CÁLCULO DE OEE EN UNA CELDA AUTOMATIZADA CON ROBOTS COLABORATIVOS

TESIS

QUE PARA OBTENER EL GRADO DE

MAESTRO EN SISTEMAS COMPUTACIONES

PRESENTA

HUMBERTO HILARIO PEDRAZA ACOSTA

DIRECTOR

M.C. OSCAR INDALESIO RAMÍREZ HUERTA

H. NOGALES, SONORA, MÉXICO. **JUNIO DE 2021.**

Instituto Tecnológico de Nogales
Departamento de Comunicación y Difusión

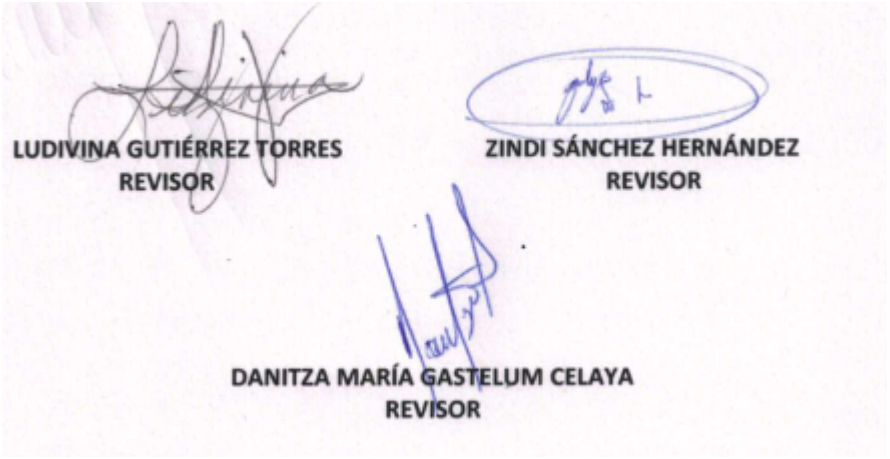
H. Nogales Son., 15/junio/2021
Oficio No. DEPI/089/2021.

REYNALDO GUTIÉRREZ GUTIÉRREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
PRESENTE.

Por este medio le comunicamos a Usted que el trabajo de Tesis denominado: ***"CÁLCULO DE OEE EN CELDA AUTOMATIZADA CON ROBOTS COLABORATIVOS."****, que presentó el alumno* ***HUMBERTO HILARIO PEDRAZA ACOSTA****, con número de control 17341005, candidato a obtener el grado de* ***Maestro en Sistemas Computacionales****, ha sido revisado por los miembros del Comité Tutorial y cubiertas las observaciones realizadas, se Autoriza su Impresión y se Acepta para su Evaluación en la presentación del Examen de Grado.*

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.

A T E N T A M E N T E
Excelencia en Educación Tecnológica•
"La Ciencia y la Tecnología para la Liberación del Hombre"•



Av. Instituto Tecnológico # 911 Col. Granja. C.P. 84065, Nogales, Sonora
Tels. (631) 314-8436, Conmutador: (631) 159-0001
e-mail: dir_nogales@tecnm.mx
www.tecnm.mx | www.nogales.tecnm.mx

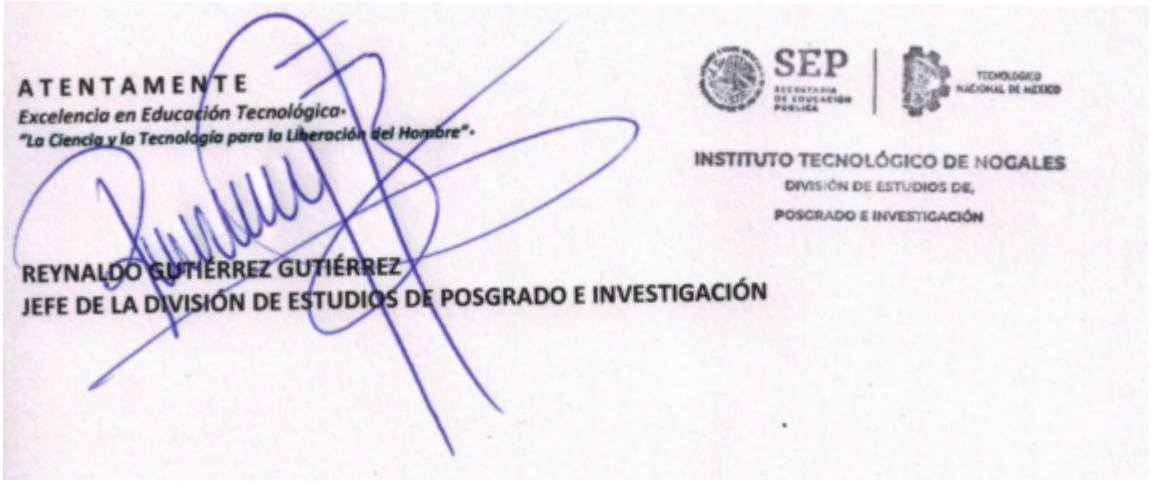
Instituto Tecnológico de Nogales
Departamento de Comunicación y Difusión

H. Nogales Sonora, 15/junio/2021
Oficio No. DEPI/090/2021.

HUMBERTO HILARIO PEDRAZA ACOSTA
CANDIDATO A OBTENER EL GRADO DE MAESTRO EN SISTEMAS COMPUTACIONALES
PRESENTE.

*De acuerdo con el Reglamento de Titulación del Sistema Nacional de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que el Comité Tutorial realizó, con respecto a su Tesis titulada: **"CÁLCULO DE OEE EN CELDA AUTOMATIZADA CON ROBOTS COLABORATIVOS."**, la División de Estudios de Posgrado e Investigación Autoriza su Impresión.*

Agradeciendo de antemano el apoyo brindado al presente, le reitero mi consideración distinguida.



ccp. Archivo

RGG/cmmj

Av. Instituto Tecnológico # 911 Col. Granja. C.P. 84065, Nogales, Sonora
Tels. (631) 314-8436, Conmutador: (631) 159-0001
e-mail: dir_nogales@tecnm.mx
www.tecnm.mx | www.nogales.tecnm.mx

RESUMEN

En un futuro próximo las empresas de manufactura estarán llevando a cabo la implementación de robots para sus procesos. Lo cual provocará que la mano de obra humana sea desplazada con el fin de tener mayor precisión, eficacia y trabajo continuo.

Sin embargo, aún estamos en el inicio de estas implementaciones. Es de hacer notar que los primeros en ser utilizados en la industria de la manufactura son los robots colaborativos. Ya que estos sirven de apoyo a los seres humanos en procesos repetitivos.

Uno de los objetivos principales en las empresas de manufactura es ser más eficientes, en el entorno de producción. Por lo tanto hoy día es indispensable para la industria conocer la Eficiencia General de los Equipos OEE (Overall Equipment Effectiveness), ya que esta permite determinar si sus procesos están sufriendo pérdidas o están en un camino de mejora o simplemente están logrando la excelencia.

En este trabajo, se propone desarrollar una aplicación la cual apoye al cálculo del OEE en tiempo real. Esta herramienta permite visualizar el OEE por turno y por hora para una celda de producción automatizada por robots colaborativos.

Con el fin de lograr este objetivo se realizó el análisis, diseño e implementación de un desarrollo de software, construido a la medida para este propósito. Este realiza los cálculos de forma automática.

Para lo anterior se analizaron y diseñaron los modelos necesarios, y posteriormente se implementaron obteniéndose un prototipo.

Este desarrollo consiste en la creación de una interfaz de usuario elaborada en Angular, un servicio web desarrollado en C# y una base de datos en MySQL.

El prototipo de software construido fue alimentado con los datos de los tipos de equipo, línea de producción y los equipos que la conforman. En la sección de equipos se registraron las horas de trabajo, el tiempo de ciclo, el rendimiento por hora y el total de unidades a producir en el tiempo estipulado de trabajo.

Se corrió una simulación con el prototipo y los resultados obtenidos del OEE muestran que es una gran herramienta que facilita en gran medida la toma de decisiones para las empresas de manufactura, ya que ofrece resultados en tiempo real.

ABSTRACT

In the near future, manufacturing companies will be implementing robots for their processes. This will cause human labor to be displaced in order to have greater precision, efficiency and continuous work.

However, we are still in the beginning of these implementations. It should be noted that the first to be used in the manufacturing industry are collaborative robots. These serve to support human beings in repetitive processes.

One of the main objectives in manufacturing companies is to be more efficient in the production environment. Therefore, today it is essential for the industry to know the general equipment efficiency OEE (Overall Equipment Effectiveness). General equipment efficiency determines if their processes are suffering loss, on a path of improvement, or are simply achieving excellence.

In this work, it is proposed to develop an application which supports the calculation of the OEE in real time. This tool allows visualizing the OEE as shift and by hour for a production cell automated by collaborative robots.

In order to achieve this objective, we custom built the analysis, design and implementation of a software development. This tool performs the calculations automatically.

For the above, the necessary models were analyzed and designed, and later they were implemented, obtaining a prototype.

This development consists of the creation of a user interface elaborated in Angular, a web service developed in C# and a database in MySQL.

The software prototype built is managed with equipment types, necessary tools, and the production line. In the equipment section, work hours, cycle time, hourly performance and total units to be produced in the stipulated work time were recorded.

A simulation prototype was tested. The results obtained from the OEE show that it is a useful tool that greatly facilitates decision-making for manufacturing companies; it offers results in real time.

AGRADECIMIENTOS

Agradezco a mi familia a mi esposa y mis hijas e hijo por darme fuerzas y superar los obstáculos y las dificultades.

Agradezco a mi señor padre por siempre apoyarme en todo en esta vida y brindarme su compresión y estima.

Agradezco a mi señora madre por darme la vida y estar presente en todos mis logros.

Agradezco al MC Oscar I Ramírez, director de tesis por su valiosa guía y asesoramiento a la realización de la misma.

Agradezco al MC Ludivina Gutiérrez, por su apoyo en este trabajo de tesis, así como su paciencia y pasión.

DEDICATORIA

Dedico este trabajo a Dios, por haberme dado la vida y permitirme llegar a donde he llegado hasta este momento tan importante de mi formación profesional. A mi esposa por creer en mí y hacerme más fuerte en cada paso, A mis hijos que son el motor de mis ganas de superarme y me regalan emociones únicas. A mi padre, que a pesar que ya no se encuentra físicamente más en este mundo. A mi madre por darme la vida, valores y amor. A mi maestro Oscar I. Ramírez que significa mucho para mí del lado personal y profesional de la vida quien me ha brindado las mayores herramientas para desarrollarme en esta vida profesional. A mi maestra Ludivina quien me mostro la otra cara de la moneda respecto a proyectos de investigación y me hizo crecer como persona y profesionalmente.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

CAPÍTULO 1. INTRODUCCIÓN

1.1 Antecedentes

A nivel global en la industria actual se cuenta con diferentes tipos de procesos en cuanto a su grado de automatización, es decir; procesos con bajo grado de automatización en los que el desarrollo del proceso es ejecutado en su mayor parte por el factor humano; y otros con alto grado de automatización que en el desarrollo de su ejecución implican una reducida interacción con el factor humano.

Los sistemas automatizados posibilitan la transferencia del trabajo humano a las máquinas. Esto es de utilidad sobre todo en procesos que representan riesgos para la seguridad del operador, por ejemplo, subprocesos que involucren temperaturas elevadas o entornos peligrosos como los que implican manejo de productos químicos, radiactivos o nocivos (Linares González, 2018). Estos sistemas están compuestos por sensores, actuadores, robots, equipos de protección, así como los hardware y software necesarios para la ejecución de las operaciones automáticas. Estos tipos de sistemas reducen costos de producción y especialmente garantizan la seguridad de las personas.

En México, el área de oportunidad para implementar automatizaciones está representada por un 52% de tareas que pueden ser ejecutadas por robots: entre los sectores económicos donde las automatizaciones se implementan de forma vertiginosa se encuentra el automotriz, electrónico, metalúrgico, químico, plástico y de goma; siendo de éstos el sector automotriz el más predominante.

México ha crecido a pasos agigantados en la industria automotriz, por lo cual el país se está colocando en el ranking de potencias industriales a nivel global en la producción de automóviles, ocupando el séptimo lugar en términos de producción; por lo que se espera que para el 2020 ocupe el sexto lugar.

La inversión extranjera directa de alemanes y japoneses en México ha generado más presión en los fabricantes y proveedores de partes automotrices para que mejoren su calidad, aumenten su producción y reduzcan sus costos. Para alcanzar estos estándares se está recurriendo a la automatización y la robótica; siendo esta última la que más ha crecido en el país (Soria, 2016).

1.1.1 Industria 4.0

El término *industria 4.0* se refiere a un nuevo modelo de organización y de control de la cadena de valor a través del ciclo de vida del producto y a lo largo de los sistemas de fabricación apoyado y hecho posible por las tecnologías de la información (Del Val, 2014).

El término industria 4.0 se acuñó en Alemania, pero se utiliza de manera generalizada en Europa. También es habitual referirse a este concepto con términos como “Fábrica Inteligente” o "Internet industrial". En definitiva, se trata de la aplicación a la industria del modelo "Internet de las cosas" (IoT). Todos estos términos tienen en común el reconocimiento de que los procesos de fabricación se encuentran en un proceso de transformación digital, una "cuarta revolución industrial" producida por el avance de las tecnologías de la información y, particularmente, de la informática y el software.

El concepto de Industria 4.0 abarca la digitalización de la cadena de valor horizontal y vertical, la innovación en productos y servicios y la creación de nuevos modelos de negocios dando lugar a una industria más inteligente (Dimitrov, 2018).

Esto hace que se puedan implementar sensores, procesadores, robots y otros dispositivos conectados entre sí (Hermann, Pentek, & Otto, 2015). Gracias a ello, es posible determinar la *Eficiencia General de los Equipos* (*Overall Equipment Effectiveness)* u OEE por sus siglas en inglés.

La adquisición de robots en la industria se debe a que los robots han evolucionado enormemente, brindando una mayor utilidad, ya que son cada vez más autónomos, flexibles y cooperativos. Estas características permiten la fácil integración a la denominada industria 4.0. (Flores, 2017).

Tan sólo en 2015 las ventas de robots en México rompieron todos los registros de años anteriores. Con un total de 6,320 unidades, las ventas sobrepasaron casi tres veces las del año anterior de 2014. La industria de la robótica crece en el país de manera acelerada, colocando a México como una potencia en Industria 4.0 (Flores, 2017).

1.1.2 Eficiencia general de los equipos

El concepto de *Overal Eficiency Equipment* (OEE) nace en 1989 y se utiliza por primera vez por Seiichi Nakajima, el fundador del TPM *(Total Productive Maintenance)* Mantenimiento Productivo Total. Este indicador pretendía identificar las ineficiencias que se originan durante el proceso de fabricación como lo son equipos de ensamble, pruebas y robots (Kenneth Kennedy, 2017).

El OEE representa una métrica muy simple para indicar inmediatamente el estado actual de un proceso de fabricación; pero a la vez es también una herramienta compleja que permite comprender el efecto de los diversos problemas en el proceso de fabricación y cómo afectan.

La implementación correcta del OEE está directamente relacionada al resultado final del proceso de manufactura. Esto se debe a que se reducen los tiempos de inactividad de las máquinas, se establecen las causas por las que hay pérdidas de rendimiento (cuellos de botella y velocidades reducidas), y se incrementa el índice de calidad del producto, reduciendo los retrabajos y los productos defectuosos (Greiner, 2015).

El valor del indicador OEE permite clasificar una o más líneas de producción, o de toda una planta con respecto a las mejores de su clase y que ya han alcanzado el nivel de excelencia (Cruelles Ruiz, 2010).

Tabla 1.1 Tabla de Clasificación del OEE

OEE	Calificativo	Consecuencias
OEE< 65%	Inaceptable	Importantes pérdidas económicas. Baja competitividad.
65%< OEE < 75%	Regular	Pérdidas económicas. Aceptable solo si está en proceso de mejora.
75%< OEE <85%	Aceptable	Ligeras pérdidas económicas. Competitividad ligeramente baja.
85% < OEE < 95%	Buena	Buena Competitividad. Entramos ya valores del "World Class".
OEE >99%	Excelente	Competitividad Excelente.

La clasificación del OEE permite ser más competitivo en el desarrollo de procesos y monitoreo para conocer los verdaderos factores detrás de los problemas de producción y de esta forma determinar las causas raíces, además de proporcionar información sobre dónde y cómo se pueden realizar cambios de ingeniería para obtener el mejor rendimiento de la maquinaria nueva y existente, y si se requiere una inversión en equipo adicional.

Es importante asegurarse de que las máquinas se están utilizando en su modo de eficiencia y eficacia óptimas, esto con el fin de lograr el máximo retorno de la inversión en el menor tiempo posible (ROI por sus siglas en inglés).

1.2 Planteamiento del problema

En la industria manufacturera uno de los objetivos constantes es mejorar el rendimiento de procesos para que la operación sea más rentable. La implementación de robots en operaciones claves proporciona un aumento de productividad y calidad, logrando así maximizar el rendimiento de operación.

Hoy en día, siguen existiendo empresas de manufactura que utilizan el método tradicional para sus máquinas y robots, este método se enfoca en:

- Productividad: Cantidad de unidades producidas relacionadas a sus defectos y cantidad de unidades defectuosas.
- Tiempo de inactividad: Tiempo de espera de una máquina o robot provocado por fallas en su funcionamiento y averías.

Por otra parte, existen empresas que tienen como indicador principal el OEE para sus máquinas y robots, con este indicador es posible categorizar cada una de las deficiencias las cuales son:

- Averías
- Tiempo de espera
- Microparadas
- Reducción de velocidades
- Scrap
- Retrabajos.

Las esperas en las estaciones de producción representan una de las grandes pérdidas en el día a día de la industria, ya que se depende del material y del tiempo en que el operador ejecuta el subproceso.

Para no depender del factor humano en las operaciones en que puede causar desviaciones significativas en el subproceso en la operación de equipos, las empresas de manufactura han colocado robots, lo cual da como resultado que los tiempos de espera se vean reducidos y así lograr tiempos uniformes en el subproceso.

Por otra parte, tanto el OEE como el método tradicional generan informes diarios, semanales y mensuales. Cabe mencionar que la elaboración de éstos es una tarea compleja ya que se capturan de forma manual, lo que los hace susceptibles de errores humanos, olvidos y desfases de horarios.

Aun así, estos datos se digitalizan en una hoja de cálculo, por lo cual no existe certeza de que dichos datos sean totalmente fiables al momento de hacer una evaluación de la situación real de las máquinas y robots. Esto provoca dificultad a las gerencias operativas para la toma de decisiones, además de que la información no está disponible en tiempo real.

Esta falta de control en tiempo real de los indicadores del OEE impacta, ya que no se puede detectar de manera inmediata la mala gestión de los procesos en el área de producción o en otras áreas de la empresa tales como materiales, compras, planeación y logística, lo cual provoca que las fallas en los procesos no se identifiquen e impacten de forma importante; pudiendo haberse tomado acciones preventivas en su momento si se tuviera un buen control de estos indicadores.

Esto origina que los costos de manufactura se incrementen, ya que se tiene que recurrir al tiempo extra para alcanzar la meta esperada de unidades producidas (*yield*), y de igual forma es necesario expeditar el producto terminado para cumplir en tiempo y forma con el cliente, ya que de no hacer esto, se tendría una penalización.

A pesar de todo lo anterior, hoy en día no existe una herramienta o dispositivo que permita obtener y monitorear el OEE en tiempo real en estaciones y celdas de producción en las que los robots ejecutan carga y descarga de material. Objetivo general

1.3 Objetivo general

Desarrollar e implementar una aplicación que permita obtener el OEE y sus diferentes indicadores en tiempo real en líneas y celdas de producción donde se utilicen robots para carga y descarga de material.

1.3.1 Objetivos específicos

- Definir el equipo a utilizar en la celda de producción.
- Definir las metas del OEE respecto al tiempo de operación, velocidad y unidades producidas por máquina.
- Definir la secuencia que debe seguir el robot para ejecutar el proceso de carga y descarga.
- Definir los tiempos de carga y descarga del robot.
- Diseñar un modelo que contemple las metas del OEE por máquina, la configuración del flujo de secuencia y los tiempos de carga y descarga del robot.
- Desarrollar una aplicación que contemple el modelo obtenido.
- Desarrollar una aplicación que tenga la capacidad de simular operaciones del modelo.
- Procesar la información obtenida en tiempo real, para detectar las principales causas de ineficiencia de las máquinas.
- Elaborar un reporte digital donde se resuma el OEE de todas las estaciones.

1.4 Hipótesis

Es posible determinar el OEE de manera automática por hora y por turno en una celda automatizada con robots colaborativos a través de una herramienta de software.

1.5 Justificación del proyecto

La importancia de desarrollar este proyecto estriba en que no existe una aplicación que tenga las funciones para determinar y calcular el OEE en tiempo real, y que incluya el tiempo de manejo de un robot manipulador (carga y descarga de cada operación), además de que tenga la capacidad de comparar resultados con las metas establecidas respecto a:

1. Tiempo de operación de la máquina.
2. Velocidad de la máquina.
3. Unidades producidas por máquina.

Aunado a esto, se requiere de una aplicación que tenga la habilidad de categorizar y clasificar las deficiencias de una máquina de manera automática para establecer una matriz de estado y de esta forma controlar la secuencia del robot en una celda de producción.

Al contar con una herramienta de obtención de OEE en tiempo real se tendrán los siguientes beneficios:

- Monitoreo constante de la eficiencia de producción.
- Reducción en el tiempo de investigación en el análisis del motivo de la parada.
- Reducción de gastos debido a la disminución de pérdidas.
- Incremento de la satisfacción del cliente debido a la mejora de calidad.

- Acorta el retorno de inversión debido al aumento del funcionamiento de la máquina.

Estos beneficios representarán herramientas importantes para la toma de decisiones de forma eficaz y en tiempo real, a la vez que se contará con reportes confiables que eliminen el rango de error por vaciado de datos de forma manual.

1.6 Alcances y limitaciones

1.6.1 Alcances

La aplicación proyectada tendrá los siguientes alcances:

1. Accesibilidad de modo web.
2. Permitirá configurar por máquina y en modo web el tiempo de operación, velocidad de producción y unidades producidas por turno.
3. Se tendrá la capacidad de configurar la secuencia en modo web (máquina de estados) que debe seguir el robot para ejecutar el proceso de carga y descarga.
4. Realizará el cálculo del OEE en tiempo real por máquina y por celda.
5. Obtendrá el tiempo de carga y descarga del robot manipulador.
6. Permitirá introducir la justificación del no cumplimiento del tiempo de operación, velocidad y unidades producidas por turno.
7. Generación de reportes del OEE directamente de las máquinas.
8. Visualización del OEE por máquina en tiempo real.

1.6.2 Limitaciones

La siguiente aplicación tendrá las siguientes limitaciones:

- Los valores calculados del OEE en tiempo real tendrá un retardo de 3 segundos máximo, ya que la velocidad de la aplicación dependerá del tráfico de la red y las peticiones pendientes del servidor.
- Las máquinas deben estar conectadas a la red de área local (LAN).
- No tendrá la capacidad de determinar las causas del tiempo de espera o las averías, se tendrán que retroalimentar en la aplicación.
- Deberán predefinirse en el robot los movimientos a realizar para cada máquina con antelación.
- El flujo del proceso deberá estar registrado en una matriz de estados.
- Deberán predefinirse en la matriz de estados los estados iniciales y finales de cada operación con antelación.

