Por ello si una aplicación o cliente no puede actualizarse para utilizar conectores y drivers habilitados para MySQL 8.0 se debe seleccionar el método antiguo (*Legacy Authentication Method*). No obstante Ignition sí soporta actualizaciones en los drivers de conexión con MySQL 8.0 y posteriores.



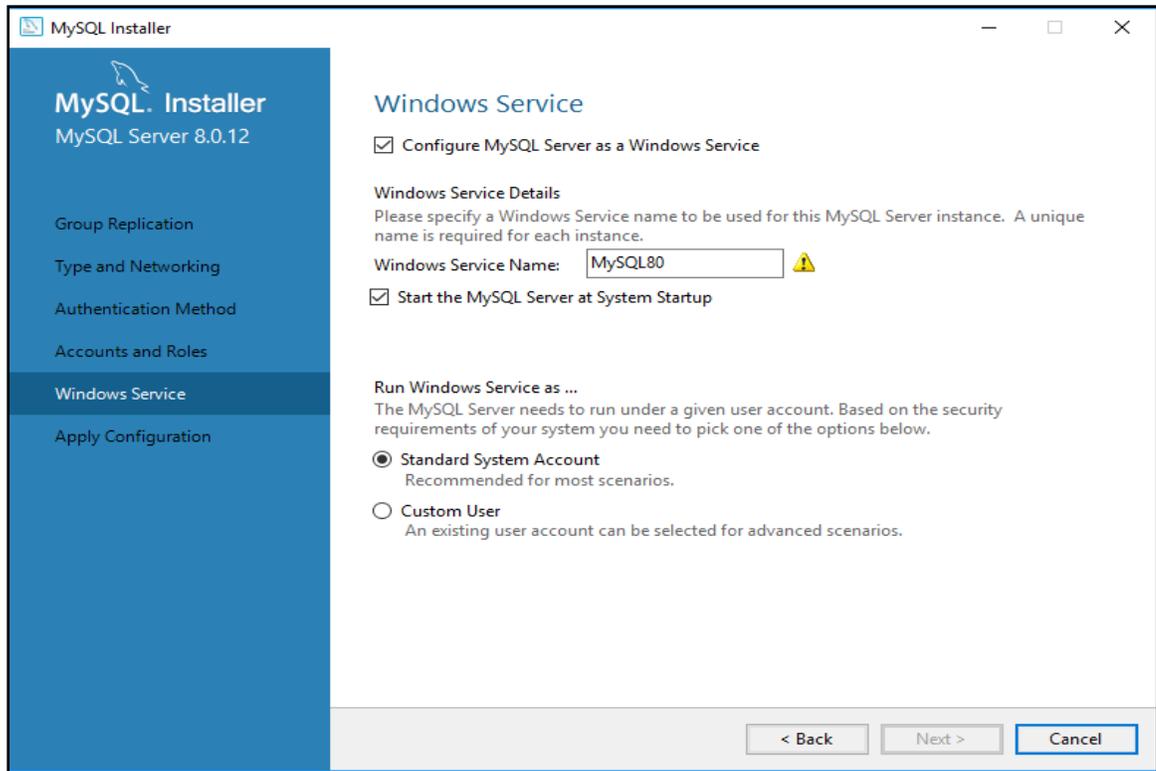
Paso 10

- Ingrese una contraseña segura para la cuenta raíz de la base de datos, puede utilizar números, mayúsculas y caracteres especiales (#_&%!...). Recuerde guardar esta contraseña en un lugar seguro (la utilizará varias veces más adelante). Puede crear nuevas cuentas de usuarios y agregarles roles distintos con una serie de privilegios o permisos a cada uno. Por cuestiones de simplicidad en este manual solo se configura la cuenta raíz.