1.7 Estado del arte

Para desarrollar la presente investigación se tomaron en cuenta las siguientes fuentes:

1. El artículo de Data driven management in Industry 4.0: a method to measure Data Productivity Del Politecnico di Milano de Italia. En este artículo se menciona que el concepto OEE hace énfasis en los datos, bajo el nuevo lema "los datos son el nuevo petróleo". A pesar de que muchos gerentes prometen lealtad a los principios de la toma de decisiones basada en datos, todavía no existe un enfoque integral para medir qué tan efectiva es una empresa para explotar el potencial de sus propios activos de información; en otras palabras, no existe una medida de "productividad de utilización de datos". Esta

tendencia toma en cuenta factores técnicos y organizativos útiles para las empresas al momento de evaluar su nivel actual de productividad y capacidad de ejecutar líneas de acción para mejorarlo (Miragliotta, Sianesi, Convertini, & Distante, 2018).

2. El artículo On the Analysis of Effectiveness in a Manufacturing Cell: A Critical Implementation of Existing Approaches de la University of Modena de Italia, se analiza la literatura sobre OEE y selecciona cuatro metodologías principales para su evaluación y examinar las diferencias subyacentes entre ellas. Se analiza un estudio de caso de la vida real para ilustrar los problemas que surgen durante la recopilación de datos y las diferencias en los resultados obtenidos junto con conclusiones trazables para mejorar el rendimiento de los sistemas de producción, tanto en plantas industriales tradicionales como en las innovadoras, siguiendo los principios de la Industria 4.0 (Rita, Luca, Francesco, & Rimini, 2017).
3. El artículo Method for Assessing the Total Cost of Ownership of Industrial Robots de la Linnaeus University de Suecia expone que la introducción de robots industriales es una estrategia eficaz contra la creciente competencia y la subcontratación de bajo costo a países. En este artículo se analiza cómo la inversión en células de robot industrial a menudo se basa en la oferta inicial y su tiempo de amortización. Sin embargo, las decisiones de adquisición basadas sólo en el precio inicial a menudo representan decisiones deficientes, ya que no consideran los factores de costos ocultos como el mantenimiento o el consumo de energía (Landscheidta & Kansb, 2016).
4. El artículo Multi-criteria Evaluation of Manufacturing Systems 4.0 under Uncertainty del Karlsruhe Institute of Technology de Alemania expone que el uso de Manufacturing Systems 4.0 es esencial para el funcionamiento de las empresas industriales competitivas a nivel global. Este documento aporta un enfoque para evaluar los sistemas de fabricación 4.0. La incertidumbre se

integra a través de la teoría de conjuntos difusos y los modelos estocásticos (Liebrecht, Jacob, Kuhnle, & Lanza, 2017).

5. En el artículo Visual Management System to Manage Manufacturing Resources de Stellenbosch Univeristy de la universidad de Sudáfrica, se expone que el uso del Internet of Things se ha difundido en la industria manufacturera, ya que al implementarse para monitorear procesos de manufactura el control de éstos aumenta de forma notable. Además, los sistemas de producción inteligentes tienen la capacidad de integrar los mundos virtual y físico para alcanzar una mejor visualización de los procesos de producción. Estos sistemas de producción inteligentes van más allá de los medios tradicionales de colaboración para hacer que las empresas cambien su estatus de “buena” a “excelente”. El sistema de producción intelignte recopila datos de planta y los refleja en un panel (Steenkampa, Hagedorn-Hansen, & Oosthuizenc, 2017).

Estas fuentes ofrecieron antecedentes, casos de estudio e implementación, estadísticas, datos e información importante para el desarrollo del presente proyecto.

CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se integrará el marco teórico mediante el análisis y descripción de las teorías, conceptos, parámetros y valores involucrados en el desarrollo del presente proyecto de implementación.

2.1 Eficiencia General de los equipos

La Eficiencia General de los equipos (Overall Eficiency Equipment) u OEE, por sus siglas en inglés, Es un indicador propuesto por Nakajima (1988) dentro del entorno del *Total Productive Maintenance* (TPM), como una métrica para el análisis de eficiencia de la utilización de los recursos dentro de una línea de fabricación (Álvarez, 2018). Este indicador ha permitido a muchas empresas realizar mejoras enfocadas a la optimización y/o eliminación de las pérdidas. El OEE, de acuerdo como lo definió Nakajima está en términos de la utilización óptima del tiempo que los equipos y los recursos están disponibles para producir.

Desde el punto de vista conceptual, el OEE se define específicamente como "el producto de tres indicadores fundamentales que son la disponibilidad, rendimiento y calidad en valores porcentuales" (Baeza, 2016, pág. 12), como se expresa a continuación.

2.1.1 Valores porcentuales

Como se mencionó en el apartado anterior, el OEE resulta de multiplicar tres indicadores en valores porcentuales que son la disponibilidad, rendimiento y calidad expresados de la siguiente forma:

$$OEE = \%Disponibilidad \times \%Rendimiento \times \%Calidad$$

Estos valores porcentuales se definirán en los apartados siguientes.

2.1.1.1 Disponibilidad

Es el tiempo real productivo de una máquina, respecto al tiempo total disponible de la misma.

Ecuación para determinar la disponibilidad:

$$Disponibilidad = \frac{Tiempo\ productivo}{Tiempo\ disponible} x100$$

2.1.1.2 Rendimiento

Es la producción real de una máquina, respecto a su capacidad productiva.

Ecuación para determinar el rendimiento:

$$Rendimiento = \frac{Produccion\ real}{Capacidad\ productiva} x100$$

2.1.1.3 Calidad

El porcentaje de calidad se calcula dividiendo las unidades conformantes sobre las unidades totales producidas. Las unidades conformantes son las que pasaron criterios de calidad.

Ecuación para determinar la calidad:

$$Calidad = \frac{Unidades\ Conformantes}{Produccion\ Total} x100$$

2.1.2 Determinación de OEE en equipos de producción

Se presentarán ejemplos de cálculo de OEE

2.1.2.1 Ejemplos:

A continuación, en la Tabla 2.1 se muestra un primer ejemplo de cómo determinar el OEE en un equipo de producción tomando en cuenta los valores de disponibilidad, rendimiento y calidad:

Tabla 2.1 Determinación del OEE en un equipo de producción

	Tiempo disponible:8h Velocidad estándar: 1000 piezas/hora Objetivo: 8000 piezas por turno	100%	
Disponibilidad	**Sólo 6 horas productivas de 8 horas disponibles, debido a paradas: tiempos de arranque, cambios, averías esperas, etc. Capacidad productiva: 6000 piezas/turno**	**1.- (Tiempo productivo)/(Tiempo disponible)x100 2.-(6 horas)/(8 horas)x100=0.75**	**75%**
Rendimiento	**Fabricadas a una media de 700 piezas/hora, debido a micro paros y velocidad de máquina reducida. Piezas reales fabricadas: 4200 piezas/turno**	**1.- (Producción real)/(Capacidad productiva)x100 2.-(4200)/(6000)x100=0.7**	**70%**
Calidad	**Del total de las piezas fabricadas, 168 piezas son defectuosas Piezas buenas fabricadas: 4032 piezas/turno**	**1.- (Unidades buenas)/(Producción Total)x100 2.-(4032)/(4200)x100=75%**	**96%**
OEE	**Se han producido 4032 piezas buenas en el turno, frente a una capacidad productiva de 8000 piezas/turno**	**"1.-Disponibilidad 75% X Rendimiento 70% X Calidad 96%"**	**50.400%**

En la tabla 2.1 *Determinación del OEE en un equipo de producción* se observa que el OEE resultante es de un 50.40. Este valor dentro de la clasificación es inaceptable.

Ahora, en la Tabla 2.2 se hará un segundo ejemplo con el mismo ejercicio anterior pero mejorando el rendimiento, la disponibilidad y la calidad, en el que las horas productivas sean de 7 horas y la media de piezas fabricada sea de 900 piezas/turno contra 250 piezas de unidades defectuosas:

Tabla 2.2 Segundo ejemplo determinación OEE en un equipo de producción

	Tiempo disponible:8h Velocidad estándar: 1000 piezas/hora Objetivo: 8000 piezas por turno	100%	
Disponibilidad	**Solo 7 horas productivas de 8 horas disponibles, debido a paradas: tiempos de arranque, cambios, averías esperas, etc. Capacidad productiva: 7000 piezas/turno**	**1.- (Tiempo productivo)/(Tiempo disponible)x100 2.-(7 horas)/(8 horas)x100=0.875**	**88%**
Rendimiento	**Fabricadas a una media de 800 piezas/hora, debido a micro paros y velocidad de máquina reducida. Piezas reales fabricadas: 5600 piezas/turno**	**1.- (Producción real)/(Capacidad productiva)x100 2.-(5600)/(7000)x100=0.8**	**80%**
Calidad	**Del total de las piezas fabricadas, 196 piezas son defectuosas Piezas buenas fabricadas: 5404 piezas/turno**	**1.- (Unidades buenas)/(Producción Total)x100 2.-(5404)/(5600)x100=0.965**	**97%**
OEE	**Se han producido 5404 piezas buenas en el turno, frente a una capacidad productiva de 8000 piezas/turno**	**1.-Disponibilidad 0.875% X Rendimiento 0.8% X Calidad 0.965%**	**67.550%**

Implementando estas mejoras se puede ver que el OEE se incrementó a una clasificación aceptable, donde se encuentra en proceso de mejora. Ahora, si tan sólo se mejora el tiempo de ciclo de 800 piezas/hora a 850 piezas/hora y el tiempo horas productivas es de 7.5 horas.

Es de notar en la Tabla 2.3 se ve el tercer ejemplo que es la determinación OEE en un equipo de producción, el incremento es notable.

Tabla 2.3 Tercer ejemplo determinación OEE en un equipo de producción

	Tiempo disponible:8h Velocidad estándar: 1000 piezas/hora Objetivo: 8000 piezas por turno	100%	
Disponibilidad	**Solo 7.5 horas productivas de 8 horas disponibles, debido a paradas: tiempos de arranque, cambios, averías esperas, etc. Capacidad productiva: 7500 piezas/turno**	**1.- (Tiempo productivo)/(Tiempo disponible)x100 2.-(7.5 horas)/(8 horas)x100=0.9375**	**94%**
Rendimiento	**Fabricadas a una media de 850 piezas/hora, debido a micro paros y velocidad de máquina reducida. Piezas reales fabricadas: 6375 piezas/turno**	**1.- (Producción real)/(Capacidad productiva)x100 2.-(6375)/(7500)x100=0.85**	**85%**
Calidad	**Del total de las piezas fabricadas, 250 piezas son defectuosas Piezas buenas fabricadas: 6125 piezas/turno**	**1.- (Unidades buenas)/(Producción Total)x100 2.-(6125)/(6375)x100=0.96078431372549**	**96%**
OEE	**Se han producido 6125 piezas buenas en el turno, frente a una capacidad productiva de 8000 piezas/turno**	**1.-Disponibilidad 0.9375% X Rendimiento 0.85% X Calidad 0.96078431372549%**	**76.563%**

Como un cuarto ejemplo se tiene que para obtener una mejor clasificación de OEE se colocara un robot en la ejecución de dicha operación, la mejora es significativamente mayor, como se puede observar en la tabla 2.4 Ejemplo determinación OEE de producción con robot:

Tabla 2.4 Ejemplo determinación OEE en un equipo de producción con robot

	Tiempo disponible:8h Velocidad estándar: 1000 piezas/hora Objetivo: 8000 piezas por turno	100%	
Disponibilidad	**Solo 7.9 horas productivas de 8 horas disponibles, debido a paradas: tiempos de arranque, cambios, averías esperas, etc. Capacidad productiva: 7900 piezas/turno**	**1.- (Tiempo productivo)/(Tiempo disponible)x100 2.-(7.9 horas)/(8 horas)x100=0.9875**	**99%**
Rendimiento	**Fabricadas a una media de 950 piezas/hora, debido a micro paros y velocidad de máquina reducida. Piezas reales fabricadas: 7505 piezas/turno**	**1.- (Producción real)/(Capacidad productiva)x100 2.-(7505)/(7900)x100=0.95**	**95%**
Calidad	**Del total de las piezas fabricadas, 50 piezas son defectuosas Piezas buenas fabricadas: 7455 piezas/turno**	**1.- (Unidades buenas)/(Producción Total)x100 2.-(7455)/(7505)x100=0.993337774816789**	**99%**
OEE	**Se han producido 7455 piezas buenas en el turno, frente a una capacidad productiva de 8000 piezas/turno**	**1.-Disponibilidad 0.9875% X Rendimiento 0.95% X Calidad 0.993337774816789%**	**93.188%**

En este último ejemplo se puede notar que la clasificación pasa de ser buena a otra donde su valor es de “World Class”.

2.1.3 Pérdidas

Por otra parte, el indicador OEE contempla a su vez 6 tipos de pérdidas significativas:

- Pérdidas por tiempo del mantenimiento.
- Pérdidas por tiempo de la disponibilidad.
- Pérdidas por tiempo ocioso.
- Pérdidas por reducción de la velocidad.
- Pérdidas por calidad.
- Pérdidas por rendimiento.

A continuación, se definirán estos tipos de pérdidas.

2.1.3.1 Primera pérdida (tiempo de mantenimiento): Averías

Es un fallo repentino o avería que genera una pérdida en el tiempo de producción. La causa de esta disfunción puede ser técnica u organizativa, como el tiempo que se pierde cuando se detiene la producción de una línea por un cambio de ingeniería.

2.1.3.2 Segunda pérdida (tiempo de disponibilidad): Tiempos de espera

El tiempo de producción se reduce también cuando la máquina está en espera. La máquina puede quedarse en estado de espera por varios motivos, por ejemplo, debido a una espera de aprobación de calibración por parte del área de calidad o ingeniería de procesos, por mantenimiento, o por un paro por parte del operador para ir a descansar o a comer.

2.1.3.3 Tercera Pérdida (pérdidas por tiempo ocioso): Microparadas

Se conoce como *microparadas* las interrupciones cortas de una máquina que provocan que no trabaje a velocidad constante. Estas microparadas y las consecuentes pérdidas de velocidad son generalmente causadas por pequeños problemas tales como bloqueos producidos por sensores de presencia o bloqueos en las cintas transportadoras. Estos aparentemente pequeños problemas pueden disminuir de forma drástica la efectividad de la máquina.

2.1.3.4 Cuarta Pérdida (Reducción de velocidad)

La velocidad reducida es la diferencia entre la velocidad que se fija para ejecutar el proceso real y la velocidad teórica o de diseño. En muchos casos, la velocidad de producción se disminuye para evitar otras pérdidas tales como defectos de calidad y averías.

2.1.3.5 Quinta Pérdida (Calidad): Deshechos (Scrap)

Los deshechos o *scrap* son aquellos productos que no cumplen los criterios establecidos por el área de calidad, por lo que no pueden seguir en el proceso y representan pérdidas en cuanto a tiempo y material.

2.1.3.6 Sexta Pérdida (Rendimiento): Retrabajo

Los productos *retrabajados* son también productos que no cumplen los requisitos de calidad en una primera inspección, pero pueden ser reprocesados para convertirlos en productos que cumplan con los criterios del área de calidad (Cruelles Ruiz, 2010).

2.1.3.7 Determinación del indicador OEE tomando en cuenta los valores de pérdidas

A continuación, en la Tabla 2.5 se muestra de manera resumida el cálculo del OEE:

Tabla 2.5 Sumariada del cálculo del OEE

Planeación	**Tiempo Total**		
	Tiempo Disponible		Mantenimiento, Cambio de productos, Descansos
Disponibilidad	**A Tiempo Disponible**		
	B Tiempo Productivo	1.Averías 2.Esperas	
Rendimiento	**C Capacidad Productiva**		
	D Producción real	1.Microparadas 2.Velocidad reducida	
Calidad	**E Producción Real**		
	F Piezas Conformes	1.Scrap 2.Retrabajo	

Ecuación de OEE tomando en cuenta las 6 grandes pérdidas

$$OEE = \frac{B}{A} x \frac{D}{C} x \frac{F}{E} = Calidad$$

Donde, según la tabla 6:

$$\frac{B}{A} = \text{Disponibilidad} \quad \frac{D}{C} = Rendimiento \quad \frac{F}{E} = Calidad$$

En general, el OEE es usado e interpretado como un valor promedio si se considera su variabilidad, se gestiona de una forma completamente determinística para la toma de decisiones.

Por tanto, es importante analizar OEE desde este enfoque que permite cuantificar las mejoras y la reducción de pérdidas basados en el aumento promedio del OEE.

2.2 Gestión de proyectos informáticos

El desarrollo de paquetes informáticos ha sido de gran utilidad para los desarrollos integrales específicos (DSI) que han logrado racionalizar sus estructuras, tanto de hardware como de software. Durante las últimas décadas en las empresas manufactureras han aumentado de forma vertiginosa las necesidades de interoperación, por lo que fue necesario rediseñar soluciones para satisfacer estas nuevas necesidades.

La lógica EAI (*Enterprise Aplication Integration*) transforma el modelo funcional de la empresa en un modelo técnico, en el que las aplicaciones están interconectadas. Este modelo contempla dos funciones imprescindibles en su desarrollo: 1) Los analistas se encargan del repositorio de procesos (el mapeo funcional general) 2) Los planificadores son los responsables de la puesta en marcha técnica de la arquitectura que implementa el repositorio de procesos.

El modelo funcional superpone los mapeos funcionales del conjunto de aplicaciones de la empresa (Guerín, 2015).

2.2.1 Sistema de gestión de bases de datos relacional MySQL

Una base de datos es un conjunto estructurado de datos que administra un equipo. Una base de datos permite almacenar datos y también clasificarlos, de esta manera se pueden encontrar rápidamente con el lenguaje SQL (Structured Query Language). El lenguaje SQL se utiliza para ejecutar acciones en la base de datos, como crear tablas, añadir o eliminar datos (Camps, Casillas, & Dolors, 2005).

2.2.1.1 Los servicios web REST

Los servicios web SOAP (WS) representan los nuevos pilares de la programación distribuida. La implementación de REST definitivamente diferirá de un desarrollador a otro. En la figura 2-1 se muestra un *Diagrama de un servicio basado en REST* se puede observar un ejemplo muy simple de un servicio basado en REST.

En este diagrama se ilustra el proceso de gestión de atención de un servidor: cuando una solicitud llega al servidor, se sirve y se descarta. Por lo tanto, la siguiente solicitud no dependerá del estado de la anterior. Cada solicitud es manejada por el servidor de forma independiente. Las solicitudes se realizan en una conexión HTTP. Cada uno de ellos toma la forma de un identificador de recursos (URI). Este identificador beneficia y ayuda a localizar el recurso requerido en el Servidor web (Aroraa & Dash, 2018).

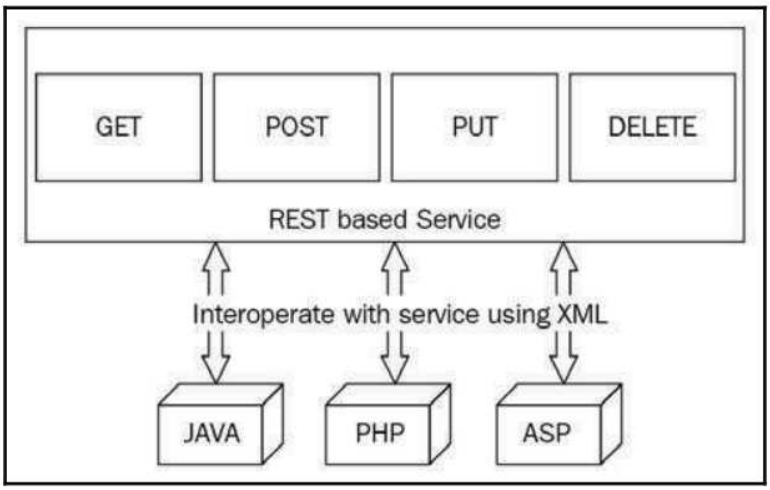


Figura 2-1 Diagrama de un servicio basado en REST

2.2.1.2 Arquitectura cliente-servidor

El cliente del servicio no debe preocuparse por cómo el servidor procesa los datos y los almacena en la base de datos. Del mismo modo, el servidor no necesita depender de la implementación del cliente, especialmente la interfaz de usuario.

2.2.2 Angular

AngularJS es un framework JavaScript open source, desarrollado por Google, que permite facilitar la creación de una *Aplicación de una Sola Página* (Single Page Application) o SPA. Su rol es proporcionar todos los mecanismos técnicos necesarios para la creación de este tipo de aplicaciones y proporcionar una estructura que permita desarrollar una aplicación robusta y organizada (OLLIVIER, 2016).

2.2.2.1 Organización de la aplicación

A nivel de la estructura de la aplicación, AngularJS se basa en el patrón de diseño MVC, Model-View-Controller. Este patrón de diseño propone organizar una aplicación en tres partes: modelo, vista y controlador. En este patrón, el modelo está compuesto por los datos que deben mostrarse, pero no posee ninguna lógica. Se trata simplemente del contenedor.

La vista se encarga, por su parte, de mostrar los datos del modelo para generar la página que será visible por los usuarios. Tiene como rol transmitir las acciones realizadas al controlador, como el clic en el botón o la selección de un elemento de una lista.

El controlador es el elemento central de este patrón, puesto que es responsable de crear el modelo, contactando una API web o consultando una base de datos, por ejemplo, y a continuación transmitirlo a la vista. Su rol es, también, responder a las acciones de los usuarios, transmitidas por la vista, para realizar el procesamiento esperado (OLLIVIER, 2016).

La ventaja que aporta este patrón estriba, principalmente, en separar la aplicación en diferentes elementos que tienen distintas responsabilidades bien definidas. Esto permite tener un código más estructurado y, por tanto, más claro, aportar una mejor mantenibilidad y ofrecer la posibilidad a los desarrolladores de probar unitariamente los controladores.

AngularJS se inspira, a su vez, en el patrón de diseño MVVM, Model-View-ViewModel, que define una organización bastante similar a la del patrón MVC, por su noción de *binding*. El *binding* permite sincronizar un dato del modelo con su representación en la vista, de manera que una modificación del modelo provoca automáticamente una actualización de la vista, y a la inversa.

Como el uso de los patrones MVC y MVVM no es suficiente para organizar una aplicación entera, estos patrones se refieren únicamente a las capas UI. AngularJS introduce otras nociones, como los servicios, cuyo rol es encapsular las funcionalidades de negocio y técnicas, o como los módulos, que permiten agrupar elementos que pueden reutilizarse en varias aplicaciones AngularJS (OLLIVIER, 2016).

2.3 Programación orientada a objetos

La Programación Orientada a Objetos (POO) es un concepto básico para el desarrollo de sistemas automatizados, su consistente base teórica y la amplia gama de herramientas que permiten crear códigos a través de diseños orientados a objetos la convierten en la alternativa más adecuada para el desarrollo de aplicaciones en la actualidad. Dado lo anterior, este concepto es primordial para la presente investigación, por lo que en este apartado se abordará dicho tema.

2.3.1 Concepto

La POO es el paradigma de programación más utilizado en la actualidad. Este concepto introduce la noción de conjunto coherente de datos y acciones, trasponiendo al mundo del desarrollo a conceptos comunes e intuitivos propios del mundo con el que se convive.

En efecto, se utiliza de manera cotidiana los objetos con todas las propiedades y acciones que tienen asociadas. Estos objetos también pueden interactuar entre sí o estar compuestos unos por otros, lo que permite formar sistemas complejos (Hugon, 2015). Un ejemplo representativo de esto sería un automóvil, ya que es un objeto que representa un sistema complejo al estar compuesto por otros objetos con sistemas propios.

2.3.2 Historia de la programación orientada a objeto

El concepto de *programación orientada a objetos* no es reciente. En los años 60, dos investigadores noruegos, Quisten Negar y Ole-Johan Del desarrollaron la base de la programación orientada a objetos, creando el lenguaje *Simula*.

Las nociones básicas de la POO como las clases, la herencia, los métodos virtuales, etc., fueron creados en este lenguaje para permitir modelizar de manera fidedigna procesos industriales complejos.

Simula-67 representó el parteaguas de la vía de los lenguajes orientados a objetos, como Smalltalk, y posteriormente, C++, Java y C# entre otros, que explotarían estos conceptos algunas décadas más tarde (Gervais, 2017).

En 1980 se implementó el Smalltalk, que es el primer lenguaje orientado a objetos disponible con un entorno de desarrollo gráfico integrado. Smalltalk fue diseñado por el equipo del americano Alan Kay, del centro de investigación informática californiana de XEROX (el famoso Palo Alto Research Center). Este lenguaje retoma y complementa los conceptos básicos, fundamentalmente con la noción de compilación dinámica de un código "intermedio", portable entre entornos heterogéneos, en un código máquina destino. Java recicla este concepto con su Just In Time Compiler, y también para C#.

En el mismo año de 1980, el danés Bjarne Stroustrup desarrolló para AT&T el lenguaje C++ (llamado originalmente C with classes). C++ es una evolución del lenguaje C inventado en UNIX por el tándem canadiense-americano Kernighan y Ritchie, diez años antes. El lenguaje C++ todavía es muy utilizado, incluso por sus detractores, que lo juzgan como demasiado complejo y no muy orientado a objetos, gracias a su compatibilidad con C (Gervais, 2017).

En la década de 1990 James Gosling desarrolla el lenguaje Oak para Sun Microsystems. Oak se transformó en Java en 1995. Este lenguaje fue diseñado para ser independiente del hardware que ejecuta sus programas "precompilados", para ser robusto y seguro y, sobre todo, más sencillo de programar que C++. En la actualidad, el lenguaje Java se utiliza ampliamente tanto en entornos Web como en

estaciones de trabajo, teléfonos y otros dispositivos, como las tabletas. Este lenguaje pertenece a la paquetería de Oracle.

En 2001, Microsoft presenta C# (pronunciado C Sharp); lenguaje desarrollado por el danés Anders Hejlsberg, creador también del Turbo Pascal y el arquitecto de Delphi. C# es un lenguaje orientado a objetos cercano a Java, que permite programar aplicaciones en entornos Microsoft, como estaciones de trabajo, aplicaciones web, smartphones o tabletas (Gervais, 2017).

2.3.3 Elementos de la programación orientada a objetos

En este apartado se definirán los diferentes elementos de la programación orientada a objetos.

2.3.3.1 Objeto

Es un elemento autocontenido utilizado por el programa. Los valores que almacena un objeto se denominan atributos, variables o propiedades. Los objetos pueden realizar acciones, que se denominan métodos, servicios, funciones, procedimientos u operaciones. Los objetos tienen un gran sentido de la privacidad, por lo que solo dan información sobre sí mismos a través de los métodos que poseen para compartir su información. También ocultan la implementación de sus procedimientos, aunque es muy sencillo pedirles que los ejecuten. Los usuarios y los programas de aplicación no pueden ver qué hay dentro de los métodos, sólo pueden ver los resultados de ejecutarlos. A esto se denomina ocultación de información o encapsulamiento de datos.

Cada objeto presenta una interface pública al resto de objetos que pueden utilizarlo. Una de las mayores ventajas del encapsulamiento es que mientras que la interface pública sea la misma, se puede cambiar la implementación de los métodos sin que sea necesario informar al resto de objetos que los utilizan. Para pedir datos a

un objeto o que éste realice una acción se le debe enviar un mensaje. Un programa orientado a objetos es un conjunto de objetos que tienen atributos y métodos. Los objetos interactúan enviándose mensajes (Márques, 2002).

2.3.3.2 Encapsulación

La encapsulación es la propiedad mediante la cual cada objeto expone los datos y funciones que permiten a los demás objetos interactuar con él y, a su vez, sólo con estos. Todos los datos propios del funcionamiento del objeto, que son inútiles desde un punto de vista de las interacciones con el exterior, permanecen ocultos. El objeto se comporta, de este modo, como una caja negra que expone solamente ciertos datos y funcionalidades cuyo uso no podrá corromper el estado del objeto (Hugon, 2015).

2.3.3.3 Herencia

La herencia define una relación entre dos clases. Esta relación significa que una clase expone las mismas funcionalidades y datos que otra clase, agregando potencialmente sus propias funcionalidades. Generalmente, se utiliza la herencia para implementar la noción de especialización.

Esto significa que, cuando una clase es la clase hija en una relación de herencia, se dice que deriva de su clase madre (Hugon, 2015).

2.3.3.4 Polimorfismo

Otro concepto fundamental de la POO es el *polimorfismo*, que está en relación directa con la herencia. El polimorfismo es, en efecto, la capacidad de un objeto para ser visto según distintas formas. Pero estas formas no son aleatorias, sino que

dependen directamente de la jerarquía de herencia de la que forma parte el objeto. Cada clase puede verse en su propia forma, pero también en la de su clase madre, así como según todas las clases "ancestras" de su clase madre. Cada objeto tiene, por tanto, como mínimo dos formas posibles: la definida por su propia clase y la definida según el tipo de objeto (Hugon, 2015).

2.3.3.5 Clase abstracta

Una clase abstracta no se puede instanciar y un método abstracto no contiene implementación, solo su declaración. Esto implica que una clase abstracta sólo se puede usar en el marco de la herencia y que la clase derivada debe implementar las funcionalidades de los miembros.

El objetivo de una clase abstracta es definir la forma de las clases hijas sin tener que definir el fondo. La implementación final es responsabilidad de la clase hija (Putier, 2015).

2.3.3.6 Persistencia

Los objetos entidad que se utilizan en los programas orientados a objetos son análogos a las entidades que se utilizan en las bases de datos orientadas a objetos puras, pero con una gran diferencia: los objetos del programa desaparecen cuando el programa termina su ejecución, mientras que los objetos de la base de datos permanecen. A esto se le denomina persistencia (Márques, 2002).

2.3.3.7 Identidad

La identidad de objetos ha existido desde hace mucho tiempo en los lenguajes de programación, pero en las bases de datos es más reciente. El objetivo es contar

con objetos que tengan una existencia independiente de sus valores. Así, dos objetos pueden ser idénticos si son el mismo objeto o pueden ser iguales si tienen los mismos valores. La identidad cobra relevancia cuando un objeto se comparte con otros y cuando se actualiza. En un modelo basado en identidad, dos objetos pueden compartir un objeto hijo (Márques, 2002).

2.3.4 Definición de los modelos de objetos

Los administradores de base de datos (DBMS por sus siglas en inglés) evolucionan con el afán de satisfacer nuevos requerimientos tecnológicos y de información. Aunque los DBMS relacionales (RDBMS) son actualmente líderes del mercado y brindan las soluciones necesarias a las empresas comerciales, existen aplicaciones que necesitan funciones con las que no cuentan. Las CAD/CAM, los sistemas multimedia, como los geográficos y de medio ambiente, los de gestión de imágenes y documentos y los de apoyo a las decisiones necesitan de modelos de datos complejos, difíciles de representar como tuplas de una tabla.

En general, estas aplicaciones necesitan manipular objetos y los modelos de datos deben permitirles expresar su comportamiento y las relaciones entre ellos.

El modelo de datos orientado a objetos es una extensión del paradigma de programación orientado a objetos (Márques, 2002).

Para establecer los modelos de objeto es necesario conocer los atributos de dichos objetos, los cuales se definirán en el siguiente apartado.

2.3.5 Atributos

Los atributos definidos son representaciones abstractas sin un tipo de dato específico. Una implementación específica definirá como la información es representada. Por ejemplo:

a) Un atributo puede ser representado como una cadena o *string* en una implementación y como numérico en otra.
b) Un valor de tipo fecha pude ser representado en formato ISO en una implantación y como calendario juliano en otra.

A continuación, se definirán los atributos en tablas. Las tablas de atributos relacionan el nombre del atributo, descripción y ejemplos de las definiciones (ISA–The Instrumentation, 2001).

2.4 Requerimientos funcionales y casos de usos

Los requerimientos funcionales son los que describen lo que un sistema debe hacer, son las capacidades que un sistema debe ser capaz de realizar para cumplir con los objetivos del negocio en cuestión, describen las transformaciones que el sistema debe hacer de acuerdo a las entradas para tener como resultado las salidas que van a satisfacer los objetivos del negocio, están relacionados con el “qué” y no con el “cómo”, esto significa que los requerimientos funcionales deben responder a las necesidades directas que tienen los usuarios, satisfacer la razón por lo cual ellos se acercan e interactúan con el sistema (Larman, 2003).

Sin estos requerimientos funcionales no puede llevarse adelante el desarrollo de ninguna pieza de software, son la base para especificar las problemáticas que necesitan ser solucionadas por las aplicaciones (Larman, 2003).

Típicamente, un analista de requisitos genera requisitos funcionales después de realizar los casos de uso. Sin embargo, esto puede tener excepciones, ya que el

desarrollo de software es un proceso iterativo y algunos requisitos son previos al diseño de los casos de uso. Ambos elementos (casos de uso y requisitos) se complementan en un proceso bidireccional.

Un requisito funcional típico contiene un nombre, un número de serie único y un resumen. Esta información se utiliza para ayudar al lector a entender por qué el requisito es necesario, y para seguir al mismo durante el desarrollo del producto.

2.4.1 Casos de uso de alto nivel

El caso de uso es un documento narrativo que describe la secuencia de eventos de un actor que utiliza un sistema para completar el proceso.

Los casos de uso son historias o casos de utilización de un sistema; no son exactamente los requerimientos ni las especificaciones funcionales, sino que ejemplifican e incluyen tácitamente los requerimientos de las historias que narran.

El caso de uso de alto nivel describe el proceso muy brevemente en dos o tres enunciados. (Larman, 2003)

El caso de uso es una poderosa herramienta para obtener los requerimientos funcionales. Los diagramas de casos de uso agregan mayor poder: debido a que conciben los casos de uso, facilitan la comunicación entre los analistas y los usuarios, y entre los analistas y los clientes. En un diagrama, el símbolo del caso de uso es una elipse.

El símbolo de un actor es una figura adjunta. Una línea asociativa conecta a un actor con el caso de uso. Los casos de uso están, por lo general, dentro de un rectángulo que representan el confín del sistema. (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004)

La inclusión se representa por una línea de dependencia con un estereotipo «incluir». La extensión se representa por una línea de dependencia con un estereotipo «extender». Las otras dos relaciones entre casos de uso son generalización, en la que un caso de uso hereda el sentido y acciones de otro, y el agrupamiento. Mismo que organiza un conjunto de casos de uso. La generalización se representa por la misma línea que muestra la herencia entre clases. El agrupamiento se representa por el icono del paquete. (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004)

Los diagramas de casos de uso figuran con fuerza en el proceso de análisis, se empieza con entrevistas con los clientes para obtener los requerimientos funcionales del sistema. Éstos proporcionan una base para entrevistar a los usuarios. Las entrevistas facilitan la obtención de los requerimientos y con ellos se elaboran posteriormente los casos de uso de alto nivel. Los diagramas resultantes de caso de uso darán los fundamentos para el diseño y desarrollo. (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004).

2.4.2 Diagrama de casos de uso

El modelado de casos de uso es una técnica que permite modelar las funciones de un sistema en términos de eventos, de quien inicia los eventos y de cómo responde el sistema a estos eventos. El modelado de casos de uso está formado básicamente por dos elementos: los diagramas de casos de usos y las narraciones de casos de uso. El diagrama de casos de uso muestra el comportamiento del sistema a partir de los usuarios que interactúan con el sistema,

mientras que las narraciones de casos de uso describen de forma escrita los eventos de negocio y como interaccionan los usuarios con el sistema. Un diagrama de casos de uso representa las interacciones entre el sistema y los sistemas externos y los usuarios (Larman, 2003).

La creación de los diagramas de secuencia detallados provoca un refinamiento del diagrama de casos de uso que consiste en que deben existir los mismos actores en los diagramas de secuencia y en los diagramas de casos de uso. Si no es así hay que corregirlo (Gutierrez, 2011).

Dentro de UML se pueden encontrar diversos diagramas que permiten representar las diversas perspectivas de un sistema, a las cuales se les conoce como modelo que es una representación simplificada de la realidad. Los Casos de uso son diagramas que permiten representar lo que hará el sistema, pero no como funciona(Gutierrez, 2011).

2.5 Modelo conceptual

2.5.1 Introducción

Un Modelo Conceptual explica (a sus creadores) los conceptos significativos en un dominio del problema, es el artefacto más importante a crear durante el análisis orientado a objetos. Identificar muchos objetos o conceptos constituye la esencia del análisis orientado a objetos, y el esfuerzo se compensa con los resultados conseguidos durante la fase de diseño e implementación (Larman, 2003).

La identificación de conceptos forma parte de una investigación del dominio del problema. El lenguaje UML contiene la notación en diagramas de estructura estática que explican gráficamente los modelos conceptuales.

Una cualidad esencial que debe ofrecer un modelo conceptual es que representa cosas del mundo real, no componentes de software.

2.5.2 Conocimiento de la nomenclatura del dominio

Además de descomponer el espacio del problema en unidades comprensibles (conceptos), la creación de un modelo conceptual contribuye a esclarecer la terminología o nomenclatura del dominio. Podemos verlo como un modelo que comunica (a los interesados como pueden serlo los desarrolladores) cuáles son los términos importantes y cómo se relacionan entre sí (Larman, 2003).

El modelo del dominio muestra (a los modeladores) clases conceptuales significativas en un dominio del problema; es el artefacto más importante que se crea durante el análisis orientado a objetos.

Un modelo del dominio es una representación de las clases conceptuales del mundo real, no de componentes del software. No se trata de un conjunto de diagramas que describen clases de software, u objetos de software con responsabilidad (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004).

El modelo de dominio es una representación visual de las clases conceptuales u objetos del mundo real en un dominio de interés. Utilizando la notación UML, un modelo de dominio se representa con un conjunto de diagramas de clases en los que no se define ninguna operación. (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004).

2.5.3 Modelos Conceptuales

El paso esencial de un análisis orientado a objetos es descomponer el problema en conceptos u objetos individuales: las cosas que se saben. Un modelo conceptual es una representación de conceptos en un dominio del problema. (Fowler , 1996)

En UML se ilustra con un grupo de diagramas de estructura estática donde no se define ninguna operación. La designación de modelo conceptual ofrece la ventaja de subrayar fuertemente una concentración en los conceptos del dominio, no en las entidades del software (Larman, 2003).

Puede mostrarnos:

- Conceptos
- Asociaciones entre conceptos
- Atributos de conceptos.

2.6 Raspberry Pi

A finales del primer trimestre de 2012, aparece la diminuta tarjeta Raspberry Pi y múltiples reseñas sobre ella. Sin embargo, el desarrollo de este pequeño dispositivo empezó mucho antes de su aparición. La Fundación Raspberry Pi ha trabajado tanto en la definición del producto como en su puesta en producción. En este capítulo se describen las etapas que se han sucedido para el desarrollo de este dispositivo (Mocq, 2016).

Imagine un ordenador del tamaño de una tarjeta de crédito que cuesta menos de 40 € y que es capaz de proporcionar vídeos en 1080p (full HD = alta definición) a 30 imágenes por segundo… esto es posible gracias a la Raspberry Pi 3.

La creación de la Raspberry Pi se genera a partir de la motivación de poner a disposición de los niños y adolescentes un ordenador de bajo costo que les permita descubrir la informática y la programación.

La Raspberry Pi 3 reúne en una tarjeta de 85 × 56 mm todos los componentes necesarios para que pueda funcionar un sistema operativo (Linux) y, de esta manera, poder usar la Raspberry Pi 3 como un verdadero PC.

Para este dispositivo, se cuentan con herramientas básicas disponibles en la mayor parte de los distribuidores: procesadores de texto, hoja de cálculo, juegos o navegadores de Internet.

Los creadores de hardware como impresoras 3D, aparatos teledirigidos (vehículos, drones, helicópteros, etc.), tendrán acceso a decenas de aplicaciones útiles o incluso totalmente recreativas.

2.6.1 Historia de la Raspberry Pi

A continuación, se describen los orígenes de la Raspberry Pi.

En los años 90, los estudiantes británicos de "nivel A" (bachillerato), estaban muy interesados en la informática. Muchos de ellos se pasaban horas programando su Amiga, BBC Micro, ZX Spectrum o Comodore 64. Llegaban a tener un nivel de programador amateur y pasaban sin problemas las pruebas.

Sin embargo, los estudiantes de los años 2000 no tienen el mismo perfil. Como mucho, alguno de ellos habrá construido algunos sitios web.

A raíz de esto, se han identificado las causas de esta involución: los cursos de informática se han transformado en la formación en aplicaciones de escritorio y la

creación de páginas web. Los PC han sustituido a los antiguos micro-ordenadores, que toda una generación habían utilizado para formarse en programación.

Estos PC, mucho más costosos y complicados que la generación anterior de "pequeños ordenadores", han hecho que los padres prohíban a los programadores principiantes experimentar con la PC familiar. Éste se usa para navegar en la web, jugar y más raramente (mucho más raramente) para gestionar la contabilidad familiar. Los jóvenes se han visto obligados a manejar las consolas de videojuegos. Por ello, no han vuelto a tener acceso a la programación.

Derivado de lo anterior, era necesario encontrar una plataforma que, al igual que los antiguos ordenadores, pudiera arrancar directamente en un entorno de programación. Para contrarrestar el hecho de que no se puede dejar a un joven experimentar sus programas en un PC, es necesaria una máquina con un precio tal que la noción de "riesgo" desaparezca.

De 2006 a 2008, Eben Upton construyó varios prototipos de lo que se convertiría en la Raspberry Pi. A partir de 2008, los microprocesadores diseñados para usos móviles fueron más accesibles y lo suficientemente potentes como para ofrecer vídeo de excelente calidad. Esta característica podría hacer que la tarjeta estuviera al alcance de los jóvenes, no obligatoriamente interesados por una herramienta destinada a la programación pura.

Eben Upton (hoy diseñador de componentes en Broadcom), Robert Mullins, Jack Lang y Alan Mycroft, junto con Pete Lomas, director de ingeniería en Norcott Technologies, así como David Braben, autor del juego Elite para la BBC Micro, crearon la Fundación Raspberry Pi para que el proyecto fuera una realidad. Tres años más tarde empezó la fabricación en serie.

En referencia a esto, la comunidad educativa ha manifestado un gran interés y apoyo. Los países en vías de desarrollo están interesados en el uso de la Raspberry Pi como herramienta de productividad en las zonas donde no disponen ni de la alimentación eléctrica ni del hardware necesario para mantener un PC.

De la misma forma, hospitales y museos se han interesado en obtener información acerca de la capacidad de la Raspberry Pi para controlar los dispositivos de visualización. Padres de niños gravemente discapacitados piensan en aplicaciones de vigilancia y accesibilidad. Existe todo un mercado interesado en este dispositivo para diseñar un robot (Mocq, 2016).

2.6.2 Cronología

De 2006 a 2008, Eben Upton realizó varios prototipos de lo que se convertiría en la Raspberry Pi.

Después de 2008, se integra al proyecto un SoC (System on a Chip = circuito integrado que agrupa todos los componentes de un ordenador) y se definen las características definitivas de la Raspberry Pi.

Para agosto de 2011 se habían construido cincuenta prototipos de tarjetas alpha. Sus características eran idénticas a las de la Raspberry Pi actual, salvo por su tamaño, más grande por la presencia de conectores previstos para la depuración.

2.7 Universal Robots

Universal Robots es una compañía danesa líder mundial en el ámbito de la robótica colaborativa. Celebra sus primeros diez años como proveedor de la industria de automoción con más de 27.000 unidades vendidas en este periodo.

Con una cuota de mercado del 60% en este tipo de autómatas y un 72% de crecimiento en 2017, la compañía, reconocida recientemente con el Premio a la Innovación Tecnológica, suministra sus cobots a manufactureras como Ford, Groupe PSA, Renault y Volkswagen, entre otros.

Tras crear la compañía en 2005, el fundador y director técnico de Universal Robots entregó el primer robot después de haber dirigido un pequeño equipo durante tres años de desarrollo en un sótano de la Universidad del Sur de Dinamarca. Por su papel pionero en el desarrollo de cobots, ha sido galardonado este año con el Premio Engelberger, el considerado como el "Premio Nobel" de robótica.

En 2016 la compañía danesa lanzó Universal Robots+, una nueva plataforma que aprovecha el innovador ecosistema global de la compañía permitiendo a los desarrolladores de terceros crear productos como pinzas, sistemas de visión, software y otros accesorios, muy utilizados en el sector del envase y embalaje, certificados para trabajar sin problemas con los cobots de UR. Actualmente, el showroom de UR+ incluye alrededor de 130 productos UR+ certificados y más de 390 compañías de desarrollo comercial aprobadas en el programa de desarrollo de UR+.

Un año más tarde, en 2017, se creó la Universal Robots Academy para aumentar la alfabetización en torno a los cobots. Actualmente, consta de nueve módulos interactivos gratuitos de formación en línea sobre el dominio de la programación, la instalación y el funcionamiento de los cobots. El programa ha sido ampliamente adoptado en todo el mundo, con más de 45.000 usuarios de más de 130 países, ya que los módulos están disponibles en ocho idiomas, incluyendo español, inglés, alemán, francés, chino, japonés, coreano y tailandés. Universal Robots es el único proveedor de cobots que ofrece formación en robótica de esta importancia de forma gratuita.

Según la firma escandinava, los cobots son ahora el segmento de mayor crecimiento de la automatización industrial, donde el envasado y el embalaje adquieren una gran importancia y se espera que en 2025 se multipliquen por diez hasta alcanzar el 34% de todas las ventas de robots industriales, según datos de la Asociación de Industrias Robóticas (RIA, por sus siglas en inglés).

Con el fin de fortalecer su posición de liderazgo en esta especialidad, el pasado mes de junio, Universal Robots lanzó una nueva generación de sus cobots, la gama e-Series, consistente en una plataforma que eleva el estándar de los robots colaborativos y permite un desarrollo de soluciones y un despliegue aún más rápido de una mayor variedad de aplicaciones que beneficien a empresas de todos los tamaños (Østergaard, 2018).

2.8 Cobots

Tanto los *cobots* como los robots industriales se introdujeron hace ya bastantes años en el mundo de la industria y es evidente que existe una competencia entre ambos. Presentes durante décadas en las cadenas de producción, los robots industriales convencionales han demostrado durante mucho tiempo su eficacia. Pero en cuanto a ciertas dificultades, los cobots son más prácticos y eficaces. A continuación, se describirán algunas diferencias entre cobots y robots, y la forma en que se complementan.

2.8.1 ¿Qué son los cobots?

Los cobots son brazos robóticos de 6 ejes que están articulados. Desde que fueron creados, los cobots se han empleado para realizar trabajos en entornos industriales y, durante muchos años, han logrado impulsar la productividad a grandes escalas. Gracias a los avances tecnológicos los robots colaborativos son capaces hoy en día de llevar la automatización industrial a nuevas alturas; ya que pueden ocuparse de tareas inaccesibles para los robots industriales (Pelegrí, 2018).