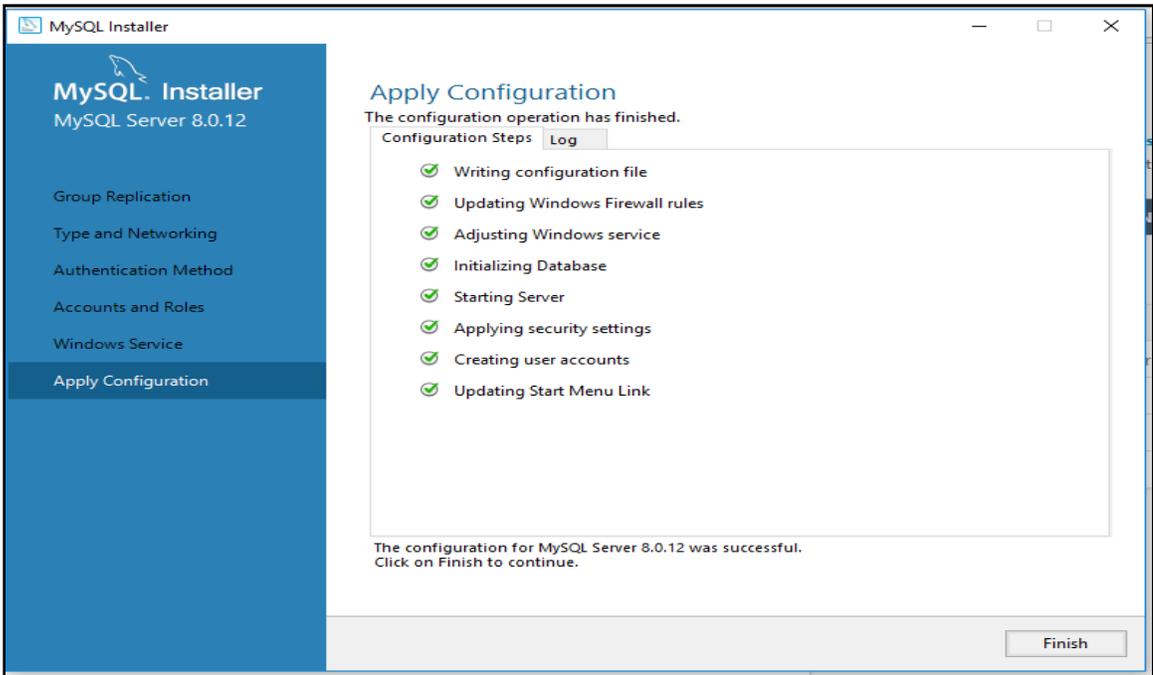
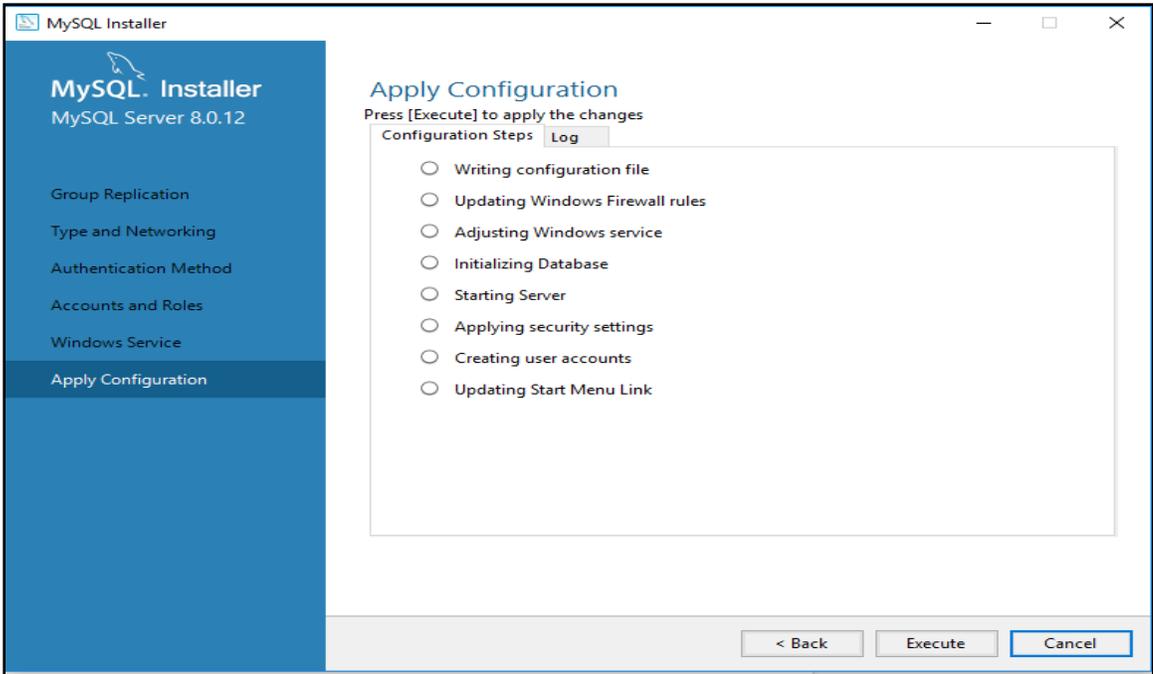
Paso 11

- Realice la configuración del servidor MySQL como un servicio de Windows (programa que opera en un segundo plano dentro del ordenador), y asígnele un nombre de servicio no utilizado (único). Marque la casilla *Start the MySQL Server at System Startup*, de modo que el servicio comience cuando se inicia el sistema operativo y se ejecuta en un segundo plano. Por último ejecute el servicio como una cuenta de sistema estándar (Standard System Account) tal como lo recomienda el instalador.