Los cobots y los robots industriales tradicionales se diferencian en muchos aspectos. El robot industrial produce de forma masiva, ocupa mucho espacio y, a menudo, permanece en una posición fija. En cambio, el cobot es compacto y ocupa poco espacio. Además, se puede reubicar en cualquier lugar sin ningún tipo de problemas (Pelegrí, 2018).

2.8.2 Seguridad

Los robots industriales, por su actividad intensiva, en ocasiones pueden resultar peligrosos. Por esto, no pueden instalarse sin barreras de seguridad que protejan a las personas. La ausencia de sensores de fuerza (que dota al robot de la capacidad de conocer su entorno y que le permite detenerse automáticamente en caso de una intrusión en su espacio), sumado a su elevado peso, convierte a los robots industriales en herramientas potencialmente peligrosas.

Por el contrario, los cobots están equipados con sensores que les permiten detenerse en caso de obstrucción o necesidad. Gracias a su ligero peso, un choque no tendría por qué ser peligroso. En conclusión, la misión de los cobots es ofrecer la oportunidad de eliminar las barreras de seguridad y facilitar su integración en una cadena de producción (Pelegrí, 2018).

2.8.3 Uso y programación

La robótica implica programación, y para muchos profesionales, ésta sigue siendo un área específica que sólo conocen unos pocos. Esto ocurre con los robots tradicionales, pues son máquinas complejas que requieren la intervención de un experto. Ocurre lo contrario con los cobots, que son más fáciles de manejar y pueden ser programados por cualquier persona sin necesidad de pasar por un periodo de formación previo.

Además, la puesta en marcha es otro punto de diferenciación entre el robot industrial y el cobot que, a diferencia de los robots convencionales, puede estar operativo en menos de 12 horas (Pelegrí, 2018).

2.8.4 Versatilidad vs. especialidad

Realmente, la mayor diferencia entre los cobots y los robots tradicionales estriba en la visión de cada uno de ellos. Un robot tradicional se imagina, planifica, construye y utiliza con un único propósito. Y ese propósito lo podrá resolver con una eficiencia óptima. El cobot, en cambio, puede estar desarrollado para realizar una sola tarea o para realizar varias según las necesidades de producción. Se puede reprogramar de forma sencilla para que realice una nueva serie de tareas, de forma rápida y sin necesitar la ayuda de un experto en robótica.

Por último, mientras que el robot hace su trabajo solo, el cobot trabaja como ayudante del trabajador y no como sustituto; por lo que no representa pérdidas de fuentes de trabajo.

Son varias las ventajas de trabajar junto a un cobot, ya que son capaces de trabajar hasta 24 horas ininterrumpidas, están preparados para ejercer la fuerza requerida para las tareas más pesadas y realizan labores repetitivas de forma eficiente, dotando al producto de la mejor calidad. Además, permiten a las compañías lograr elevados niveles de productividad (Pelegrí, 2018).

El debate no se centra tanto en si los cobots son mejores que los robots tradicionales, o en si son más fáciles de usar. El desafío estriba en saber identificar cuál es el que mejor se adapta a cada situación. Aunque el cobot, gracias a su versatilidad y a su rápido retorno de la inversión, se indica que es capaz de cumplir con una mayor cantidad de desafíos industriales en el mundo laboral actual (Pelegrí, 2018).

2.9 IDE Universal Robots

Existen principalmente dos enfoques diferentes o formas de programar un Universal-Robot, ya sea con la muy innovadora interfaz gráfica de usuario de Polyscope usando la pantalla táctil de Teaching, o usando el lenguaje de guion UR, o una combinación de ambos enfoques.

La programación de secuencias de comandos en la UR se puede realizar mediante el envío de comandos de secuencias de comandos sin procesar desde un dispositivo externo al robot, incluso sin ningún programa realizado en Polyscope en el robot.

La programación de secuencias de comandos en la UR también se puede realizar agregando comandos de secuencia de comandos a un programa Polyscope que se ejecuta en el propio robot de la UR.

La programación de secuencias de comandos en la UR también se puede realizar como una combinación de tener un programa de servidor ejecutándose en un dispositivo externo, y un programa cliente que se ejecuta localmente en el robot de la UR, que es más elegante y es posible crear un mejor saludo y también es posible para hacer "funciones".

Hay 3 puertos 30001, 30002 y 30003 en la UR que pueden usarse para enviar secuencias de comandos sin formato desde un dispositivo externo. Al crear un escenario Cliente-Servidor, es mejor usar otro número de puerto que no se use en el sistema, por ejemplo, el puerto 12345 o el puerto 30000, etc.

Los comandos de script UR se pueden enviar desde una computadora host o PC a través de una conexión de socket Ethernet TCP directamente al robot UR para control de movimiento y acción sin usar el colgante de enseñanza. Por ejemplo, si el robot es parte de una instalación compleja grande donde todos los equipos se controlan desde la computadora host o PC.

Pero antes de saltar a la programación remota de secuencias de comandos, también hay un método alternativo de programación de secuencias de comandos provisto en la GUI a través del pendiente de enseñanza, que también es muy útil. Con este método es posible importar sentencias de Script o archivos de Script completos (por ejemplo, como "Funciones") en el programa del robot creado por el colgante de enseñanza, por lo que este método es algo intermedio al usar solo los objetos del programa de colgantes de enseñanza y el control remoto. Programación de guiones. (Pelegrí, 2018)

2.10 Protocolos y estándares de transmisión

Los enlaces con señales digitales son los más utilizados actualmente en instalaciones industriales en general. Permiten la construcción de las estructuras de comunicación como estrella, anillo y bus, a las que se conectan los distintos dispositivos de un proceso. Las características físicas del enlace responden a un determinado estándar (RS-232, RS-485, etc.) y, aunque el medio de comunicación suele ser cable de par trenzado, admiten la integración en otras redes de jerarquía superior (LAN, WAN, etc.).

La ausencia de un protocolo aceptado internacionalmente que reemplace las interfaces analógicas de 0/4 a 20 mA por sistemas de transmisión totalmente digitales ha provocado la aparición en el mercado de una multitud de protocolos que compiten entre sí e intentan imponerse como estándares. Las distintas tendencias, en torno a las cuales se agrupan empresas con peso específico propio, están sujetas a intereses económicos y hegemónicos que, sin duda, condicionarán en el futuro las características del tan deseado estándar.

A continuación, se hará una revisión de los estándares de transmisión más conocidos y utilizados como lo es el protocolo RS-232, RS-422 y RS-485, protocolo HART, y los protocolos orientados al control de proceso (Amaya & Rojas, 2016).

2.10.1 Estándares de transmisión digital

2.10.1.1 Estándar RS-232

Las normas RS-232 (Recommended Standard) fueron definidas por la EIA (Electrical Industry Association) en cooperación con la Bell System, los fabricantes de computadores y los fabricantes privados de módems con objeto de normalizar los circuitos de interconexión, llamados circuitos de enlace de interfaz entre el equipo terminal de datos (ETD) y el equipo terminal de comunicación de datos (ETCD).

Una arquitectura típica de enlace que pone de manifiesto esta terminología sería aquella constituida por un ordenador (ETD) conectado a un modem (ETCD), y este último conectado a la línea telefónica.

Las normas RS-232 cubren los tres aspectos siguientes de la comunicación entre el ETD y el ETCD: características eléctricas de las señales, características mecánicas de los conectores y descripción funcional de las señales usadas para realizar la comunicación (Amaya & Rojas, 2016).

2.10.1.2 Aspectos mecánicos y eléctricos

El conector utilizado es el DB-25 de veinticinco terminales, aunque nueve de sus líneas son más que suficientes en una comunicación bidireccional. Las características eléctricas de las señales puestas en juego son las siguientes:

a) Señal single ended (una línea para transmisión y otra para recepción, referidas a una única línea de masa).
b) En el transmisor el 1 lógico es transmitido mediante una tensión de línea comprendida entre –5 y –15V y el 0 lógico se transmite mediante una tensión entre +5 V y +15 V.
c) En el receptor se entenderá como 1 lógico toda aquella tensión comprendida entre –3V y – 15V, y como 0 lógico los niveles de tensión comprendidos entre +3V y +15V.

d) La velocidad de cambio (slew rate) marcado en la norma específica con un máximo de 30 V/μ s.
e) Todos los dispositivos transmisores y receptores deben aguantar una tensión de 25V.
f) Sólo un transmisor y un receptor por línea de comunicación.
g) Sensibilidad del receptor: 3V.
h) La impedancia de carga del transmisor está comprendida entre 3 kΩ y 7 kΩ, que es la que presenta el equipo receptor.
i) Corriente de cortocircuito inferior a 0,5 A.

La norma RS-232D redefine la RS-232C y, aunque no indica nada con respecto a la longitud del enlace, fija la capacidad máxima de la linea en 2.500 pF. Esta capacidad se obtiene con sesenta y nueve metros de par trenzado (≈ 36 pF/m), veintisiete metros de par trenzado múltiple (≈ 69 pF/m) o veinticinco metros de cable coaxial (≈ 100 pF/m).

En una transferencia síncrona de datos se precisa la existencia de una línea adicional por donde se transmitan los pulsos de reloj con objeto de interpretar la información a partir de los diferentes pulsos que se presentan en las líneas de datos. Una de las ventajas que presenta este modo de funcionamiento es que el receptor se adapta automáticamente a cualquiera de las frecuencias de reloj que proporcione el transmisor.

En las transmisiones asíncronas los datos de sincronismo son inherentes a la información transmitida, no siendo necesario ningún tipo de señal de reloj o línea adicional para tal fin.

El formato de un byte de información transmitido en serie contiene fundamentalmente un bit que indica el inicio de la transmisión (*start*), seguido de un número determinado de bits en serie que componen la información (*data*), para finalizar con un bit (*stop*) que indica la finalización de ésta. Cabe la posibilidad de

introducir, de manera opcional, un bit mas (*parity*), que será utilizado como detector de error en la paridad de la información comunicada (Amaya & Rojas, 2016).

2.10.1.3 Aspectos funcionales

Dentro del conjunto de las señales podemos distinguir cuatro grandes grupos: de datos, de control, de temporización y las masas. En la siguiente figura 2-2 se detalla el número de terminal del conector, el nombre de la señal, el sentido de conexión entre ETD y el ETCD y una breve descripción de su función (Amaya & Rojas, 2016).

Núm. de terminal DB-9	DB-25	Nombre	Dirección	Función
1	8	DCD	Entrada ETD Salida ETCD	(*Data Carrier Detect*) El ETCD informa al ETD de que ha detectado portadora.
2	3	RxD	Entrada ETD Salida ETCD	Terminal de recepción de datos.
3	2	TxD	Entrada ETCD Salida ETD	Terminal de transmisión de datos.
4	20	DTR	Entrada ETCD Salida ETD	(*Data Terminal Ready*) Terminal de datos preparado.
5	7	GND	ETD - ETCD	Línea común de referencia (masa).
6	6	DSR	Entrada ETD Salida ETCD	(*Data Set Ready*) Lo activa el ETCD para indicar que ha marcado número.
7	4	RTS	Entrada ETCD Salida ETD	(*Request To Send*) El ETD activa la línea cuando desea enviar información.
8	5	CTS	Entrada ETD Salida ETCD	(*Clear To Send*) El ETCD activa la línea cuando está preparado para recibir información.
9	22	RI	Entrada ETD Salida ETCD	(*Ring Indicator*) El ETCD informa al ETD de que ha detectado llamada.

Figura 2-2 Descripción de la función del conector

2.10.2 Estándar RS-422

Cuando se requieren velocidades mayores de transmisión que las que ofrece la norma RS-232 C es necesario utilizar un sistema de transmisión diferencial, para

evitar los efectos del ruido que aparecen con tensiones en modo común en las salidas del emisor o a la entrada del receptor.

La norma RS-422 A, utiliza señales simétricas diferenciales y alcanza velocidades de hasta 10 Mbit/s. Le afectan menos las interferencias electromagnéticas, le influyen menos las caídas de tensión y presenta una mayor inmunidad a tensiones en modo común aplicadas a la línea de enlace.

Los dispositivos emisores que cumplen esta norma son capaces de transmitir señales diferenciales con un mínimo de 2V sobre un par de hilos trenzados terminados con una impedancia de 100Ω.

La ventaja de esta norma sobre la RS-232 C es que en aplicaciones de bus permite que un solo emisor pueda comunicar con varios receptores, aunque presenta la limitación de que los restantes receptores deben estar en estado de alta impedancia para no cargar el bus. Admite un transmisor (maestro) y hasta diez receptores (esclavos). Las características más significativas son las siguientes:

a. Modo de operación: diferencial.
b. Longitud máxima del enlace: 1.200 m.
c. Velocidad máxima: 10 Mbit/s.
d. Número máximo de dispositivos permitidos: un transmisor y diez receptores.
e. e) Resistencia mínima de carga del transmisor: 100Ω.
f. g) Resistencia mínima de entrada del receptor: 4kΩ (Amaya & Rojas, 2016).

2.10.2.1 Estándar RS-485

Esta norma sólo define características eléctricas. Derivada del RS-422, también utiliza señales diferenciales y permite la comunicación half duplex de hasta 32 dispositivos maestros y 32 dispositivos esclavos.

Se considera como una interfaz multipunto que permite la comunicación de hasta treinta y dos pares de emisores-receptores (transceptores) en un bus de datos común y que al mismo tiempo satisface los requerimientos de la norma.

2.10.2.2 Características

a. Modo de operación: diferencial.
b. Longitud: 1.200 m.
c. Velocidad máxima: 10 Mbit/s.
d. Número máximo de dispositivos permitidos: treinta y dos transmisores y treinta y dos receptores (o treinta y dos transceptores).
e. Señal de salida del transmisor: entre ±1,5 V y ±5 V (tensión negativa: 1; tensión positiva: 0).
f. Resistencia mínima de carga del transmisor: 54 Ω.
g. Señal de entrada al receptor: –7 V a 12 V.
h. Sensibilidad del receptor: ± 200mV.
i. Resistencia mínima de entrada del receptor: 12kΩ.

La velocidad de comunicación está estrechamente ligada a la longitud del enlace. Así pues, se tiene que para distancias próximas a los mil metros la velocidad efectiva está por debajo de los 100 kbit/s; para distancias de en torno a doscientos metros la velocidad es de 500 kbit/s y para distancias por debajo de los diez metros se consigue la mayor velocidad (≈10 Mbit/s).

Características	RS-232C	RS-422A	RS-485
Número máximo de dispositivos	Un transmisor Un receptor	Un transmisor Diez receptores	Treinta y dos transmisores Treinta y dos receptores
Tipo de señal	Simple con respecto a masa	Diferencial	Diferencial
Longitud	15 m	1.200 m	1.200 m

Características	RS-232C	RS-422A	RS-485
Número de hilos	3	4	2
Velocidad	19,2 kbits/s	10 Mbits/s	10 Mbits/s
Comunicación	*Duplex*	*Duplex*	*Half-duplex*
Rango de la señal de entrada al receptor	−15V...+15V	−7V...+7V	−7V...+12V
Sensibilidad de entrada del receptor	± 3V	± 200 mV	± 200 mV

Figura 2-3 Características estándar 485

2.10.3 Protocolos orientados a la configuración

2.10.3.1 Protocolo HART

HART (Highway Adressable Remote Transducter) es desarrollado por Resemount, en los años 80, como protocolo abierto, formando un grupo de usuarios en 1990. En 1983 se crea HART Communication Foundation, con la finalidad de mantener la propiedad de la tecnología, gestionar los estandares y asegurar así la accesibilidad de la tecnología a todos los sectores industriales.

Se trata de un protocolo muy difundido en la industria de procesos. Los fabricantes de módulos de este tipo se agrupan en el HART User Group, garantizando el soporte técnico gracias a la HART Communication Foundation.

Pretende reemplazar el captador clásico de 4-20mA por un captador inteligente, minimizando las modificaciones de cableado. Mediante la técnica de modulación FSK (Frequency Shift Keying), se superpone una señal de datos a la señal medida, de 4 a 20mA (Amaya & Rojas, 2016).

2.10.3.2 Comunicación tipo Punto a Punto

En el modo Punto a Punto, la señal tradicional de 4-20 mA es usada para comunicar una variable de proceso mientras otras variables adicionales parámetros de configuración y otras informaciones de aparato son transmitidas digitalmente usando el protocolo HART. La señal análoga de 4-20 mA no es afectada por la señal HART y puede ser usada para el monitoreo o control en la forma normal. La señal de comunicación digital HART le da acceso a variables secundarias y a otras informaciones, que pueden ser usadas para propósitos de operación, mantención y diagnóstico, ver Figura 2-4.

2.10.3.3 Comunicación tipo Multipunto (Multidrop)

El modo Multipunto requiere solamente un par de alambres y si es aplicable, el lazo también puede tener barreras de seguridad y fuentes de poder auxiliares para hasta 15 aparatos de terreno. Todos los valores de proceso son transmitidos digitalmente; en el modo Multipunto, las direcciones de "Polling" de los aparatos de terreno son mayores que 0 y la corriente a través de cada equipo está sujeta a un mínimo valor (típicamente 4mA).

Los dispositivos basados en protocolo HART son los únicos capaces de soportar comunicación analógica 4-20mA y la digital en el mismo cable, lo cual permite utilizar los dos canales de forma simultánea para verificar la integridad de los lazos de control y permitir mantenimiento preventivo en procesos delicados.

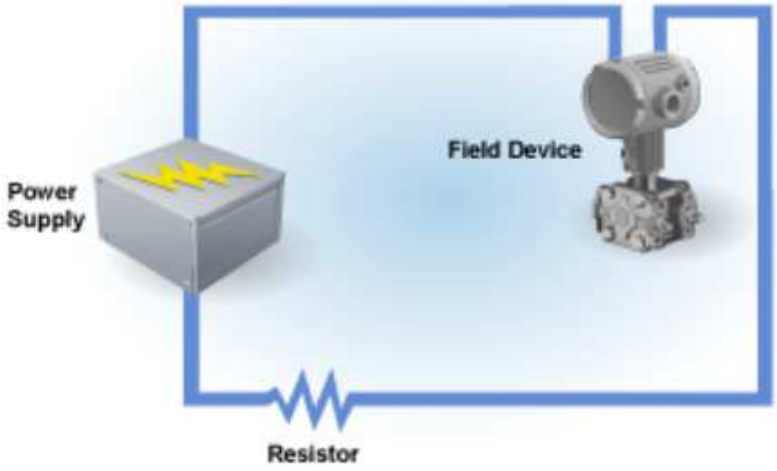


Figura 2-4 Hart, comunicación punto a punto

2.10.4 Protocolo orientado al control de proceso

2.10.4.1 AS-i

El AS-Interface o Interfaz de Actuador/Sensor fue creado en el año 1994 para la sustitución de la gran cantidad de señales provenientes de los sensores y dirigidos hacia los actuadores desde el controlador.

El AS-Interface, también conocido de forma abreviada como bus AS-i, es un sistema de enlace para el nivel más bajo de procesos en instalaciones de automatización.

Los mazos de cables utilizados hasta ahora en este nivel son reemplazados por un único cable eléctrico, el cable AS-i. Por medio del cable AS-i y del maestro AS-i se acoplan sensores y actuadores binarios de la categoría más simple a las unidades de control a través de módulos AS-i en el nivel de campo.

El AS-Interface presenta varias características fundamentales, como son:

• Idóneo para la conexión de actuadores y sensores binarios. A través del cable AS-i tienen lugar tanto el intercambio de datos entre sensores/ actuadores (esclavos AS-i) y el maestro AS-i como la alimentación eléctrica de los sensores y los actuadores.

• Cableado sencillo y económico; montaje fácil con técnica de perforación de aislamiento; gran flexibilidad gracias al cableado tipo árbol.

• Reacción rápida: el maestro AS-i necesita como máximo 5 ms para el intercambio de datos cíclico con hasta 31 estaciones conectadas.

• Las estaciones (esclavos AS-i) conectadas al cable AS-i pueden ser sensores/ actuadores con conexión AS-i integrada o módulos AS-i, a cada uno de los cuales se pueden conectar hasta ocho sensores/actuadores binarios convencionales.

• Con módulos AS-i estándar pueden funcionar hasta 124 actuadores y 124 sensores conectados al cable AS-i.

• Si se utilizan módulos AS-i con un espacio de direcciones ampliado, es posible la operación de hasta 186 actuadores y 248 sensores con un maestro extendido.

• Maestros AS-i de SIMATIC NET extendidos soportan una posibilidad de acceso especialmente sencilla a sensores/actuadores analógicos o a módulos que trabajen segun el perfil de esclavo AS-Interface 7.3/7.4 (Amaya & Rojas, 2016).

2.10.4.2 PROFIBUS

Profibus es uno de los buses con mayor implantación tanto a nivel europeo como mundial, y ha sido desarrollado sobre la base del modelo ISO/OSI (International Standard Organizaron / Open System Interconnect) para servicio de comunicación de datos.

Existen tres perfiles distintos de Profibus, como son:

• Profibus FMS (*Fieldbus Message Specification*): Está implementado en el nivel 7 del modelo OSI, su aplicación es la transferencia de gran volumen de datos entre diferentes dispositivos inteligentes conectados en una misma red. Hoy en día, con el uso creciente de Ethernet y TCP/IP va relegando este perfil a un segundo plano. El sistema está basado en una estructura Cliente- Servidor.

• Profibus DP (*Distributed Peripheral*): Su aplicación está basada en el intercambio a gran velocidad de un volumen medio de información entre un controlador, que hace las funciones de maestro, y diferentes controladores o diferentes periféricos, como son autómatas programables, módulos de E/S, convertidores de frecuencia, paneles de visualización, etc., que actúan como dispositivos esclavos, distribuidos por el proceso y conectados a una misma red de comunicación. El Profibus DP trabaja dentro de los niveles 1 y 2 del modelo OSI y bajo las especificaciones de la norma física RS-485 (Amaya & Rojas, 2016).

2.10.4.3 PROFINET

El PROFINET es un perfil moderno pensado para acercar ciertas funcionalidades de la automatización al nivel de dirección en las industrias.

Basado en Ethernet (IEEE 802.3 e ISO 8802.3), con topología de bus y protocolo de acceso al medio CSMA/CD (*carrier sense múltiple access/collision detection*), permite la integración de buses de campo (en particular PROFIBUS) de forma simple y sin realizar modificación alguna. De esta manera, las técnicas regularizadas y establecidas por IT (information technology) en el área de la ofimática; también pueden ser usadas en el mundo de la automatización, permitiendo enlazar el nivel de planificación de recursos de la empresa con el nivel de producción y el nivel de campo. Los objetivos de PROFINET son:

• Ser un estándar abierto para la automatización basado en Industrial Ethernet.

• Que los componentes de Industrial Ethernet y Standard Ethernet puedan utilizarse conjuntamente, aunque los equipos de Industrial Ethernet sean más robustos y, por consiguiente, más apropiados para el entorno industrial.

• Usar estándares TCP/IP y de tecnologías de la información.

• Automatización con Ethernet en tiempo real.

• Integrar de forma directa sistemas con bus de campo. El PROFINET especifica las funciones para la realización de una solución total de automatización desde la instalación de la red hasta el diagnóstico basado en la web. Gracias a su estructura modular, PROFINET puede ampliarse fácilmente mediante la implementación de funciones en el futuro.