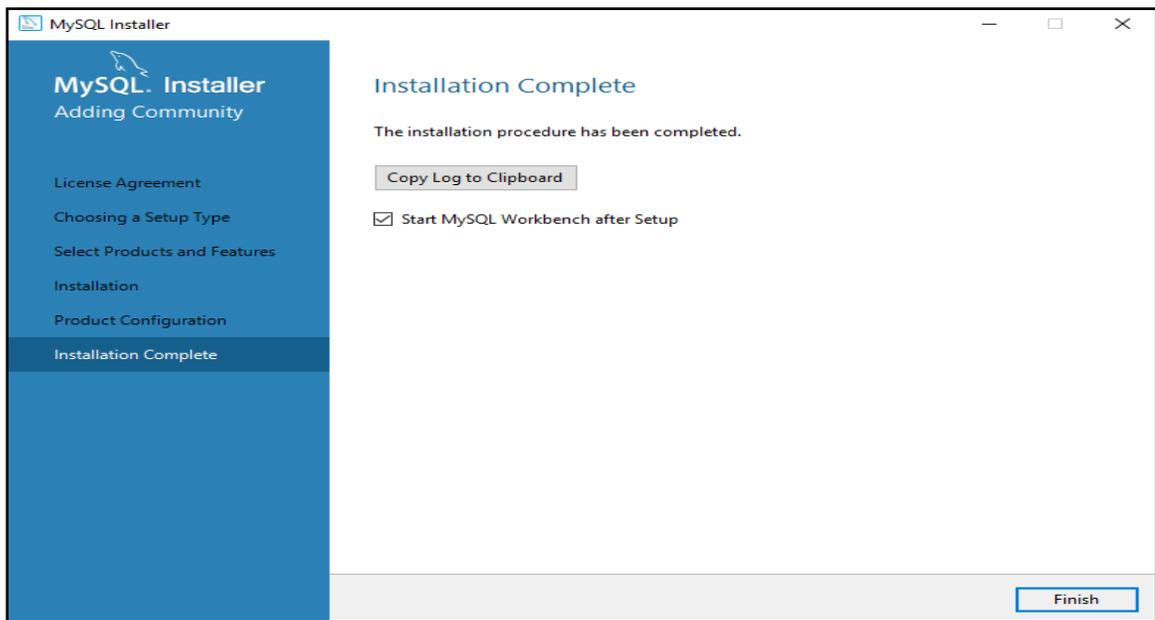
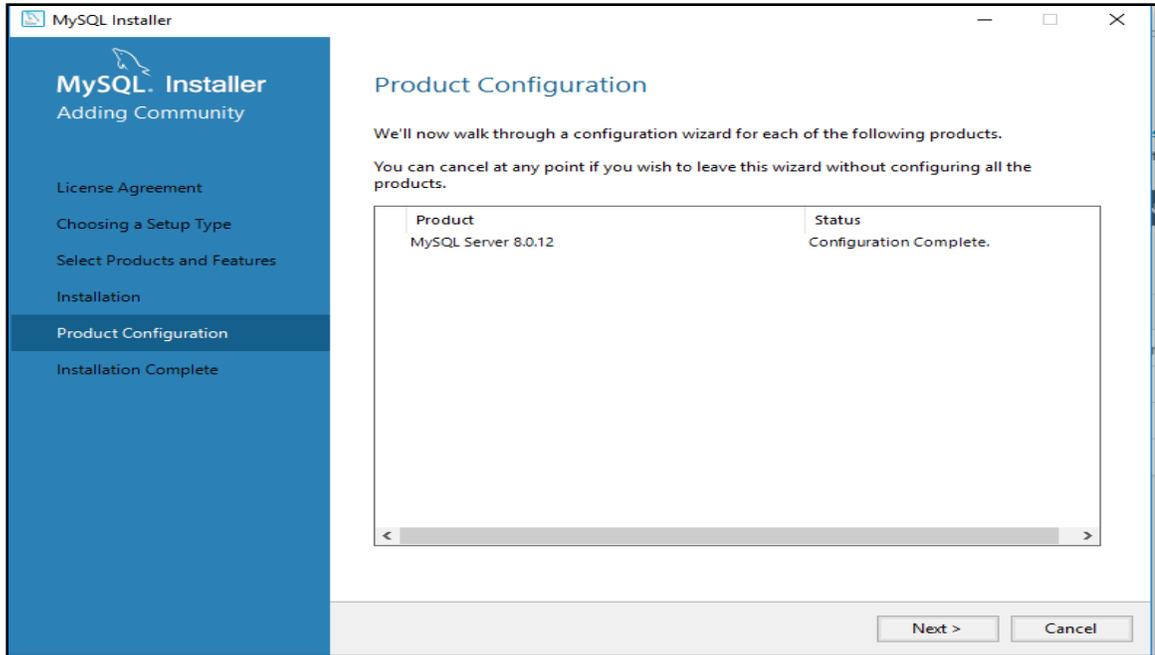
Paso 12

- Haga *click* en ejecutar (*execute*) para aplicar la configuración. Espere a que cada paso se complete de manera exitosa (check verde) y aparezca un mensaje en la parte inferior de la ventana indicando que la configuración del servidor MySQL fue exitosa. Presione finalizar para continuar (*Finish*).



Paso 13

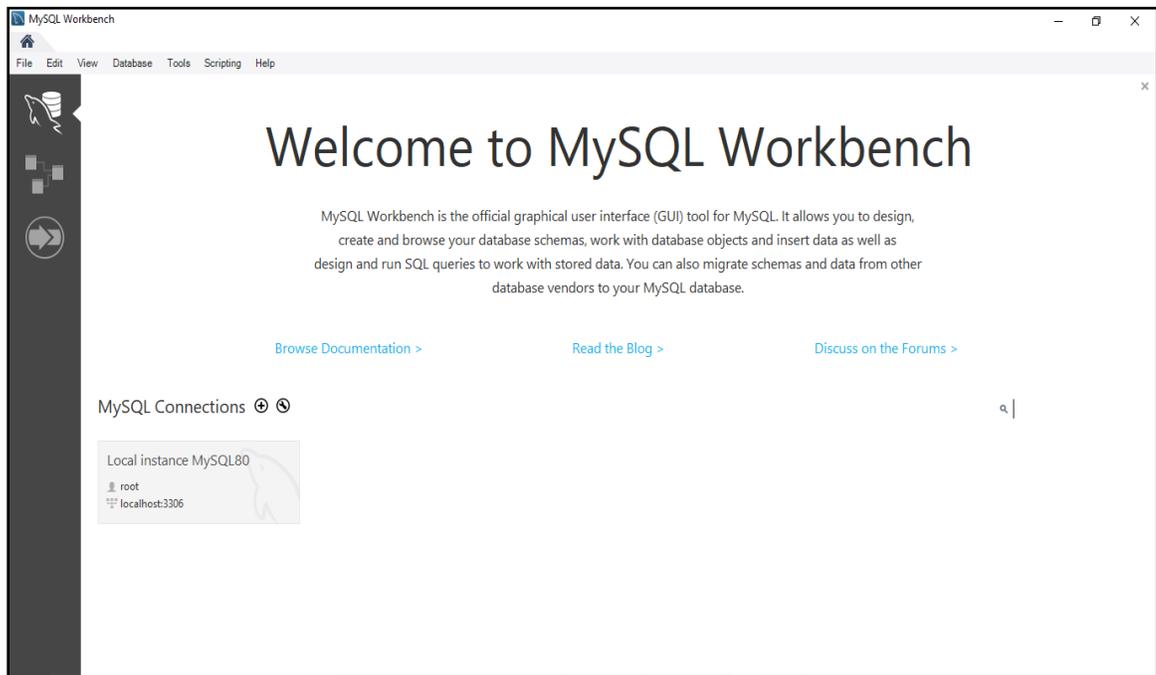
- ¡Estamos por finalizar la instalación! Haga *click* en continuar (*Next>*) y por último seleccione finalizar (*Finish*).