Dado lo anterior, podemos deducir las siguientes ventajas:

- Flexibilidad gracias al empleo de Ethernet y de los acreditados estándares IT.
- Ahorro de ingeniería.
- Protección de la inversión para equipos y aplicaciones Profibus.
- Más rápido que los actuales buses especiales en el ámbito de Motion Control.
- Amplio abanico de productos disponibles en el mercado (Amaya & Rojas, 2016).

2.10.4.4 MODBUS

Este es un protocolo desarrollado por Modicon en 1979, utilizado para establecer comunicaciones maestro-esclavo y Cliente-Servidor entre dispositivos inteligentes y con dispositivos de campo.

Entre sus funciones están transmitir señales digitales, analógicas y registros entre ellos, o monitorizar dispositivos de campo.

Es un protocolo ideal para la monitorización remota vía radio de elementos de campo (*RTU, remote terminal unit*). Actualmente, está implementándose en sectores ajenos a su idea original, tales como la domótica o el control de procesos (climatización, control de procesos, bombeos, etc.).

Define una estructura de mensaje que los controladores podrán reconocer y utilizar sin tener en cuenta el tipo de red que éstos utilizarán para comunicarse. Durante las comunicaciones llevadas a cabo en una red Modbus, el protocolo determina como cada controlador reconocerá las direcciones, si un mensaje está dirigido a él, determinar la acción a llevar a cabo y extraer los datos del mensaje. De la misma manera, se define el protocolo y acciones de respuesta.

Los controladores de una red pueden comunicarse mediante la red punto a punto, siendo cualquiera de éstos el que puede iniciar el diálogo con los otros controladores. De esta manera, un controlador puede funcionar como Maestro o Esclavo en comunicaciones independientes (Amaya & Rojas, 2016).

2.10.5 PROFIBUS PA

Profibus Process Automation es un prototipo ampliado del Profibus DP, diseñado para trabajar en los ámbitos de control de procesos, es decir, en zonas denominadas "Ex" de seguridad intrínseca.

Este perfil sigue lo enunciado en la norma IEC 1158-2, es decir, posibilita la conexión de dispositivos de campo en zonas con riesgo de explosión en las que se requiere una red intrínsecamente segura y donde la alimentación de los dispositivos se realiza a través de la propia línea de enlace. En este caso, el acoplo a la red DP se realiza a través de acopladores de segmento (DP/PA link).

Este perfil está específicamente diseñado para procesos de automatización en los que se enlazan, con un tiempo de ciclo de bus de unos pocos ms, los distintos dispositivos de campo con PLC o controladores, describiendo las funciones normalizadas de la aplicación mediante bloques de función, y puede ser integrado en el mismo bus conjuntamente con FMS y DP cuando implementa su capa física con EIA RS485 (Amaya & Rojas, 2016).

2.10.6 FIELDBUS FOUNDATION

Es un protocolo de comunicación digital para redes industriales, específicamente utilizado en aplicaciones de control distribuido. Puede comunicar grandes volúmenes de información, ideal para aplicaciones con varios lazos

complejos de control de procesos y automatización. Está orientado principalmente a la interconexión de dispositivos en industrias de proceso continuo.

Los dispositivos de campo son alimentados a través del bus Fieldbus cuando la potencia requerida para el funcionamiento lo permite.

La Fieldbus Foundation es la organización dedicada a la consecución de unas especificaciones orientadas a crear un bus de campo único y abierto, así como elementos de hardware y software para las compañías que lo quieran integrar en sus productos.

La Fieldbus Foundation ha orientado las tecnologías de comunicación de forma que puedan soportar aplicaciones críticas donde la transferencia de datos y el manejo de información son esenciales. Es el único protocolo de bus de campo digital desarrollado para el cumplimiento de las especificaciones SP50, de ISA. De la misma forma, es el único protocolo que soporta los requerimientos de la zona de seguridad intrínseca, áreas peligrosas, procesos con riesgo de explosión y ambientes de regulación difíciles.

Puede comunicar grandes volúmenes de información, ideal para aplicaciones con varios lazos complejos de control de procesos y automatización de la fabricación (HSE, High Speed Ethernet).

El protocolo proporciona herramientas dedicas de control y comunicación para la ejecución periódica y precisa de las funciones de control, eliminando los tiempos muertos y demás problemas que provocan las comunicaciones.

La comunicación en las plantas industriales se ha hecho imprescindible en la industria actual. Muchos sistemas están conformados por equipos de diferentes fabricantes y funciones en diferentes niveles de automatización. Pese a que pueden estar distanciados entre sí, se desea que trabajen en forma coordinada para un

resultado satisfactorio del proceso. El objetivo principal es la comunicación totalmente integrada en el sistema.

Esto reporta la máxima flexibilidad y permite integrar sin problemas productos de otros fabricantes a través de los interfaces de software estandarizados. Las aplicaciones industriales basadas en comunicación digital se han incrementado haciendo posible la conexión entre sensores, actuadores y equipos de control en una planta de procesamiento.

La comunicación digital debe integrar la información provista por los elementos de campo en el sistema de control de procesos. En este sentido, el objetivo de un bus de campo es sustituir las conexiones punto a punto entre los elementos de campo y el equipo de control a través del tradicional lazo de corriente de 4-20mA o 0 a 10VDC, según corresponda.

Generalmente son redes digitales, bidireccionales, multipunto, montadas sobre un bus serie, que conectan dispositivos de campo como PLCs, transductores, actuadores, sensores y equipos de supervisión. No obstante, hasta la fecha no existe un bus de campo universal.

CAPÍTULO 3. ANÁLISIS Y DISEÑO

3.1 Introducción

En el diseño orientado a objetos, se definen los atributos externos de un objeto y los métodos que exporta. El resto de la información queda oculta. Además, se pueden especificar relaciones de herencia y de polimorfismo. El diseño orientado a objetos está directamente ligado, aunque no es obligatorio, con los lenguajes de programación orientada a objetos. Usando estas técnicas se puede conseguir crear clases que tengan características fundamentales como: la abstracción, la encapsulación, la ocultación de la información y la modularidad.

3.2 Actores

Un actor describe el rol que desempeña un usuario externo al sistema durante una interacción con el sistema.

Se distinguen dos categorías de actores:

- Los actores primarios, para los cuales el objetivo del caso de uso es esencial y constituye un objetivo del actor.
- Los actores secundarios, para los que el objetivo del caso de uso no es esencial, si bien interactúan con él (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004).

3.3 Perfil de los actores

a. **Robot**: Es el responsable de interactuar con el producto y lleva acabo la carga y descarga del producto.
b. **Estación**: Es donde se realiza las pruebas al producto (prueba eléctrica, ensamblado y flash), donde al finalizar manda un estatus de pasa o falla.

c. **Usuario**: Es el responsable de llevar a cabo la configuración de las metas establecidas por equipos, y llenar los catálogos de líneas y proceso.

d. Destinatario (jefe de grupo/Supervisor): Es el responsable de llevar a cabo el llenado del formulario donde se describe el problema del no cumplimiento de la meta establecida.

3.4 Requerimientos funcionales

En esta sección del capítulo 3 se describen los requerimientos funcionales que constituyen el prototipo.

3.4.1 Módulo de Tipos de Equipo

Tabla 3.1 Requerimientos funcionales de Tipos de Equipo

Identificador (ID)	Función	Categoría
ReqTipo1.1	La aplicación deberá registrar los tipos de estación para los diferentes equipos.	Evidente
ReqTipo 1.2	La aplicación deberá editar los tipos de estación para los diferentes equipos.	Evidente
ReqTipo 1.3	La aplicación deberá borrar los tipos de estación para los diferentes equipos.	Evidente

3.4.2 Módulo de Líneas

Tabla 3.2 Requerimientos funcionales de Líneas

Identificador (ID)	Función	Categoría
RLíneas1.1	La aplicación deberá registrar las líneas con que cuenta la fábrica de manufactura.	Evidente
RLíneas1.2	La aplicación deberá editar las líneas con que cuenta la fábrica de manufactura.	Evidente
RLíneas1.3	La aplicación deberá borrar las líneas con que cuenta la fábrica de manufactura.	Evidente

3.4.3 Módulo de Equipos

Tabla 3.3 Requerimientos funcionales de Equipo

Identificador (ID)	Función	Categoría
REquip1.1	La aplicación deberá registrar los equipos con los que cuenta la línea de producción o celda	Evidente
REquip1.2	La aplicación deberá editar los equipos con los que cuenta la línea de producción o celda	Evidente
REquip1.3	La aplicación deberá borrar los equipos con los que cuenta la línea de producción o celda	Evidente

3.4.4 Módulo de Matriz

Tabla 3.4 Requerimientos funcionales de la Matriz

Identificador (ID)	Función	Categoría
RMatriz1.1	La aplicación deberá registrar la secuencia como va operar la matriz de estados por equipo y el tiempo estimado de operación.	Evidente
RMatriz1.2	La aplicación deberá editar la secuencia de operación de la matriz de estados por equipo y el tiempo estimado de operación.	Evidente
RMatriz1.3	La aplicación deberá borrar la secuencia de operación de la matriz de estados por equipo y el tiempo estimado de operación.	Evidente

3.4.5

3.4.6 Módulo de Resultados de Equipos

Tabla 3.5 Requerimientos funcionales de Resultados de Equipos

Identificador (ID)	Función	Categoría
REquipResult1.1	Los equipos enviarán el estatus y tiempo de la operación de cada equipo a la aplicación, donde se determinará el siguiente paso.	Oculto
REquipResult1.2	La aplicación realizará la sumatoria total de los valores de disponibilidad, rendimiento y calidad por equipo, donde se compararán respecto a los valores predefinidos en el catálogo de equipos.	Oculto
REquipResult1.3	La aplicación realizara la actualización del cálculo de la disponibilidad, rendimiento y calidad en tiempo real.	Evidente
REquipResult1.4	La aplicación realizará el cálculo del promedio, valor máximo, valor mínimo, cantidad de cargas y desviación estándar por equipo del manejo del producto (carga y descarga)	Oculto
REquipResult1.5	El sistema generará cada hora un reporte evaluando los valores obtenidos respecto los valores predefinidos en el catálogo de equipos de la disponibilidad, rendimiento y calidad.	Evidente
REquipResult1.6	En caso de no cumplir con los valores predefinidos del catálogo de equipos, el sistema deberá enviar un correo el cual contiene un URL al jefe de línea, supervisor y Gerente, donde se llenaran los comentarios de lo ocurrido en ese lapso de tiempo.	Evidente
REquipResult1.7	En un formulario se deberá describir el problema del no cumplimiento de la metas establecida y lo deberá llenar el jefe de la línea de producción.	Evidente

3.5 Casos de uso de alto nivel

En esta sección del capítulo 3 se describen los casos de uso de alto nivel que constituyen el prototipo.

3.5.1 Casos de uso del módulo tipo de equipos

Tabla 3.6 Caso de uso Registro de tipo de equipos

Caso de uso:	Caso de uso registro de tipo de equipos
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	El usuario registra y define el tipo de estación con las que cuenta la línea de producción, como lo pueden ser: ICT, Flash, Atornillado.
Requerimiento:	ReqTipo1.1

Tabla 3.7 Caso de uso edición de tipo de equipos

Caso de uso:	Caso de uso edición de tipo de equipos
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario consultar y editar el registro del tipo de equipo de la estación con las que cuenta la línea de producción, como lo pueden ser: ICT, Flash, Atornillado.
Requerimiento:	ReqTipo1.2

Tabla 3.8 Caso de uso borrar tipo de equipos

Caso de uso:	Caso de uso borrar tipo de equipos
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	El usuario borra el registro del tipo de equipo de la estación con las que cuenta la línea de producción.
Requerimiento:	ReqTipo1.3

3.5.2 Casos de uso del módulo líneas

Tabla 3.9 Caso de uso de registro de Líneas

Caso de uso:	Caso de uso de registro de Líneas
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	El usuario registra y define el nombre de la línea donde se encuentran los equipos.
Requerimiento:	ReqLineas1.1

Tabla 3.10 Caso de uso de edición Líneas

Caso de uso:	Caso de uso de edición Líneas
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario consultar y editar el registro del nombre de la línea donde se encuentran los equipos.
Requerimiento:	ReqLineas1.2

Tabla 3.11 Caso de uso borrar Líneas

Caso de uso:	Caso de uso borrar Líneas
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	El usuario borra el registro de la línea donde se encuentran los equipos.
Requerimiento:	RequLineas1.3

3.5.3 Casos de uso del módulo de Equipos

Tabla 3.12 Caso de uso de registro de Equipo

Caso de uso:	Caso de uso de registro de Equipo
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario registrar el nombre, el tipo del equipo y la línea a la que pertenece, del mismo modo se configura el tiempo de disponibilidad y la unidad deseada por hora.
Requerimiento:	REquip1.1

Tabla 3.13 Caso de uso de edición de Equipo

Caso de uso:	Caso de uso de edición de Equipo
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario consultar y editar el registro del nombre del equipo y del mismo modo se configura el tiempo de disponibilidad y las unidades deseadas por hora.
Requerimiento:	REquip1.2

Tabla 3.14 Caso de uso de borrar Equipo

Caso de uso:	Caso de uso de borrar Equipo
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	El usuario borra el registro del equipo.
Requerimiento:	REquip1.2

3.5.4 Casos de uso del módulo de Matriz

Tabla 3.15 Caso de uso de registro Matriz

Caso de uso:	Caso de uso de registro Matriz
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario registrar definición de la secuencia de la operación del robot y del equipo, colocando el tiempo estimado de cada operación.
Requerimiento:	RMatrix1.1

Tabla 3.16 Caso de uso de edición Matriz

Caso de uso:	Caso de uso de edición Matriz
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permite al usuario consultar y editar el registro y redefinir la secuencia de la operación del robot.
Requerimiento:	RMatrix1.2

Tabla 3.17 Caso de uso de borrar Matriz

Caso de uso:	Caso de uso de borrar Matriz
Actores:	Usuario, Aplicación
Tipo:	Primario
Descripción:	La aplicación permitirá al usuario borra el registro de la matriz de estado alterando la secuencia de la operación del robot.
Requerimiento:	RMatrix1.3

3.5.5 Casos de uso del módulo de Resultado de equipos

Tabla 3.18 Caso de uso de envío de estatus (Paso/Fallo)

Caso de uso:	Caso de uso de envió de estatus (Paso/Fallo)
Actores:	Estación, Aplicación
Tipo:	Primario
Descripción:	La estación manda el estatus de paso o fallo por medio de comunicación modbus a la aplicación y esta lo registra y dependiendo del status se determina la siguiente operación y muestra el resultado.
Requerimiento:	REquipResult1.1

Tabla 3.19 Caso de uso de registro de tiempo

Caso de uso:	Caso de uso de registro de tiempo
Actores:	Estación, Aplicación
Tipo:	Primario
Descripción:	La aplicación registra el tiempo carga/descarga por medio de comunicación modbus y del mismo modo se realiza la sumatoria de valores de disponibilidad, rendimiento y calidad, en tiempo real y los valores son evaluados y comparados con respecto a los valores predefinidos en el catálogo de equipos.
Requerimiento:	REquipResult1.2

Tabla 3.20 Caso de uso de cálculo de los valores disponibilidad, rendimiento y calidad (Tiempo Real)

Caso de uso:	Caso de uso de cálculo de los valores disponibilidad, rendimiento y calidad (Tiempo Real)
Actores:	Aplicación
Tipo:	Primario
Descripción:	La aplicación calcula los valores disponibilidad, rendimiento y calidad y los muestra en pantalla.
Requerimiento:	REquipResult1.3

Tabla 3.21 Caso de uso de despliegue de valores estadísticos

Caso de uso:	Caso de uso de despliegue de valores estadísticos
Actores:	Aplicación
Tipo:	Primario
Descripción:	La aplicación calcula el manejo de producto (carga y descarga) obteniendo valores como el promedio, el valor máximo, el valor mínimo, cantidad de cargas y descargas, desviación estándar.
Requerimiento:	REquipResult1.4

Tabla 3.22 Caso de uso de Reporte

Caso de uso:	Caso de uso de Reporte
Actores:	Aplicación
Tipo:	Primario
Descripción:	La aplicación genera un reporte cada hora donde se evalúan la disponibilidad, rendimiento y calidad respecto con los valores predefinidos en el catálogo de equipos, del mismo modo envía un correo electrónico con la información al jefe de grupo, supervisor y gerente.
Requerimiento:	REquipResult1.5

Tabla 3.23 Caso de uso de No cumplimiento

Caso de uso:	Caso de uso de no cumplimiento
Actores:	Aplicación
Tipo:	Primario
Descripción:	La aplicación detecta el no cumplimento de los valores predefinidos por lo cual enviará un correo electrónico con un URL al jefe de línea, donde se le solicitará especificar la razón del no cumplimento de las metas.
Requerimiento:	REquipResult1.6

Tabla 3.24 Caso de uso de No cumplimiento formulario

Caso de uso:	Caso de uso de no cumplimiento formulario
Actores:	Usuario
Tipo:	Primario
Descripción:	La aplicación permitirá al usuario introducir en un formulario la descripción del problema y del no cumplimiento de los valores predefinidos anteriormente en el catálogo de quipos.
Requerimiento:	REquipResult1.7

3.6 Diagrama de casos de uso

En la Figura 3-1 se muestra en el diagrama de casos de uso el registro, la edición, y borrar el tipo de equipo. Teniendo el usuario del sistema estas capacidades.

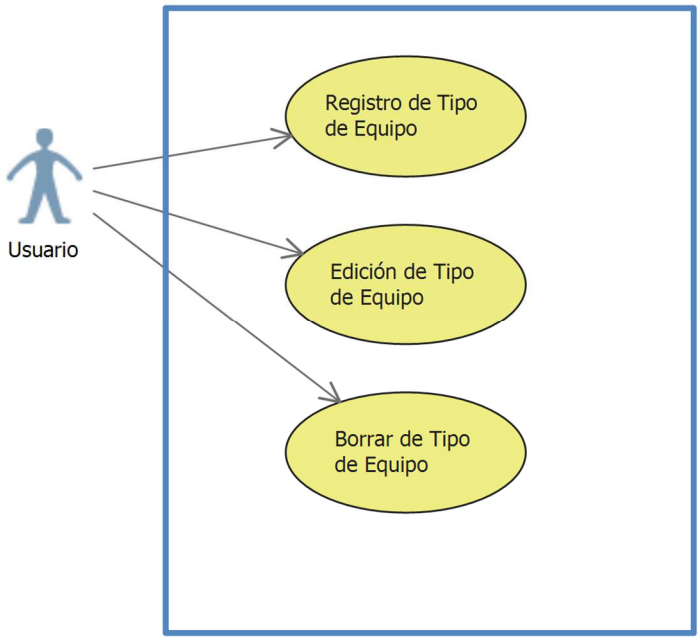


Figura 3-1 Diagrama de casos de uso del módulo de tipo de equipos

3.6.1 Diagrama de casos de uso del módulo líneas

En la Figura 3-2 se muestra en el diagrama de casos de uso el registro, la edición, y borrar de líneas. Teniendo el usuario del sistema estas capacidades.

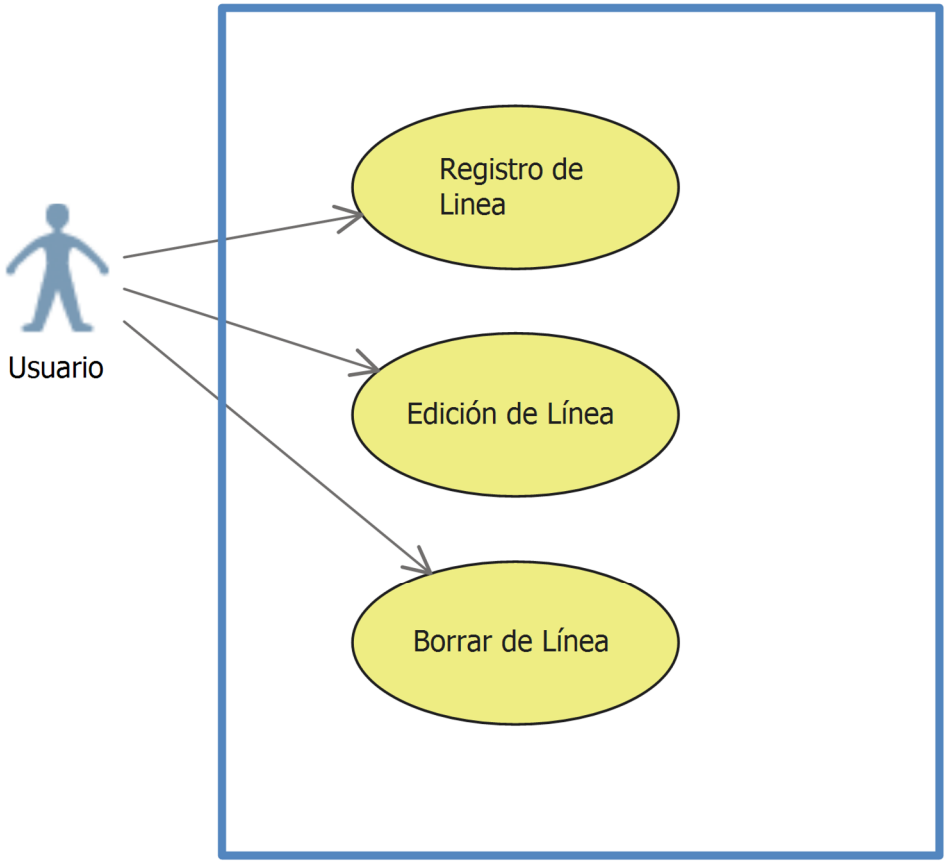


Figura 3-2 Diagrama de casos de uso del módulo líneas

3.6.2 Diagrama de casos de uso del módulo de equipos

En la Figura 3-3 se muestra en el diagrama de casos de uso el registro, la edición, y borrar de equipo. Teniendo el usuario del sistema estas capacidades.