Paso 14

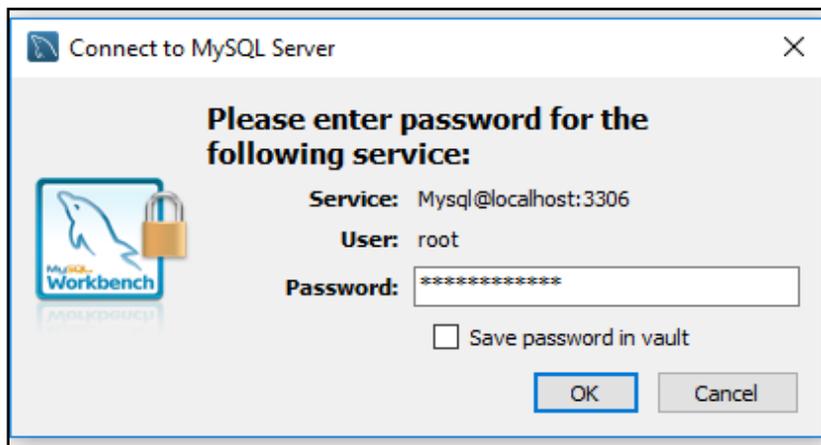
- Al finalizarse la instalación debería abrirse automáticamente el *Workbench* de MySQL (interfaz gráfica para poder interactuar con las bases de datos). En el lado inferior izquierdo de la ventana verifique que haya una conexión de instancia local

MySQL80 (dependiendo de la versión instalada), y que el usuario sea la cuenta raíz (*root*) utilizando el localhost: 3306 (puerto asignado previamente durante la instalación). Haga doble *click* en esta conexión.



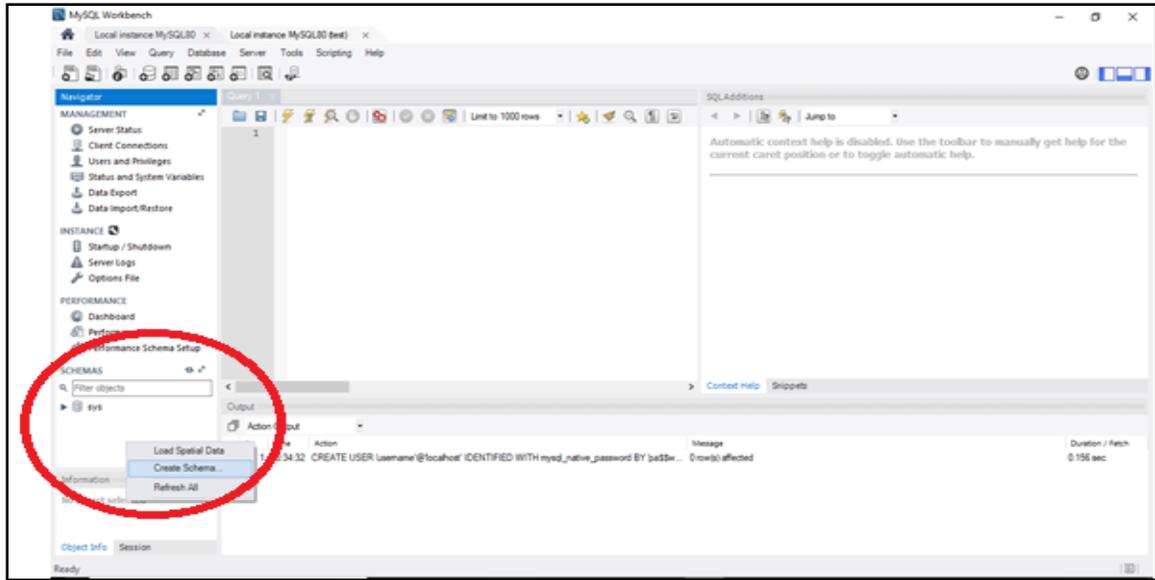
Paso 15

- Ingrese la contraseña para la cuenta raíz, definida en el [paso 10](#) de este manual de usuario.



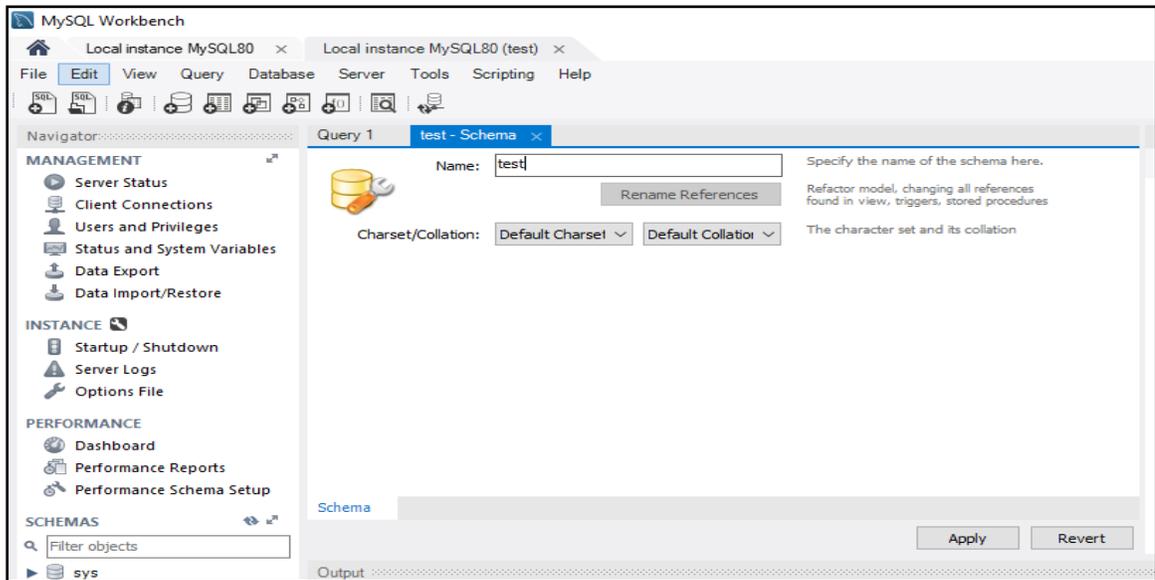
Paso 16

- Haga click derecho en la sección de esquemas de bases de datos (esquina inferior izquierda de la ventana) y seleccione la opción *Create Schema...* para crear la base de datos que posteriormente se vinculará con Ignition.