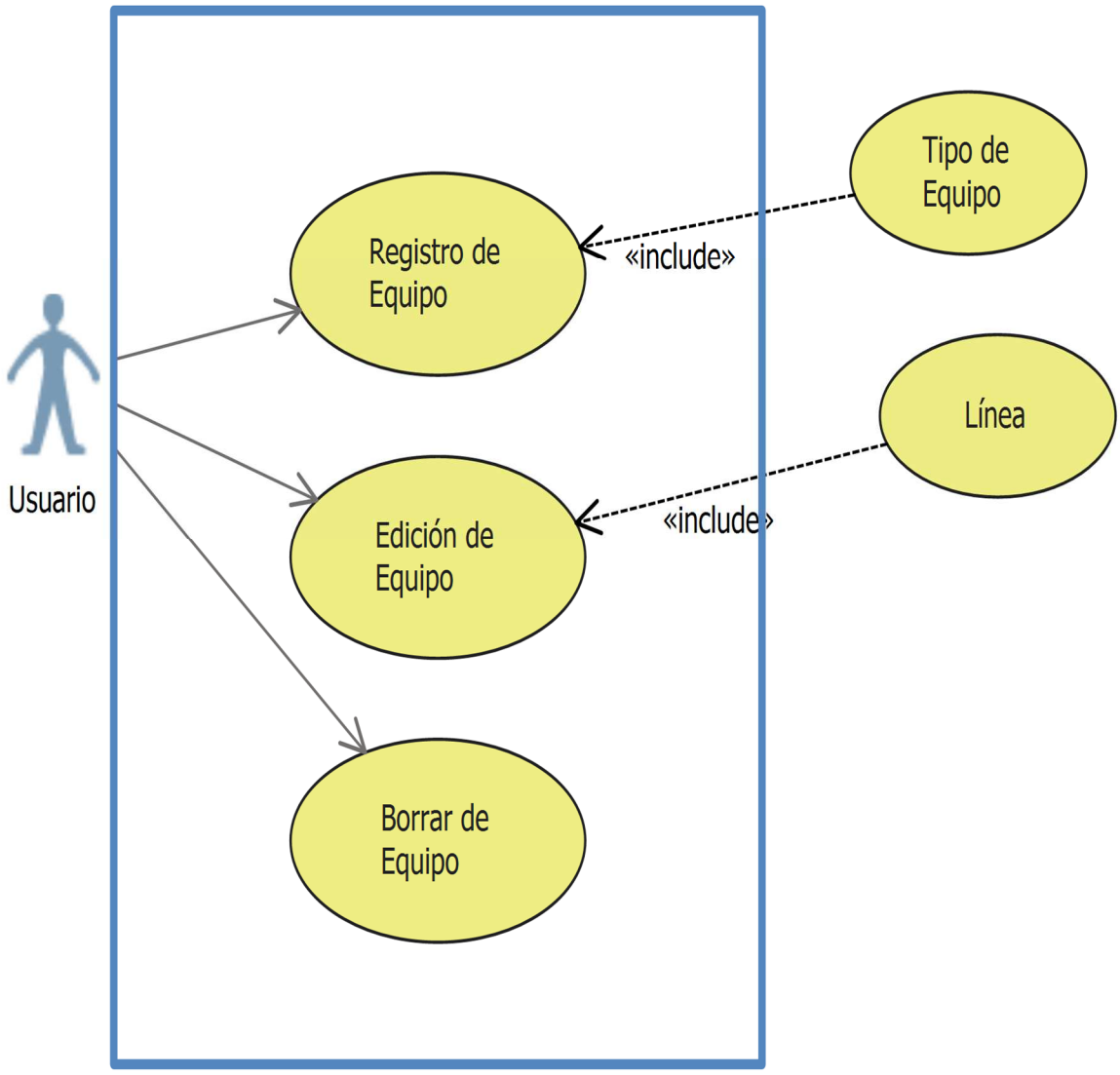


Figura 3-3 Diagrama de casos de uso del módulo de equipos

3.6.3 Diagrama de casos de uso del módulo matriz

En la Figura 3-4 se muestra en el diagrama de casos de uso el registro, la edición, y borrar de matriz. Teniendo el usuario del sistema estas capacidades.

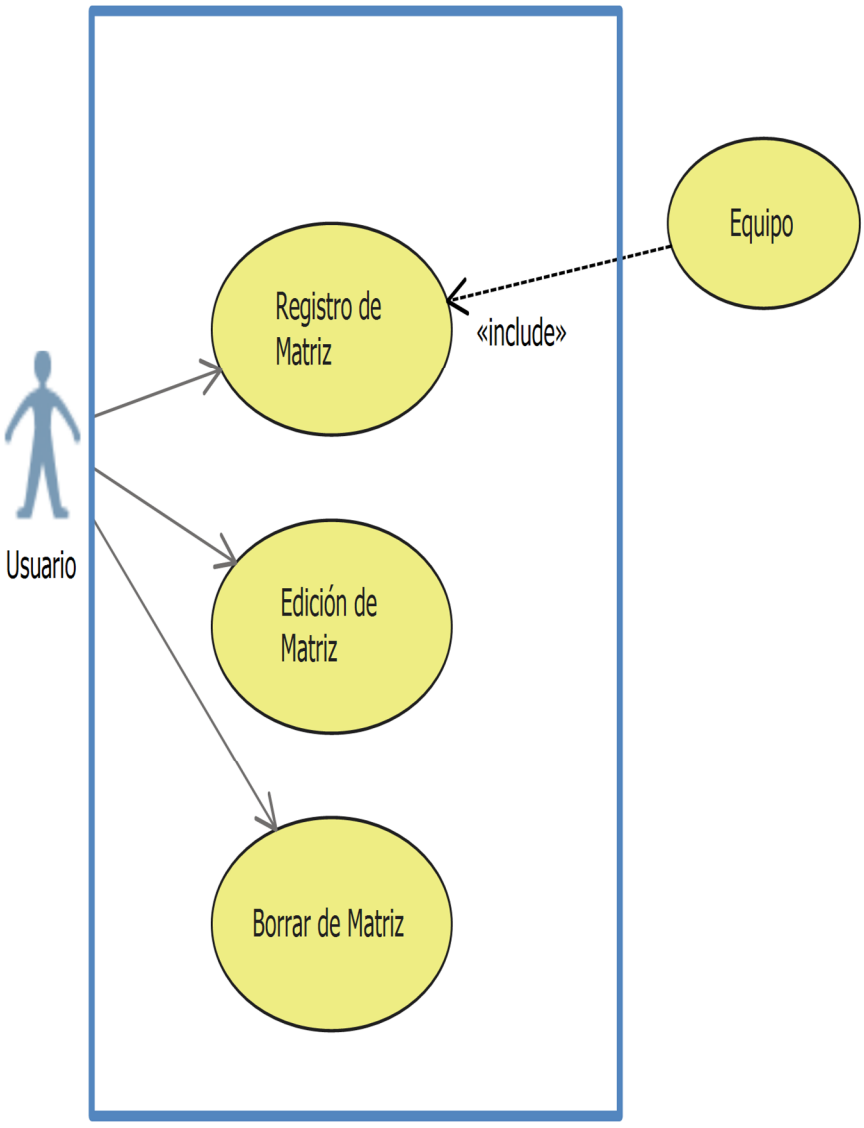


Figura 3-4 Diagrama de casos de uso del módulo matriz

3.6.4 Diagrama de casos de uso del módulo resultado de equipos

En la Figura 3-5 se muestra el diagrama de casos de uso el envió del status, registro de tiempo, cálculo de tiempo (carga/descarga), adquisición de valores(disponibilidad, rendimiento y calidad), despliegue de valores, reporte de metas y formulario de no cumplimiento.

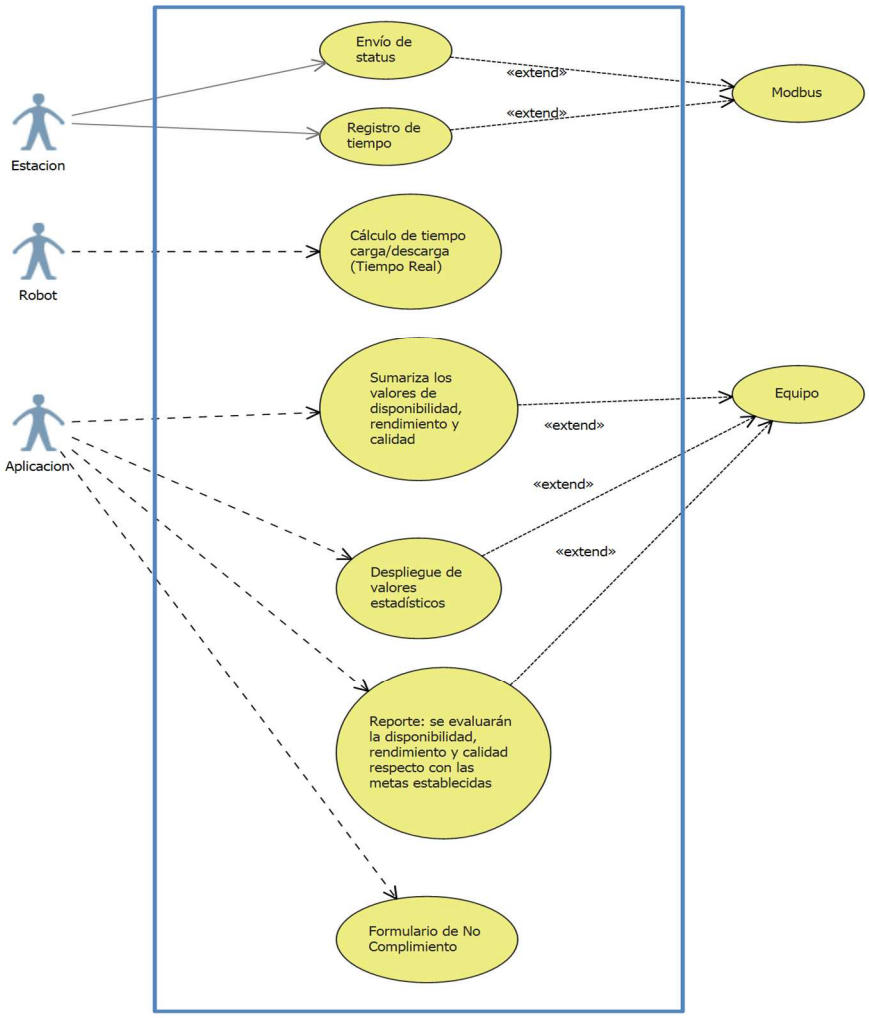


Figura 3-5 Diagrama de casos de uso del módulo resultado de equipos

3.7 Modelo conceptual

En esta sección del capítulo 3 se describen el modelo conceptual que constituyen el prototipo. Para la creación del modelo conceptual de este prototipo se tomaron en cuenta los requerimientos funcionales, casos de uso de alto nivel y los diagramas de caso de uso.

3.7.1 Conocimiento de la nomenclatura del dominio

En este capítulo para la obtención del modelo conceptual se obtuvieron los siguientes conceptos a partir del dominio (Figura 3-6):

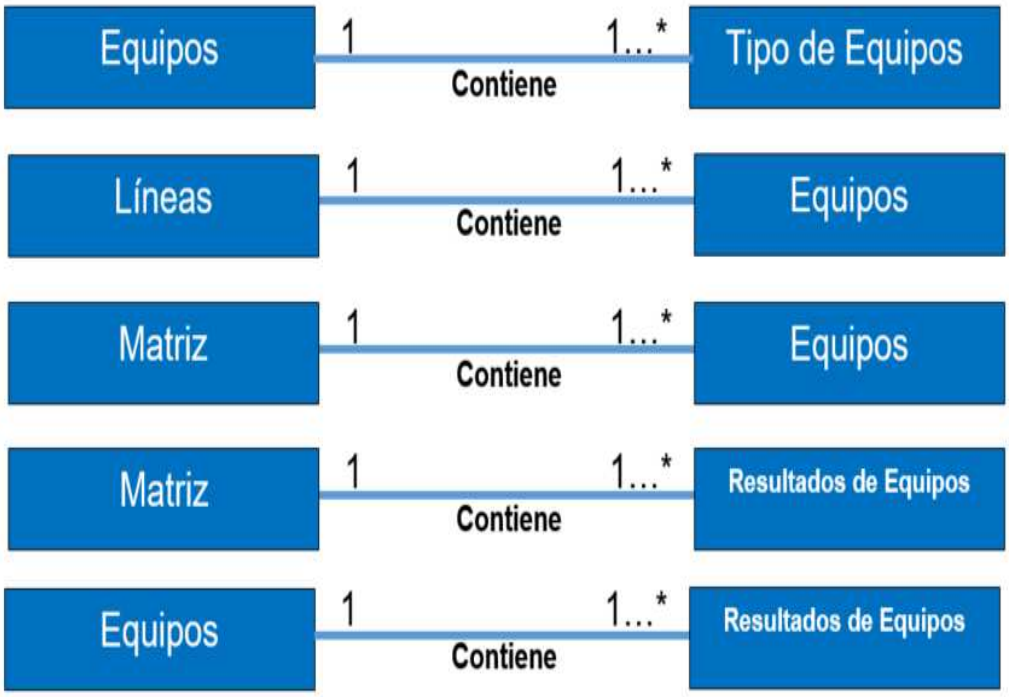


Figura 3-6 Casos de uso del módulo de resultado de equipos

A partir del modelo conceptual individual se obtiene un modelo conceptual resultante (Figura 3-7), en donde se visualiza todas las interacciones entre cada uno de ello y el actor. Horas Reales Producidas

Como puede observarse el modelo conceptual obtenido para este domino en particular es una representación visual de las clases conceptuales objetos del mundo real (Carretero & Garcia, Problemas resueltos de programación en lenguaje C++, 2004).

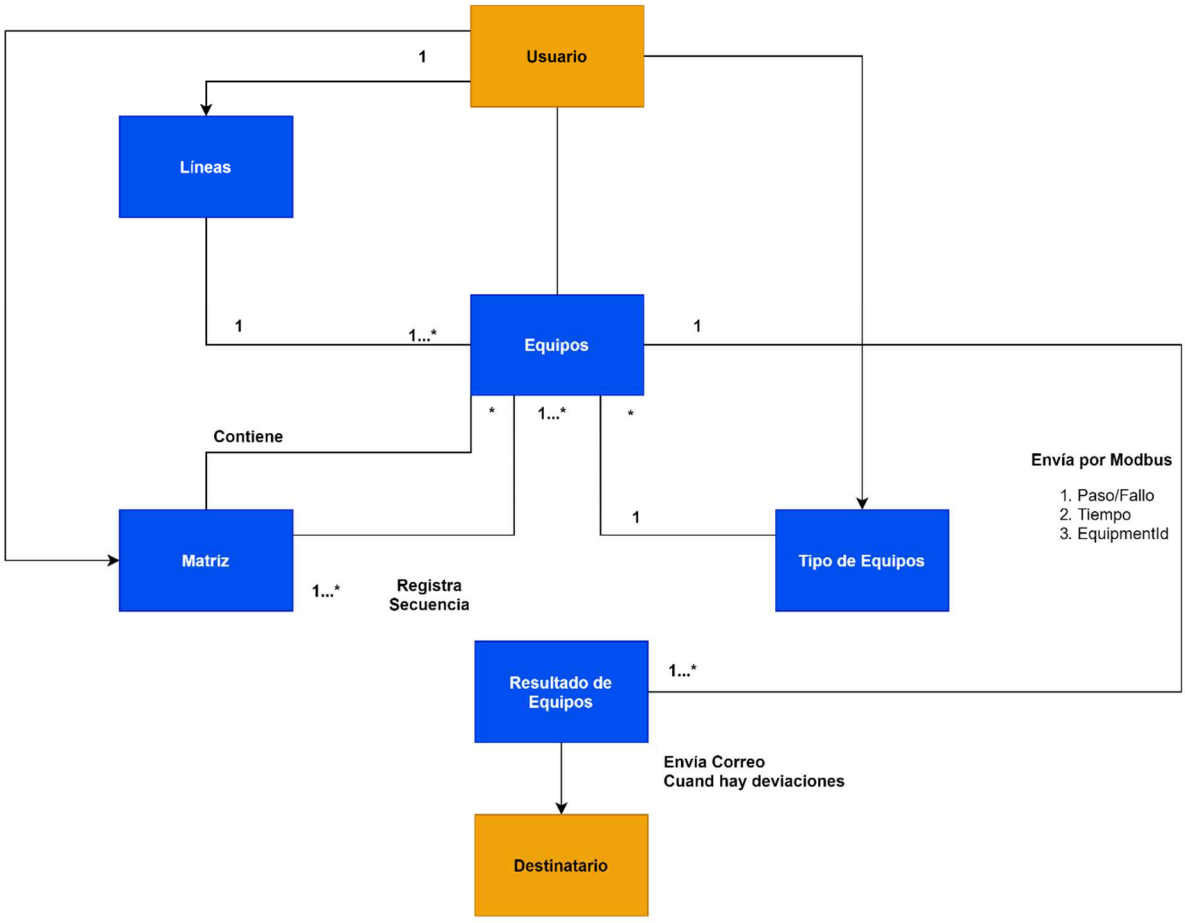


Figura 3-7 Modelo Conceptual del sistema

3.7.2 Casos de uso expandidos

Los casos de uso expandidos del sistema permiten entender de mejor manera la relación que existe entre el sistema y su entorno. Estos casos se consideren más importantes ya que más son útiles para alcanzar un conocimiento más profundo de los procesos y los requerimientos asociados a un caso de uso.

Además detalla las acciones del usuario en el sistema como una secuencia de interacciones de los actores con el sistema se crea un caso de uso ampliado "para cada" funcionalidad del sistema que está presente en el diagrama de casos de uso.

3.7.3 Caso de uso del módulo de Resultado de Equipos

Tabla 3.25 Registro de tiempo y estatus

Caso de uso:	Registro de tiempo y estatus
Actores:	Estación, Aplicación
Propósito:	Primario
Descripción:	El equipo o estación envía el estatus y tiempo de la duración de la operación, la aplicación realizará la sumatoria total de los valores de disponibilidad, rendimiento y calidad por equipo, además de actualizar los datos en pantalla, así como calcular los valores como el promedio, el valor máximo, el valor mínimo, cantidad de cargas y descargas, desviación estándar.
Tipo:	Primario y esencial
Requerimiento:	REquipResult1.1, REquipResult1.2, REquipResult1.3, REquipResult1.4, REquipResult1.5, REquipResult1.6, REquipResult1.7

3.7.4 Curso normal de los eventos

Tabla 3.26 Curso normal de los eventos

Acción de los actores	Respuesta
1.- Equipo envía el estatus y el tiempo de la operación de la estación	2.- La aplicación verifica el estatus y dependiendo del estatus se determina el siguiente paso donde si paso coloca el producto en la siguiente estación.
	3.- independientemente, el estatus del proceso ó estación (Pasa o Falla). La aplicación realiza sumatorias de los valores de disponibilidad, rendimiento, calidad y OEE por equipo y son comparados con los valores predeterminados en el catálogo de equipos
	4.- La aplicación muestra los valores como el promedio, valor máximo, valor mínimo, cantidad de cargas y desviación estándar por equipo.
5. – El sistema genera un reporte (cada hora) donde evalúa los valores obtenidos, con respecto a los predefinidos en cada equipo.	

3.7.5 Curso alterno

- Línea 2: En caso de obtener una falla procedente del equipo la unidad será colocada por el robot en el contenedor de fallas.
- Línea 5: En caso de no cumplir, se envía un correo electrónico al jefe de grupo y/o supervisor, para indicar cuál fue la razón de no cumplir las metas.

3.8 Descripción de casos reales de uso

3.8.1 Introducción

Los casos reales de uso presentan un diseño concreto del Caso de Uso. Es una de las primeras actividades en el ciclo de desarrollo.

Su creación depende de los Casos esenciales de uso generados en el análisis. Un caso real de uso describe el diseño concreto del caso de uso a partir de una tecnología particular de entrada y salida, y su implementación global. Por ejemplo, si hay una interfaz gráfica para el usuario, el caso de uso real incluirá diagramas de las ventanas en cuestión y una explicación de la interacción. En la Figura 3-8 se representa el caso real de uso del módulo de resultado de equipos.

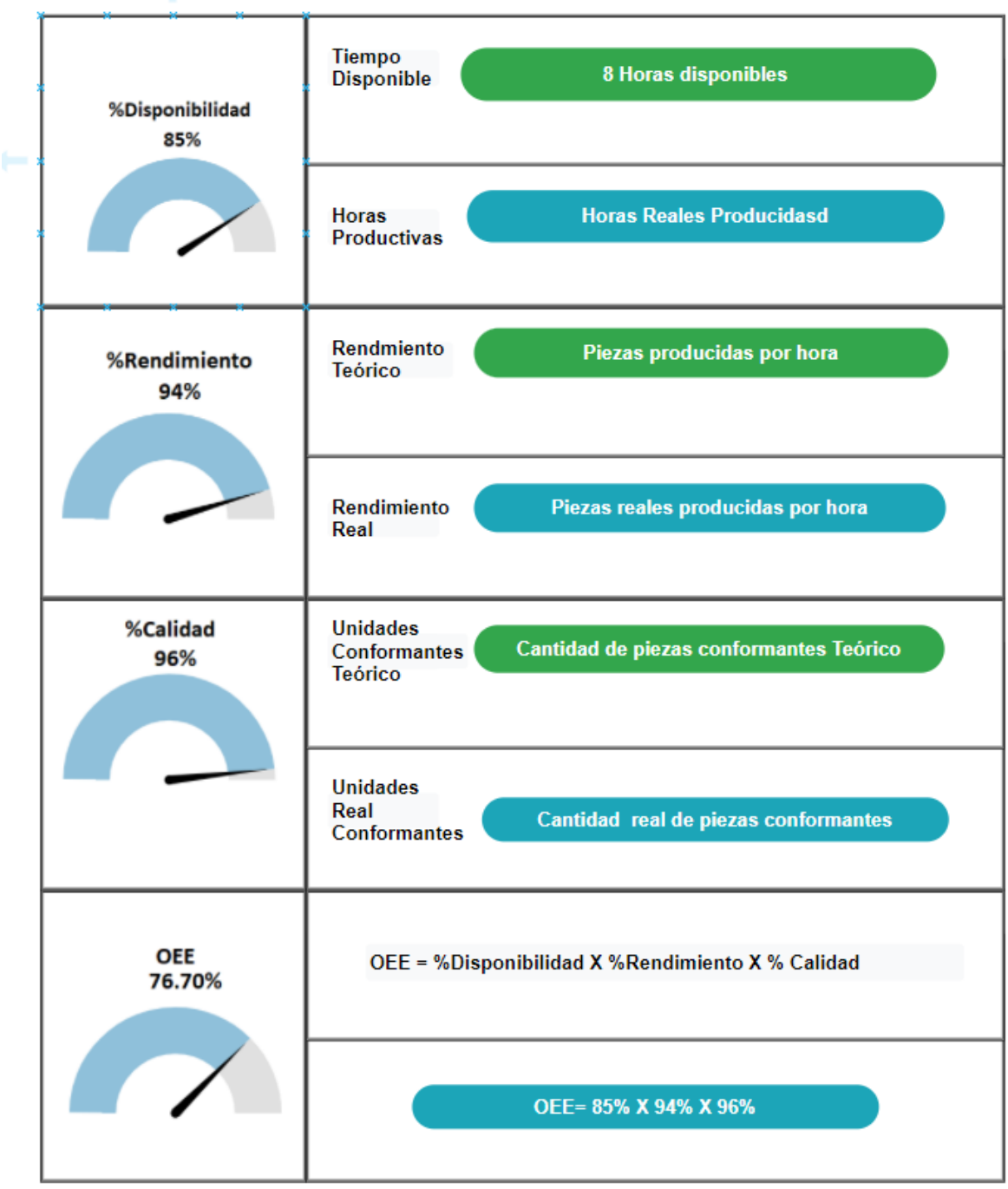


Figura 3-8 Casos de uso del módulo de Resultado de equipos

CAPÍTULO 4. DESARROLLO

El prototipo consta de las siguientes interfaces gráficas de usuario (Línea, Tipos de Equipos, Matriz, Resultado de equipos), las cuales permiten llevar a cabo la configuración para poder determinar el cálculo de OEE.

4.1 Herramientas necesarias para el desarrollo del prototipo

Para el desarrollo del prototipo las herramientas utilizadas son: Un Robot colaborativo UR Robot (URSim), Node Red, este fue seleccionado por su facilidad en el manejo del protocolo Modbus además de su versatilidad al poder ser utilizado para hacer consumo de API, del mismo modo se está utilizando Net Core para C#, Angular y MySQL.

4.1.1 Robot colaborativo UR Robot (URSim)

Un robot colaborativo, o COBOT, es un robot industrial que se caracteriza por ser de menores dimensiones que sus predecesores, han sido creados para automatizar procesos de producción repetitivos compartiendo el entorno de trabajo codo con codo con operarios humanos, ver Figura 4-1.