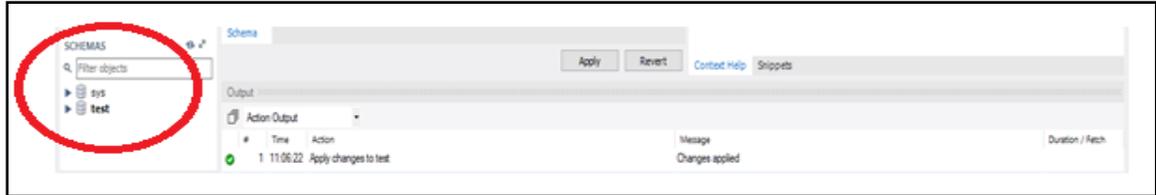
Paso 17

- Asígnale un nombre nuevo al esquema (para el caso de este proyecto se escogió el nombre “test” y en otros documentos generados para este proyecto de graduación que hagan referencia a la base de datos en MySQL hacen referencia a esta instancia con el nombre “test”).



Paso 18

- Verifique que se haya creado el nuevo esquema exitosamente.
- ¡Hemos terminado el proceso de instalación de la base de datos MySQL satisfactoriamente!



Manual Usuario para la conexión de Base de Datos MySQL con plataforma de Ignition a través del Gateway.

Este manual pretende servir como guía para la conexión de Ignition con una base de datos MySQL. Nótese por lo tanto que se requiere haber instalado alguna base de datos MySQL, al igual que contar con alguna versión de Ignition instalada en el servidor central (si todavía no ha instalado algunas de estas en los siguientes hipervínculos encontrará el manual de usuario para la instalación de [MySQL](#) y el manual de usuario para la instalación de [Ignition](#)).

En caso de haber instalado o actualizado la base de datos MySQL a una versión 8.0 o posterior, la conexión con Ignition fallará inicialmente y el *gateway* indicará los siguientes errores:

```
org.apache.commons.dbcp.SQLNestedException: Cannot create
PoolableConnectionFactory (Unable to load authentication plugin
'caching_sha2_password'.)
org.apache.commons.dbcp.SQLNestedException: Cannot create
PoolableConnectionFactory (java.lang.ClassCastException:
java.math.BigInteger cannot be cast to java.lang.Long)
```

Esto se debe a que hubo una serie de cambios que se introdujeron en MySQL 8.0 que provocaron que el driver JDBC que está incluido por default en Ignition no sea compatible. Por ello estas nuevas versiones de bases de datos requieren un driver JDBC actualizado por el desarrollador. En este caso en particular, el problema se resuelve sencillamente actualizando el driver JDBC de MySQL a la última versión publicada por Oracle.

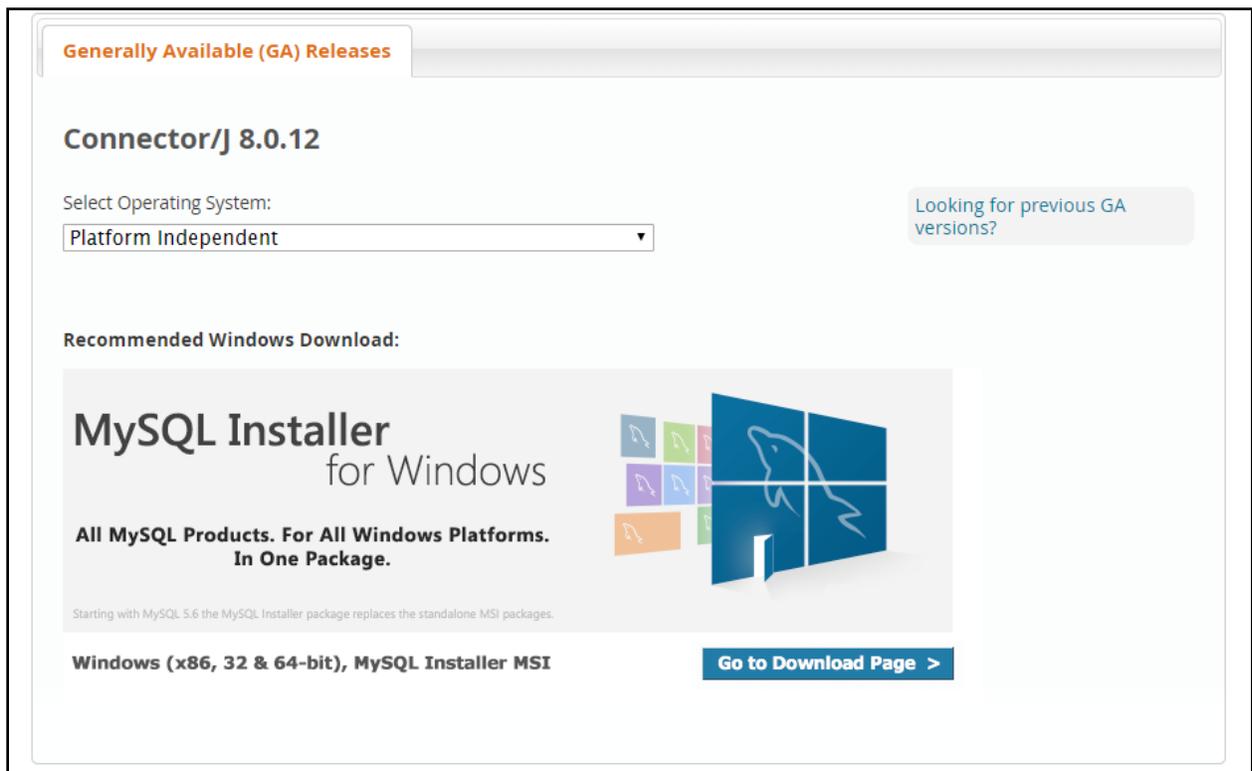
En caso de contar con una versión de MySQL 8.0 o posterior siga detalladamente la totalidad de los siguientes pasos; en caso contrario (para versiones anteriores) puede dirigirse al [paso 6](#) directamente para comenzar con la configuración de la conexión.

Paso 1

- Diríjase a <https://dev.mysql.com/downloads/connector/j/> donde podrá descargar la última versión disponible del driver JDBC para MySQL (la versión 8.0.12 al momento de escribir este manual).

Paso 2

- Seleccione la opción *Platform Independent* (Plataforma Independiente) de la lista desplegable para la selección del sistema operativo.



Generally Available (GA) Releases

Connector/J 8.0.12

Select Operating System:
Platform Independent ▼

[Looking for previous GA versions?](#)

Recommended Windows Download:

MySQL Installer
for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the standalone MSI packages.

Windows (x86, 32 & 64-bit), MySQL Installer MSI

[Go to Download Page >](#)

Generally Available (GA) Releases

Connector/J 8.0.12

Select Operating System:

Looking for previous GA versions?

Platform Independent (Architecture Independent), Compressed TAR Archive (mysql-connector-java-8.0.12.tar.gz)	8.0.12	4.8M	Download
Platform Independent (Architecture Independent), ZIP Archive (mysql-connector-java-8.0.12.zip)	8.0.12	5.5M	Download

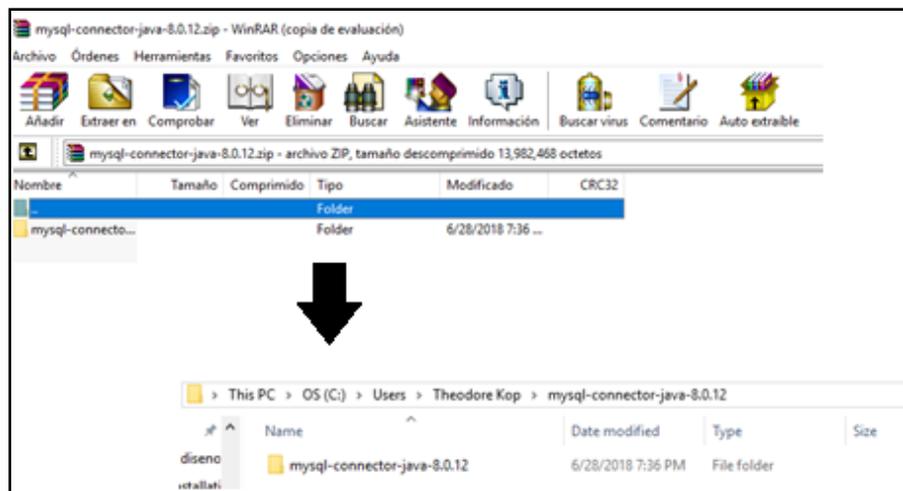
MD5: 368c686aec816ca7d2ad9c98ac0e739f | [Signature](#)

MD5: d7f2ef57f603d245dc7b86db2941ccd6 | [Signature](#)