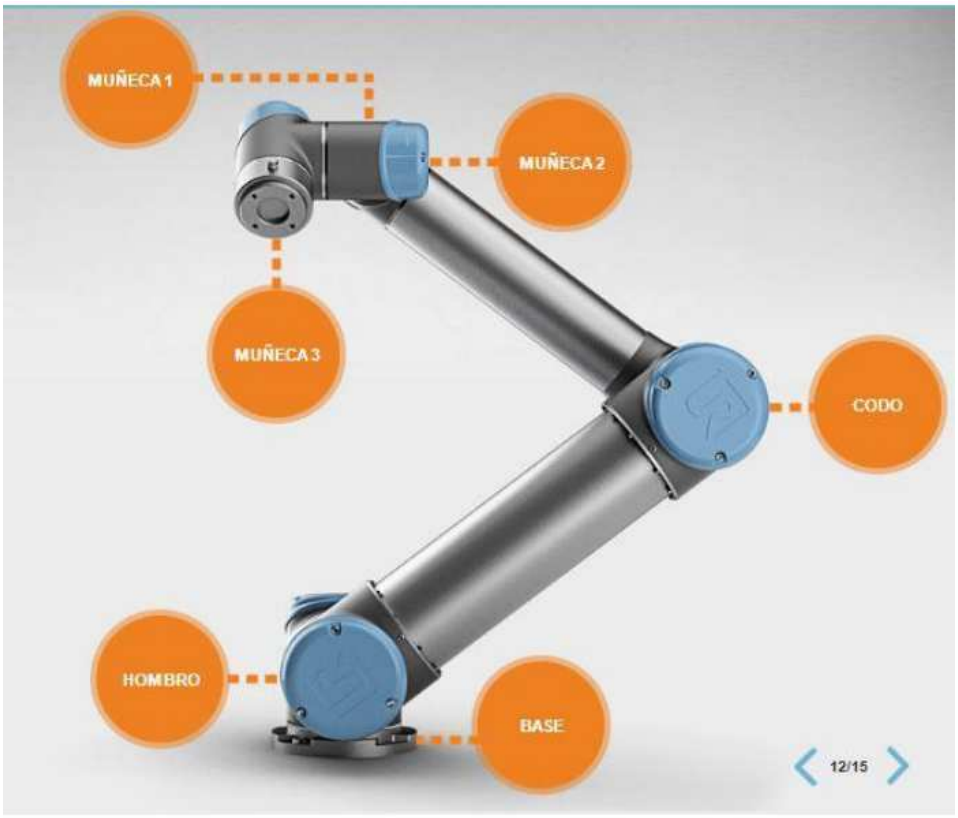


Figura 4-1 Ur Cobot

Para programar y controlar el brazo robot se utiliza un controlador o caja de control y la consola de programación.

En la caja de control se encuentra el tablero de control con las entradas y salidas al controlador y la conectividad con los equipos periféricos como pulsadores, interfaz de la máquina, dispositivos de seguridad y sensores.

También se cuenta con la placa madre, un microordenador de rápido poder de computación, Ethernet y conexión USB, se encuentra también un junto con el todo el software como el sistema operativo Lynux, la interfaz de programación PolyScope, patentada por Universal Robots, y los programas creados por el usuario.

Mediante la caja de control se pueden conectar 16 entradas digitales,16 salidas digitales, 2 entradas analógicas y 2 salidas analógicas y soporta comunicaciones TCP/IP de 100 Mbit, Modbus TCP, Profinet y Ethernet IP, ver Figura 4-2.

Figura 4-2 Módulo de comunicación, entradas y salidas digitales y análogas

4.1.1.1 Entorno de trabajo

En este apartado se realiza una breve introducción a las plataformas utilizadas que trabajarán en conjunto ya sea para la configuración, funcionamiento, visualización, envío de datos, generación de aplicaciones o acceder a los dispositivos utilizados.

4.1.1.2 URSim

URSim es un software de simulación que se utiliza para la programación y simulación de aplicaciones off-line con los robots de Universal Robots, ver Figura 4-3

Al no trabajar con los robots físicos aparecen una serie de limitaciones como el control de fuerza real, aunque es posible incluso simular entradas digitales externas.

Figura 4-3 Inicio Virtual Machine

URSim está hecho para funcionar sobre el sistema operativo Linux por lo que para ejecutarlo sobre Windows necesitaremos trabajar con máquinas virtuales y así poder disponer de ambos sistemas operativos, aunque para ello necesitaremos un reproductor virtual, en nuestro caso hemos optado por VMWare Player.

En este proyecto se ha utilizado la máquina virtual de URSim versión 3.12.1 y una vez que arranca en el virtual player se visualiza la pantalla principal de PolyScope, interfaz de programación patentada por Universal Robots, ver Figura 4-4.

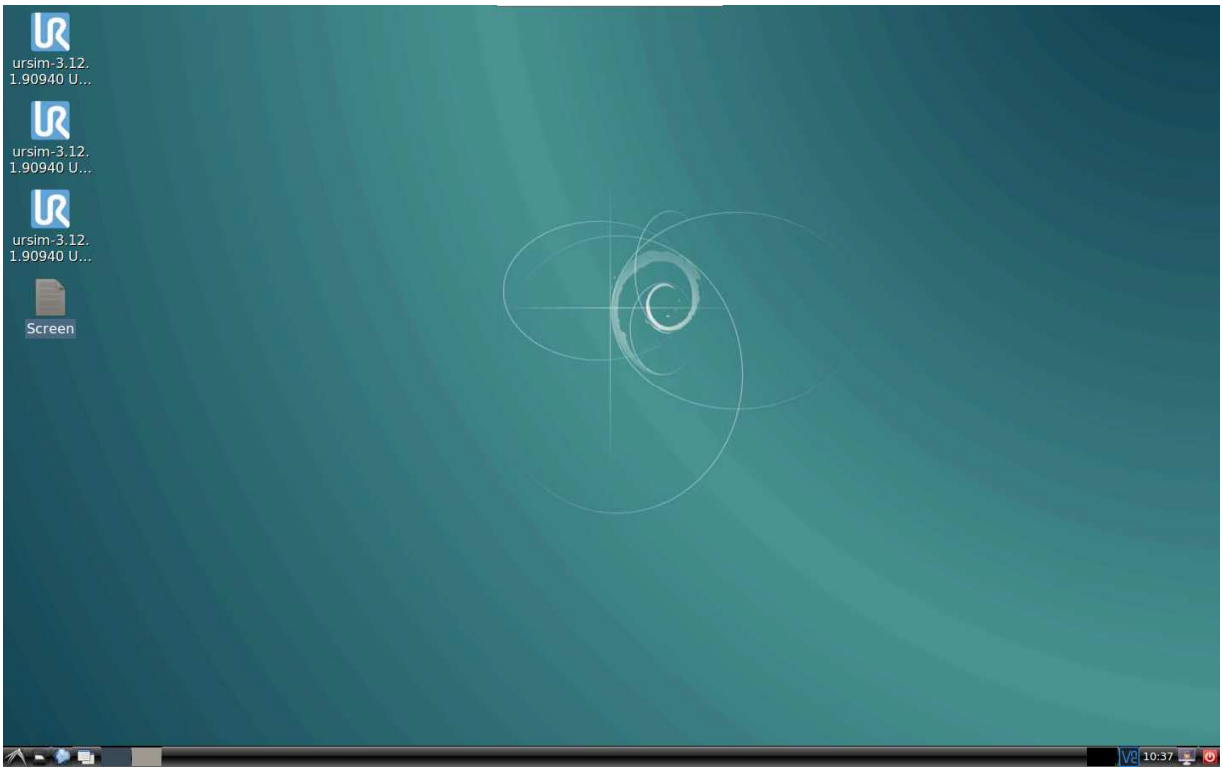


Figura 4-4 PolyScope UI

Se selecciona el acceso directo del simulador UR3, se configuran los registros de Modbus y se diseña el programa

4.1.2 Node-Red

Node-Red es un instrumento de programación visual basada en código abierto que permite integrar dispositivos de hardware y APIs. Su programación se basa en nodos y líneas de flujo que interconectan los nodos.

En la Figura 4-5 se muestra que las aplicaciones realizadas son equivalentes a flujos, que se encuentran compuestos por nodos; por lo tanto, un servidor puede ejecutar múltiples flujos estos son compuestos por un gráfico de nodos; además de los mensajes explícitos enviados, la información también se la puede enviar a través del flujo global y textos compartidos

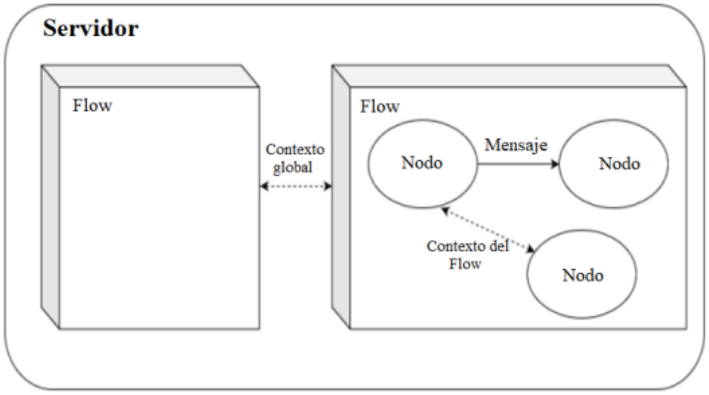


Figura 4-5 Diagrama de funcionamiento del Node-Red

Provee de una interfaz basada en un navegador web y permite al usuario crear flujos de eventos, realizando dicha conexión sin tener unos conocimientos avanzados de programación y utilizando una serie de bloques con unas funciones concretas para poder enlazarlos unos con otros.

Para instalar Node-Red es necesario que los diferentes sistemas operativos que permitan ejecutar la librería Node.js14. Por ejemplo, en dispositivos Windows, Mac o Linux.

Node-Red es una plataforma de código abierto por lo que es gratuita. Las herramientas están basadas en JavaScript y diseñadas sobre la plataforma Node.js. Node-Red maneja un sistema que facilita a los nodos ser representados por iconos apropiados. Se opera de dos maneras diferentes: arrastrar, soltar y conectar nodos, o importar código JavaScript. Los desarrolladores pueden conectar entradas y salidas a sus respectivos nodos de procesamiento y facilitar el flujo para el procesamiento de datos y al mismo tiempo controlar letras, mensajes etc.

Node-Red es una herramienta adaptable y al mismo tiempo poderosa, utilizada para crear prototipos. Este sistema permite la creación rápida de aplicaciones, especialmente aplicaciones que se activan en un evento como las aplicaciones de IoT.

La esencia de esta herramienta es permitir crear y configurar fácilmente en tiempo real aplicaciones en dispositivos finales (Figura 4-6)

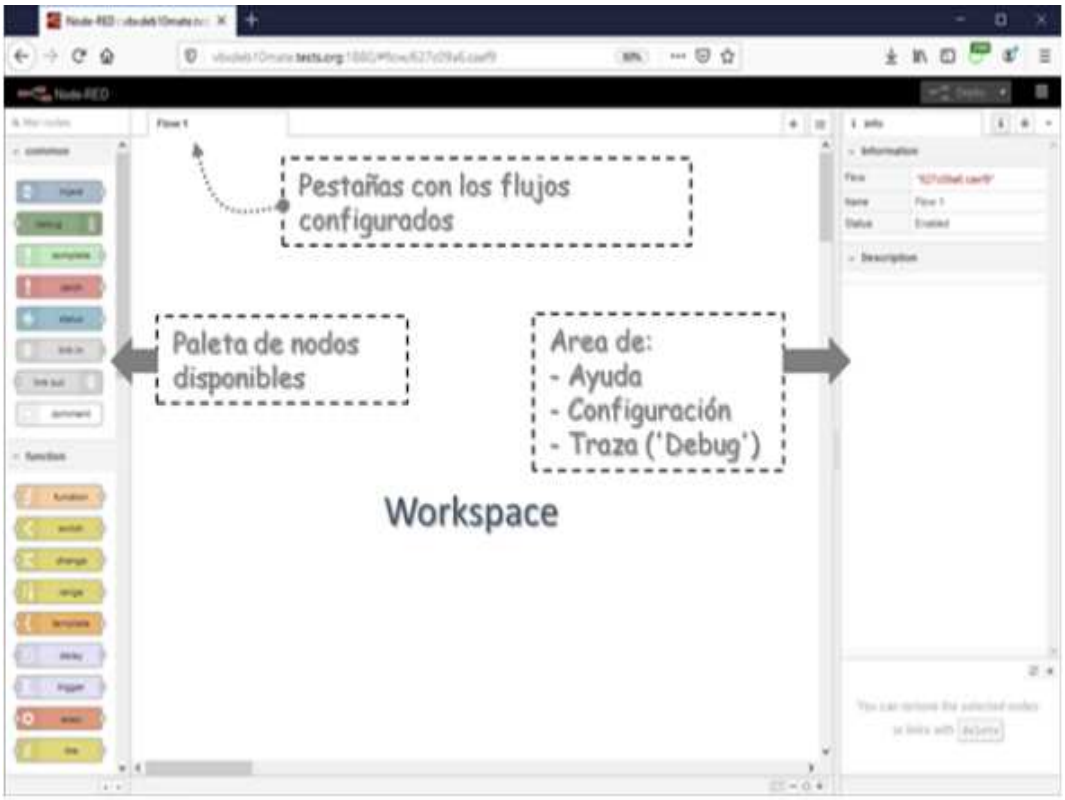


Figura 4-6 Interface de usuario de Node red

4.1.3 Net Core

El lenguaje de programación utilizado es C# parte de la plataforma de desarrollo .Net, se usará el IDE de Visual Studio 2019 con el Framework ASP.NET Core 2.2.

Net Core debido a que es una herramienta muy potente capaz de implementar protocolos Web API permite procesar y transforma información para que sea almacenada en la base de datos, además de contar con un módulo que tiene la versatilidad de su uso en tiempo real.

En la figura 4-7 se observa la estructura del modelo del lado del servidor, donde serán consumidos los servicios Web.

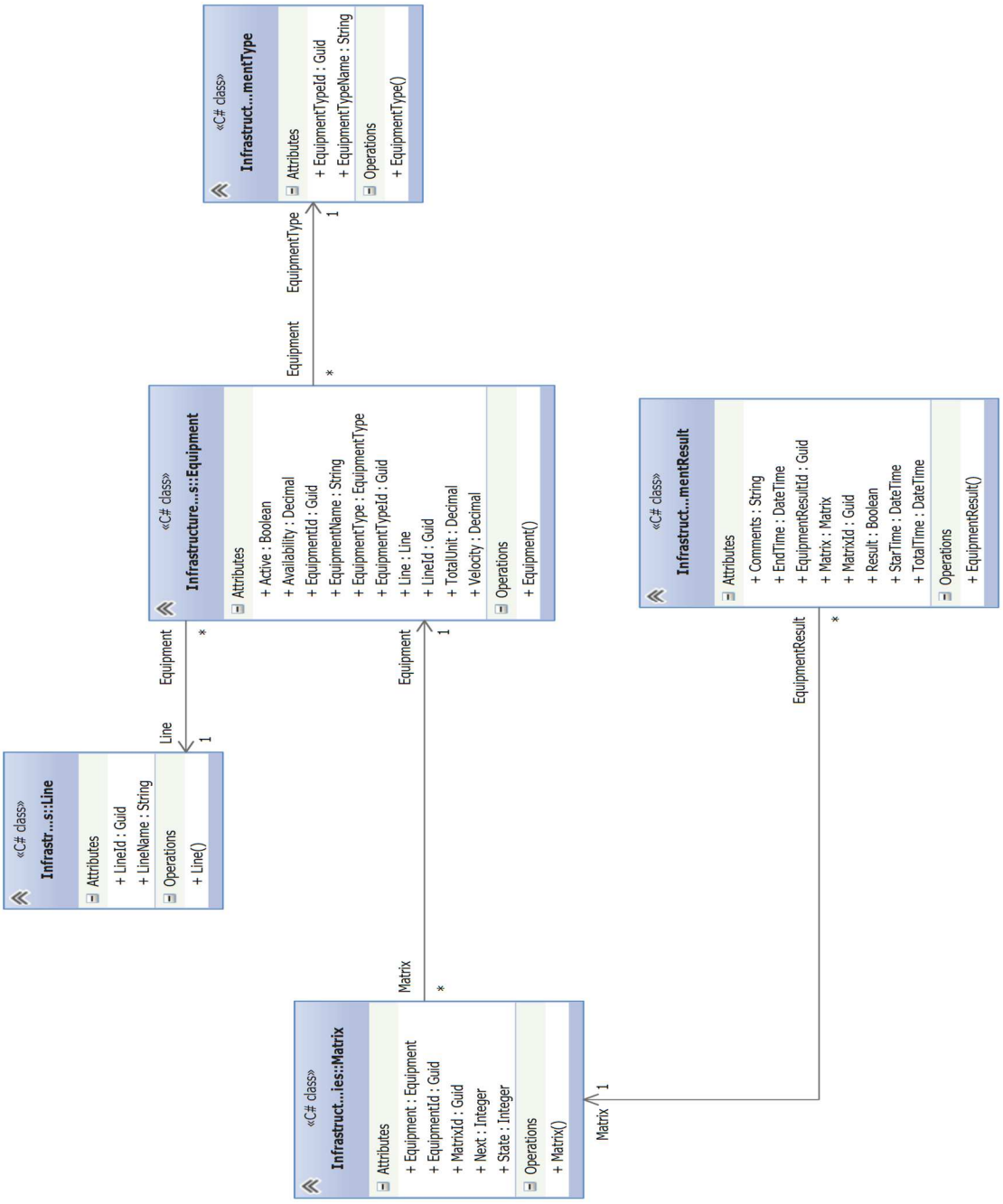


Figura 4-7 Diagrama de Clases

4.1.4 Angular

Angular es la herramienta usada para la construcción de interface gráfica para el usuario, en ella el administrador y el usuario ordinario serán capaces de informarse además de configurar los equipos y líneas para cada estación de trabajo conforme lo determinado.

Angular será capaz de enlazarse por medio de servicio web a la aplicación y de esta manera se llevará a cabo el intercambio de información en la interfaz de usuario y el servidor.

4.1.5 Base de datos

El motor de base de datos que se utilizo fue MySQL debido a su ligereza y de fácil implementación con la tecnología Net Core.

Se definieron los campos con los que va contar cada tabla y la descripción de cada campo.

4.1.5.1 Line

En la tabla 4.1 se definen los atributos para el objeto Line.

Tabla 4.1 Atributos para objeto Line

Nombre del atributo	Descripción	Ejemplo
LineId	Identificador único de la línea de producción. El identificador puede ser usado en otras partes del modelo cuando se	08d65973-e268-26e9-c4b4-525bd9efe682

	necesita identificar la línea.	
LineName	Contiene el nombre de la línea de producción.	Ford-GEN 3

4.1.5.2 EquipmentType

En la tabla 4.2 se definen los atributos para el objeto EquipmentType.

Tabla 4.2 Atributos para objeto EquipmentType

Nombre del atributo	Descripción	Ejemplo
EquipmentTypeId	Identificador único del tipo de estación de producción. El identificador puede ser usado en otras partes del modelo cuando se necesita identificar el tipo de estación.	08d65973-656a-0635-ecec-b23d1c808bb1
EquipmentTypeName	Contiene el nombre del tipo de equipo de la línea de producción.	ICT, FLASH

4.1.5.3 Equipment

En la tabla 4.3 se definen los atributos para el objeto Equipment.

Tabla 4.3 Atributos para el objeto Equipment

Nombre del atributo	Descripción	Ejemplo
EquipmentId	Identificador único del equipo de producción. El identificador puede ser usado en otras partes del modelo cuando se necesita identificar el equipo.	08d6816c-7024-149c-0078-40040964fb4f
EquipmentName	Contiene el nombre de la del equipo en la línea de producción.	Equipo de ICT del Modelo XXX
LineId	Contiene el Identificador de la línea de producción asignada.	08d65973-e268-26e9-c4b4-525bd9efe682
EquipmentType	Contiene el Identificador del tipo de estación.	08d65973-656a-0635-ecec-b23d1c808bb1
Active	Indica si el equipo se encuentra activo o inactivo.	True
Availability	Contiene la disponibilidad en minutos del equipo en estado ideal.	8000 min.
Velocity	Contiene la velocidad en	30 piezas por 60 min.

	que el equipo procesa cada unidad.	
TotalUnit	Contiene la cantidad total de unidades que debe de procesar en total en un turno determinado.	6000

4.1.5.4 Matrix

En la tabla 4.4 se definen los atributos para el objeto Matrix.

Tabla 4.4 Atributos para el objeto Matrix

Nombre del atributo	Descripción	Ejemplo
MatrixId	Identificador único de la Matriz de producción. El identificador puede ser usado en otras partes del modelo cuando se necesita identificar elemento de la matriz de estado.	08d67fc7-da99-fce2-524c-d431bc9db514
EquipmentId	Contiene el Identificador del equipo de producción.	08d6816c-7024-149c-0078-40040964fb4f
State	Contiene el estado actual al flujo de operación.	1000
Next	Contiene el estado	2000

	siguiente, si y sólo si ha pasado de su estado actual.	

4.1.5.5 EquipmentResult

En la tabla 4.5 se definen los atributos para el objeto EquipmentResult.

Tabla 4.5 Atributos para el objeto EquipmentResult

Nombre del atributo	Descripción	Ejemplo
EquipmentResultId	Identificador único de la tabla de resultados por equipo. El identificador puede ser usado en otras partes del modelo cuando se necesita identificar elemento el resultado de una operación.	08d67fc7-da99-fce2-524c-d431bc9db514
MatrixId	Contiene el Identificador de la matriz.	08d67fc7-da99-fce2-524c-d431bc9db514
StartTime	Contiene la fecha y hora que el equipo empezó a probar la unidad de producción.	24/01/19 00:00:00
EndTime	Contiene la fecha y hora que el equipo termino de probar la unidad de	24/01/19 00:05:00

	producción.	
TotalTime	Contiene la diferencia entre el start-time y el end-time en modo tiempo.	5 min
Comments	Contiene los comentarios y las razones por las que no se cumplió con las metas establecidas del OEE por equipo.	No, se cumplió ya que el equipo estaba dañado y no se pudo cumplir con la meta de las unidades por hora.

En base a lo expuesto se puede concluir que el desarrollo del paradigma orientado a objetos aporta un gran cambio en el modo en que observan los datos y los procedimientos que actúan sobre ellos. Tradicionalmente, los datos y los procedimientos se habían almacenado de forma separada: los datos y sus relaciones en la base de datos y los procedimientos en los programas de aplicación.

La orientación a objetos, sin embargo, combina los procedimientos de una entidad con sus datos.

Esta combinación se considera como un paso adelante en la gestión de datos. Las entidades son unidades auto contenidas que se pueden reutilizar con relativa facilidad. En lugar de ligar el comportamiento de una entidad a un programa de aplicación, el comportamiento es parte de la entidad en sí, por lo en cualquier lugar en el que se utilice la entidad, se comportará de un modo predecible y definido.

El modelo orientado a objetos también soporta interacciones entre muchos objetos, siendo éste el primer modelo que lo permite. Aun así, se debe ser muy cuidadoso al diseñar estas relaciones para evitar pérdidas de información.