 We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

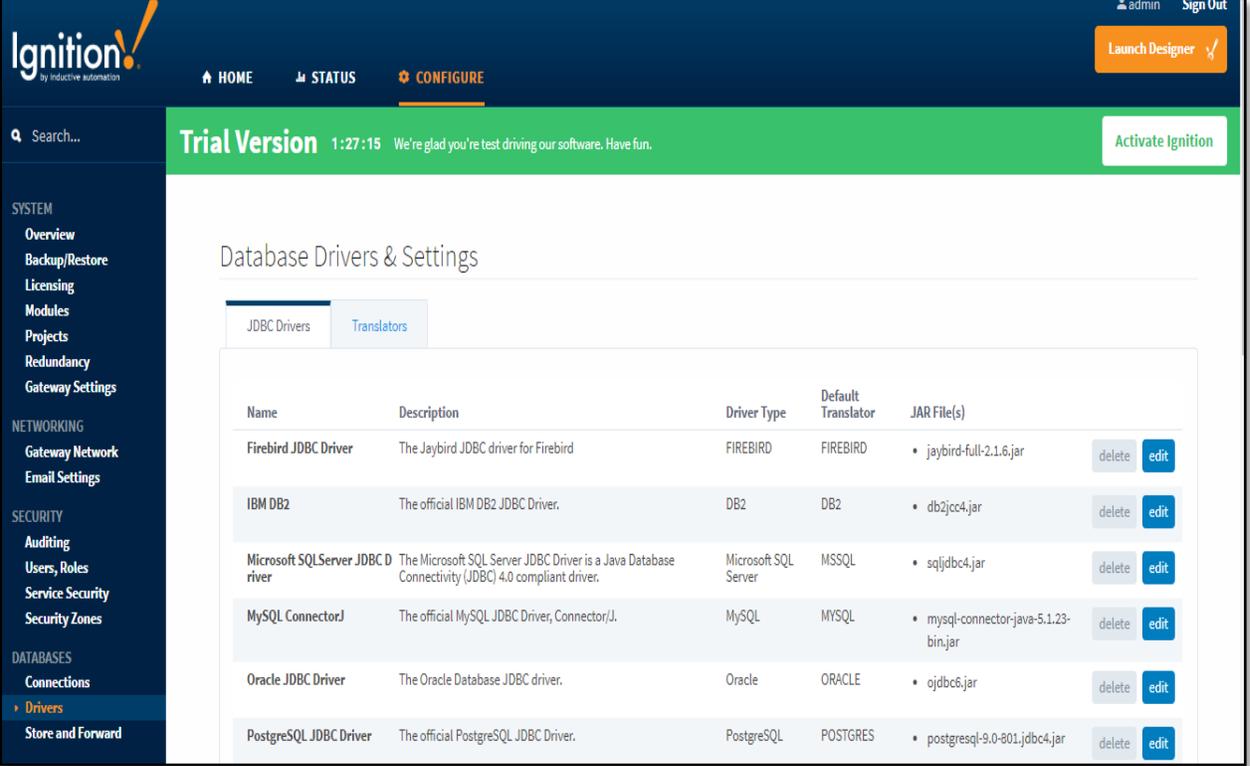
Paso 3

- Descargue el archivo ZIP. Una vez se haya descargado, extraiga los contenidos del archivo y guárdelos en algún directorio que sea fácilmente accesible. Este archivo se utilizará más adelante.



Paso 4

- Acceda al Gateway de Ignition. Diríjase a *Configure>Databases>Drivers*. A continuación verá una lista con todos los drivers y conectores que vienen incluidos en Ignition para facilitar la conexión con las bases de datos más populares. Seleccione *edit* (editar) para MySQL ConnectorJ.



The screenshot shows the Ignition Gateway configuration interface. The top navigation bar includes 'HOME', 'STATUS', and 'CONFIGURE'. The left sidebar lists various system and networking settings. The main content area is titled 'Database Drivers & Settings' and contains a table of installed JDBC drivers. The table has the following data:

Name	Description	Driver Type	Default Translator	JAR File(s)	delete	edit
Firebird JDBC Driver	The Jaybird JDBC driver for Firebird	FIREBIRD	FIREBIRD	• jaybird-full-2.1.6.jar	delete	edit
IBM DB2	The official IBM DB2 JDBC Driver.	DB2	DB2	• db2jcc4.jar	delete	edit
Microsoft SQL Server JDBC Driver	The Microsoft SQL Server JDBC Driver is a Java Database Connectivity (JDBC) 4.0 compliant driver.	Microsoft SQL Server	MSSQL	• sqljdbc4.jar	delete	edit
MySQL ConnectorJ	The official MySQL JDBC Driver, Connector/J.	MySQL	MYSQL	• mysql-connector-java-5.1.23-bin.jar	delete	edit
Oracle JDBC Driver	The Oracle Database JDBC driver.	Oracle	ORACLE	• ojdbc6.jar	delete	edit
PostgreSQL JDBC Driver	The official PostgreSQL JDBC Driver.	PostgreSQL	POSTGRES	• postgresql-9.0-801.jdbc4.jar	delete	edit

Paso 5

En la ventana de edición, realice los siguientes cambios:

- Donde dice *Classname*, cambie el valor **com.mysql.jdbc.Driver** a **com.mysql.cj.jdbc.Driver**
- En la sección *JAR File(s)* haga *click* en *Choose File* y navegue a la dirección en donde guardo los archivos exportados del ZIP previamente descargado en el [paso 2](#). Dentro de este, busque el archivo .jar y selecciónelo para cargar al *Gateway* de Ignition.
- En la sección *Default Connection Properties* introduzca el siguiente valor:

```
zeroDateTimeBehavior=CONVERT_TO_NULL;connectTimeout=120000;socketTimeout=120000;useSSL=false;allowPublicKeyRetrieval=true;
```

- Por último haga *click* en *Save Changes*. De este modo se ha editado satisfactoriamente el driver JDBC para MySQL y cualquier nueva conexión que se desee realizar con una base de datos MySQL (versión 8.0 en adelante) no presentará ningún conflicto.

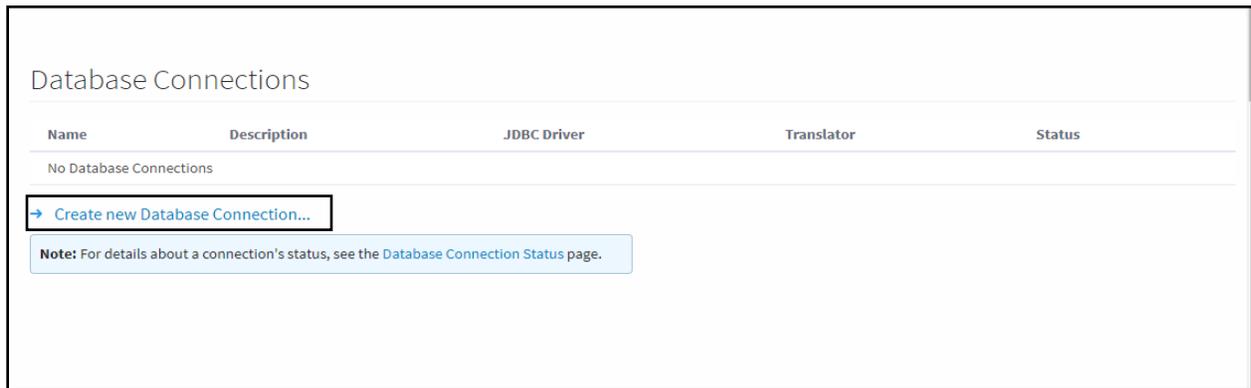
Edit JDBC Driver	
Main Properties	
Name	MySQL ConnectorJ
Description	The official MySQL JDBC Driver, Connector/J.
Classname	com.mysql.cj.jdbc.Driver Enter the classname that is the JDBC driver in the Jar File that you are uploading. (The classname should be easily found in the documentation for the JDBC driver that you are uploading.)
JAR File(s)	<input type="button" value="Choose File"/> No file chosen Files: Choose new *.jar to replace the existing ones, or leave blank to keep existing *.jar files.

Driver Defaults & Instructions	
Driver Type	MySQL Choose the appropriate database type that this driver connects to.
URL Format	jdbc:mysql://localhost:3306/test Provide a default value for the connect URL. This should provide a hint while adding a datasource connection as to the format of the connect URL that this driver requires. For example: "jdbc:dbtype://host:port/database"
URL Instructions	 The format of the MySQL connect URL is: <code>jdbc:mysql://host:port/database</code>
Default Connection Properties	zeroDateTimeBehavior=CONVERT_TO_NULL;connectTimeout=120000;socketTimeout=120000;useSSL=false;allowPublicKeyRetrieval=true; (default:)
Connection Properties Instructions	There is an extensive list of extra connection properties available for MySQL Connector/J. See
Default Validation Query	SELECT 1 (default: SELECT 1)

SQL Language Compatibility	
Default Translator	MYSQL
<input type="button" value="Save Changes"/>	

Paso 6

- Diríjase al *Gateway* de Ignition. (Para cualquier equipo dentro de la misma red que el Servidor puede ingresar al *Gateway* a través de cualquier navegador introduciendo la dirección <http://localhost:8088>)
- De click en la pestaña de Configuración que se encuentra en la esquina superior izquierda.
- Vaya a la sección de *Databases* al lado izquierdo de la ventana y de click en *Connections*.
- De click en *Create new Database Connection...* para crear una nueva conexión con una base de datos.



Paso 7

- Dado que la conexión es con una base de datos MySQL, se debe seleccionar el Driver JDBC correspondiente a la base de datos que se desea conectar. Ignition incluye una serie de drivers previamente configurados para las bases de datos más utilizadas, sin embargo en caso de que se utilice alguna otra no incluida, se puede añadir manualmente un nuevo driver que permita la conexión con esta. De igual modo, y como previamente señalado en caso de utilizar una versión de MySQL posterior a la 8.0 se debe editar el Driver antes de poder continuar (si este es el caso y aún no lo ha hecho por favor diríjase al [Paso 1](#)).
- Seleccione MySQL ConnectorJ que es el driver oficial para bases de datos MySQL y haga *click* en *Next*>.

Add Connection Step 1: Choose Driver

Select the correct JDBC Driver for the type of database you wish to connect to. If no driver corresponds to your database, go to the Driver Configuration page to add a new driver.

- Firebird JDBC Driver**
The Jaybird JDBC driver for Firebird
- IBM DB2**
The official IBM DB2 JDBC Driver.
- Microsoft SQLServer JDBC Driver**
The Microsoft SQL Server JDBC Driver is a Java Database Connectivity (JDBC) 4.0 compliant driver.
- MySQL ConnectorJ**
The official MySQL JDBC Driver, Connector/J.
- Oracle JDBC Driver**
The Oracle Database JDBC driver.
- PostgreSQL JDBC Driver**
The official PostgreSQL JDBC Driver.

Paso 8

- Elija un nombre para la conexión con la base de datos (preferiblemente algo fácil de identificar, en especial si se tienen más conexiones a otras bases de datos). En este manual se seleccionó el nombre MySQL.
- Para la sección *Connect URL* se debe introducir la dirección en donde la base de datos se encuentra “corriendo”. Se debe utilizar el formato: `jdbc:mysql://host:port/database`. En donde el *host* es la dirección IP del servidor de la base de datos, *port* es el puerto en el que el servidor está “corriendo” (por default es el 3306) y *database* es el nombre de la base de datos en el servidor MySQL a la que se está intentando conectar. En el caso de particular de este proyecto se utiliza entonces: `jdbc:mysql://localhost:3306/test`.
- En la sección de Usuario (*User*): digite “root” (usuario raíz del servidor MySQL) y en la sección de Contraseña (*Password*): digite la contraseña previamente asignada a la cuenta raíz durante la instalación de MySQL Server. Las demás secciones se deben dejar con los valores por default.
- Al finalizar, haga *click* en *Create new Database Connection*.

New Database Connection

Main Properties	
Name	<input type="text" value="MySQL"/> <p>Choose a name for this database connection.</p>
Description	<input type="text"/>
JDBC Driver	<input type="text" value="MySQL ConnectorJ"/> <p>The JDBC driver dictates the type of database that this connection can connect to. It cannot be changed once created.</p>
Connect URL	<input type="text" value="jdbc:mysql://localhost:3306/test"/> <p>The Connect URL is JDBC-driver specific. It usually contains the address of the machine that the database is running on. The format of the MySQL connect URL is: jdbc:mysql://host:port/database With the three parameters (in bold) host: The host name or IP address of the database server. port: The port that the database server is running on. MySQL default port is 3306. database: The name of the logical database that you are connecting to on the MySQL server.</p>

Username	<input type="text" value="root"/>
Password	<input type="password" value="*****"/>
Password	<input type="password" value="*****"/> <p>Re-type password for verification.</p>

Slow Query Log Threshold	<input type="text" value="60000"/> <p>Queries that take longer than this amount of time, in milliseconds, will be logged. This helps to find queries that are not performing well. (default: 60,000)</p>
Validation Timeout	<input type="text" value="10000"/> <p>The time in milliseconds between database validation checks. (default: 10,000)</p>
<input type="checkbox"/> Show advanced properties	
<input type="button" value="Create New Database Connection"/>	

Paso 9

- Verifique que el estado de la nueva conexión con la base de datos indica que la conexión es válida.
- ¡Hemos terminado el proceso de conexión de Ignition a una base de datos MySQL satisfactoriamente!

Database Connections

Name	Description	JDBC Driver	Translator	Status	
MySQL		MySQL ConnectorJ	MYSQL	Valid	delete edit

[→ Create new Database Connection...](#)

Note: For details about a connection's status, see the [Database Connection Status](#) page.