En la figura 4-8 se puede observar el diagrama entidad relación que se utilizó para este proyecto:

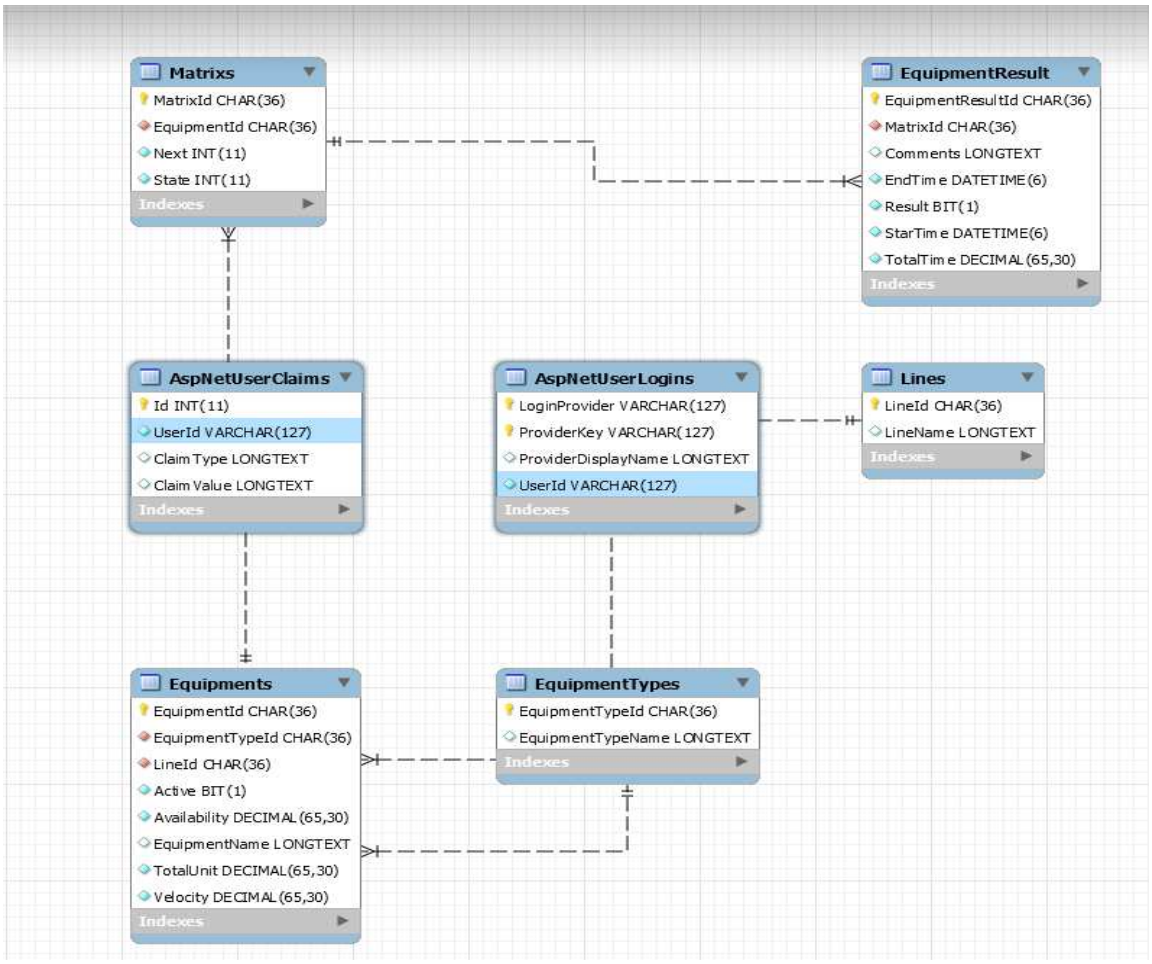


Figura 4-8 Diagrama entidad relación

4.1.6 Representación del modelo

En la figura 4-9 se muestra el diagrama de alto nivel del prototipo donde se observa interacción entre la interfaz de usuario, el servidor, base de datos y el robot lo cual es necesario para realizar el cálculo del OEE.

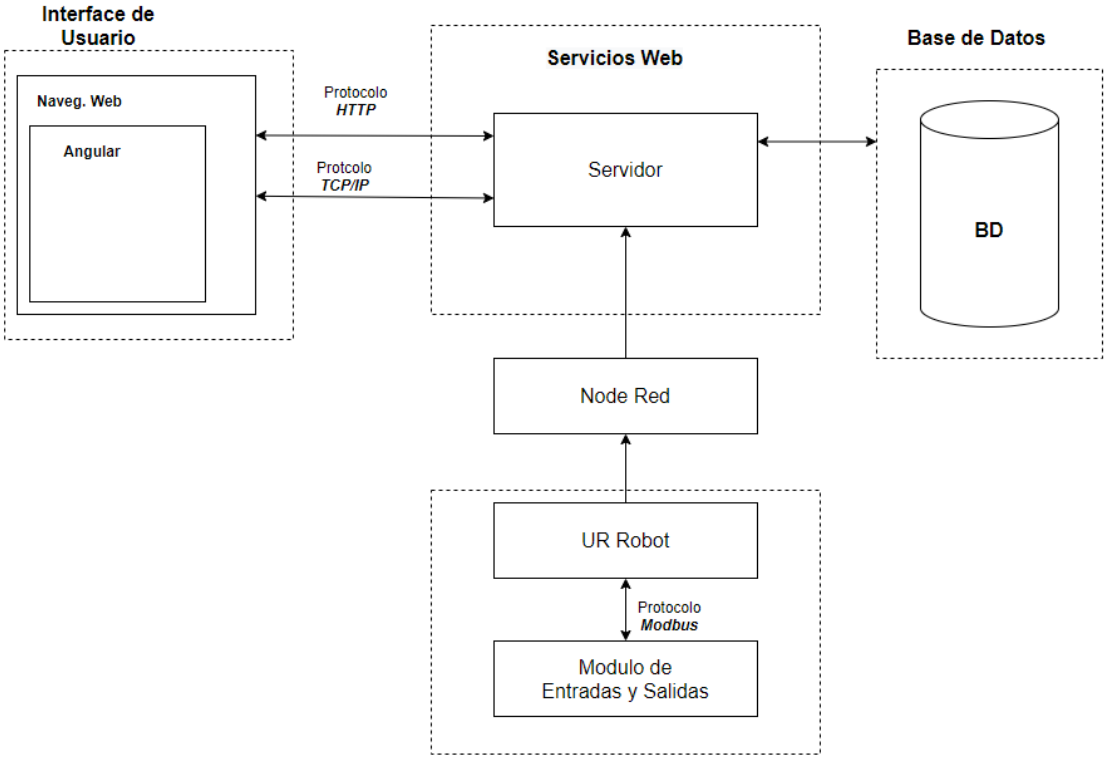


Figura 4-9 Diagrama de alto nivel

4.2 Prototipo

4.2.1 Implementación de la interface de usuario

Para la realización del prototipo se generaron páginas web de los casos de usos presentados en este documento. El prototipo cuenta con una interfaz gráfica sencilla fácil de llenar.

A continuación, se muestra de manera detallada la utilización de la aplicación para monitorear el OEE en líneas y equipos de manufactura.

En el menú de inicio, http://humbertopedraza.dynu.com:800/#/lines escoger la pestaña deseada:

- Líneas
- Equipo
- Tipo de equipo
- Matrix

4.2.1.1 Módulo de Línea

En este módulo de línea se muestra los diferentes pasos para añadir, editar y eliminar una línea de la aplicación.

Se Selecciona la opción Line (Fig. 4-10)

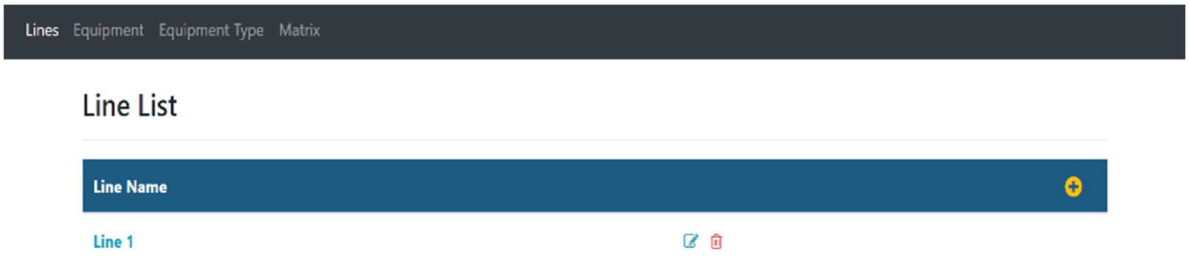


Figura 4-10 Lista de las líneas disponibles

4.2.1.1.1 Añadir una nueva Línea

En la siguiente figura 4-11 se muestra el proceso para añadir una nueva línea se hace clic sobre icono de cruz amarillo para ingresar los datos.

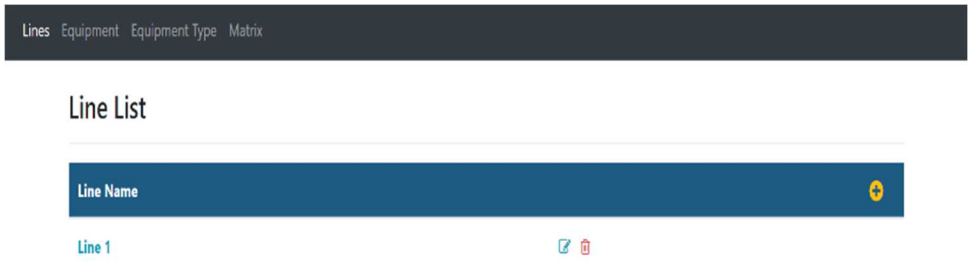


Figura 4-11 ingresar una nueva línea

Se debe introducir el nombre de la línea en el recuadro y a continuación se hace clic sobre botón Submit (Figura 4-12)

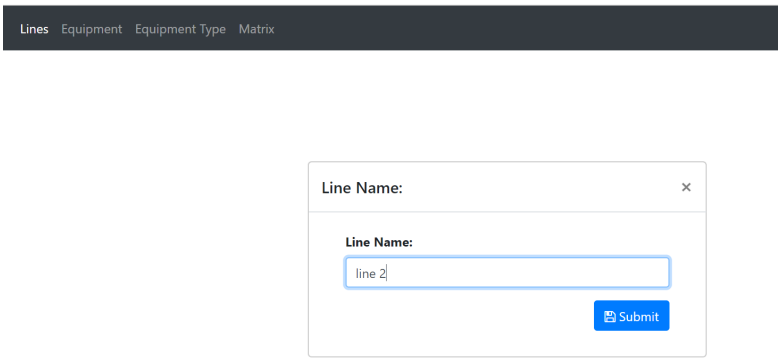


Figura 4-12 Introducir nombre de la línea

A continuación, se muestra en la figura 4-13 el listado de las líneas con su nuevo elemento agregado:

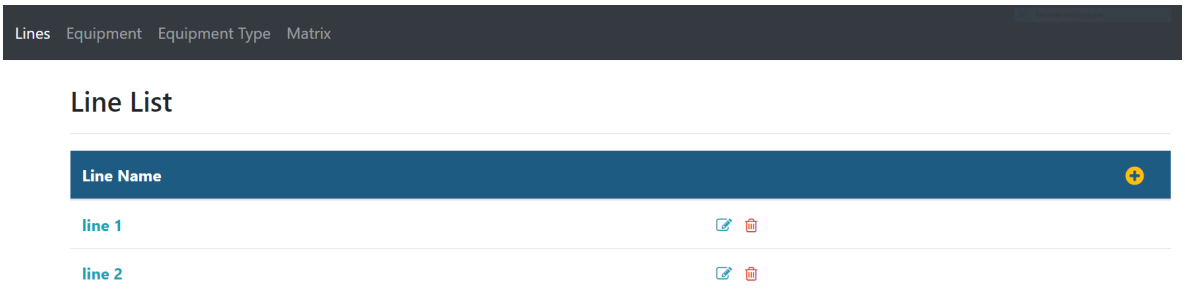


Figura 4-13 Línea agregada

4.2.1.1.2 Editar línea ya existente

En la siguiente figura 4-14 se muestra el proceso para editar una línea, se hace clic sobre icono azul con un lápiz y hoja.

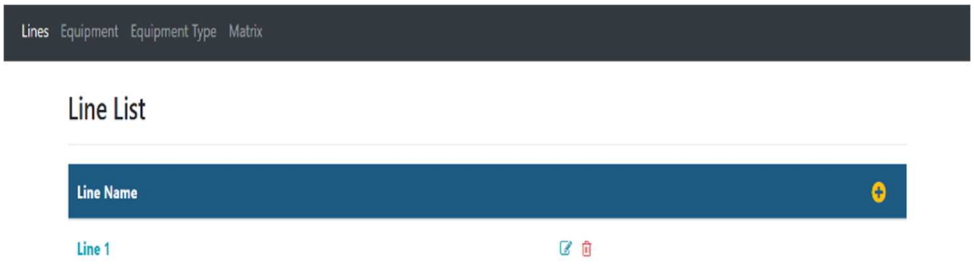


Figura 4-14 Editar una línea ya existente

Se debe introducir el nombre de la línea en el recuadro y a continuación se hace clic sobre botón Submit(Figura 4-15)

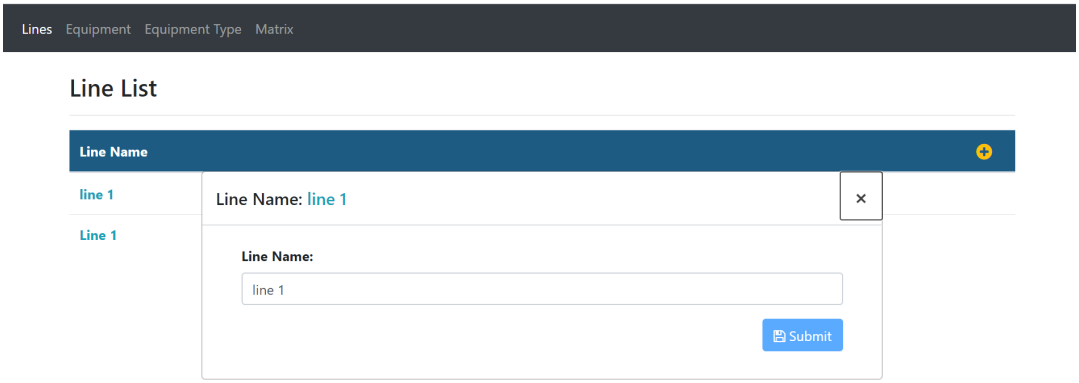


Figura 4-15 Pantalla emergente, para introducir datos

4.2.1.1.3 Eliminar línea

En la siguiente figura 4-16 se muestra el proceso para eliminar una línea.

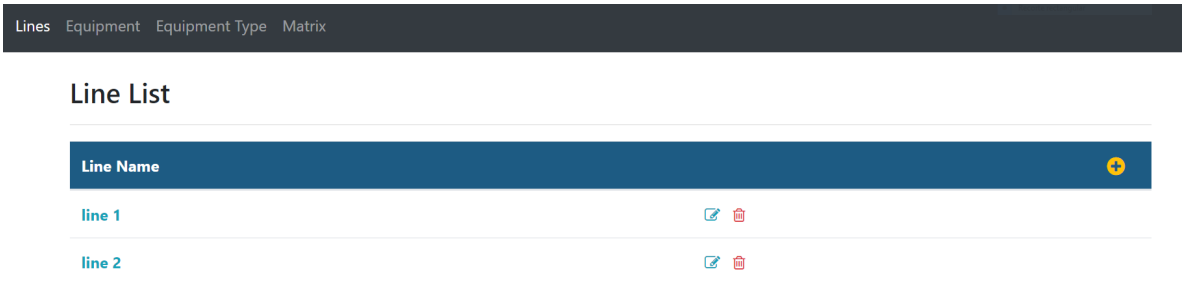


Figura 4-16 Eliminar línea

4.2.1.2 Módulo de tipo de equipos

En este módulo de tipos de equipos se muestra los diferentes pasos para añadir, editar y eliminar un tipo de equipo en la aplicación.

Se Selecciona la opción Equipment types ver en la figura (Fig. 4-17)

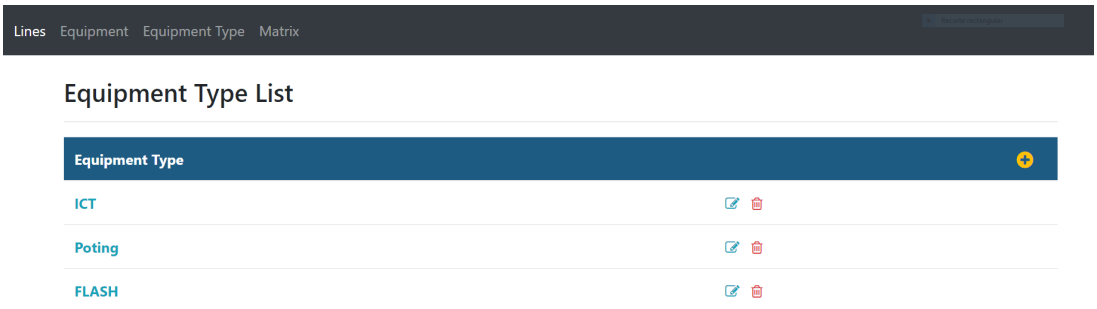


Figura 4-17 Lista de tipos de equipos disponibles

4.2.1.2.1 Añadir un tipo de equipo

En la siguiente figura 4-18 se muestra el proceso para añadir un tipo de equipo, se hace clic sobre el icono con una cruz amarilla, se introduce el nombre del tipo de equipo y dar clic en Submit

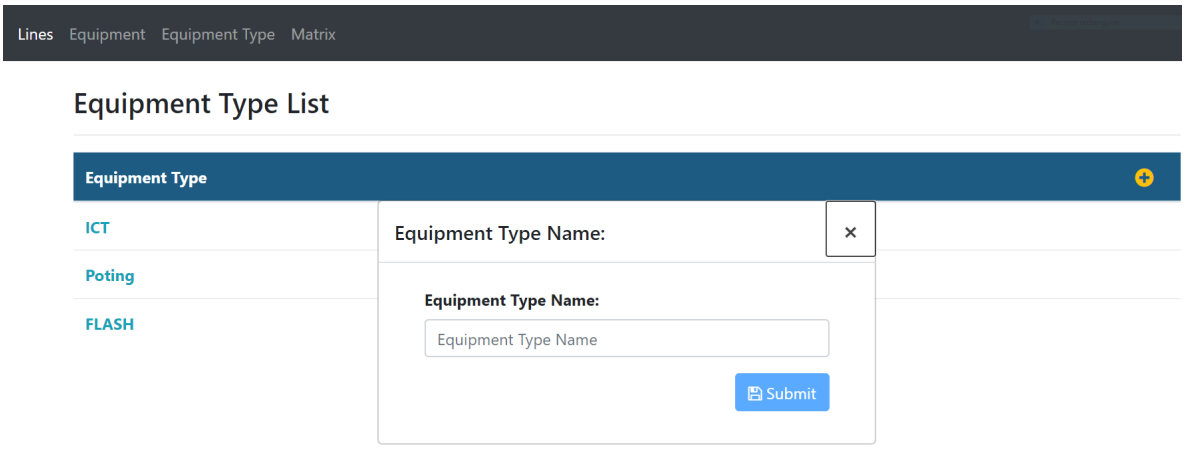


Figura 4-18 Ingresar un nuevo tipo de equipo

4.2.1.2.2 Editar tipo de equipo ya existente

En la siguiente figura 4-19 se muestra el proceso para editar un tipo de equipo, se hace clic sobre icono azul que contiene un lápiz y hoja, posteriormente se lleva a cabo la edición del registro.

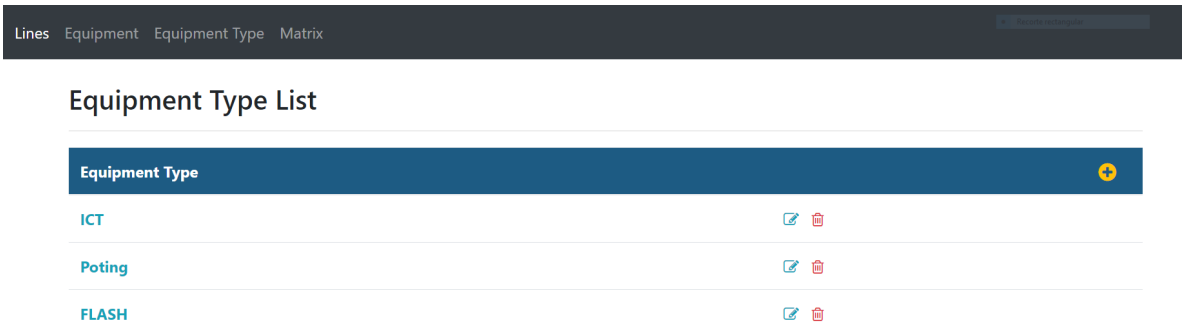


Figura 4-19 Editar un tipo de equipo

4.2.1.2.3 Eliminar tipo de equipo

En la siguiente figura 4-20 se muestra el proceso para eliminar un tipo de equipo, dando clic sobre el icono rojo de sesto de basura.

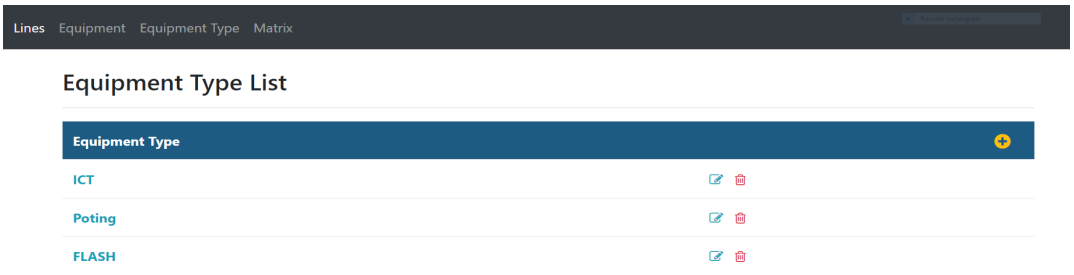


Figura 4-20 Eliminar tipo de equipo

4.2.1.3 Módulo equipos

En este módulo de equipos se muestra los diferentes pasos para añadir, editar y eliminar un equipo en la aplicación.

Se Selecciona la opción Equipment (Fig. 4-21)

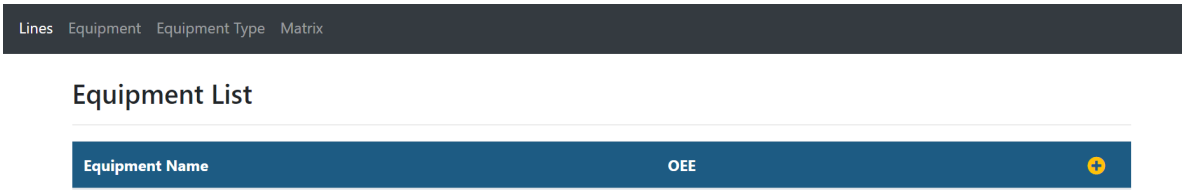


Figura 4-21 Lista de equipos disponibles

4.2.1.3.1 Añadir equipo

En la siguiente figura 4-22 se muestra el proceso para añadir un equipo, se hace clic sobre icono con una cruz, marcado con un círculo rojo, se debe de introducir los siguientes datos:

A continuación, introduzca los parámetros:

- Nombre del equipamiento: Introducir un nombre
- Línea: seleccionar la línea a la que pertenece
- Tipo de equipo: seleccionar el tipo de equipo
- Activo
- Disponibilidad
- Total de unidades
- Velocidad

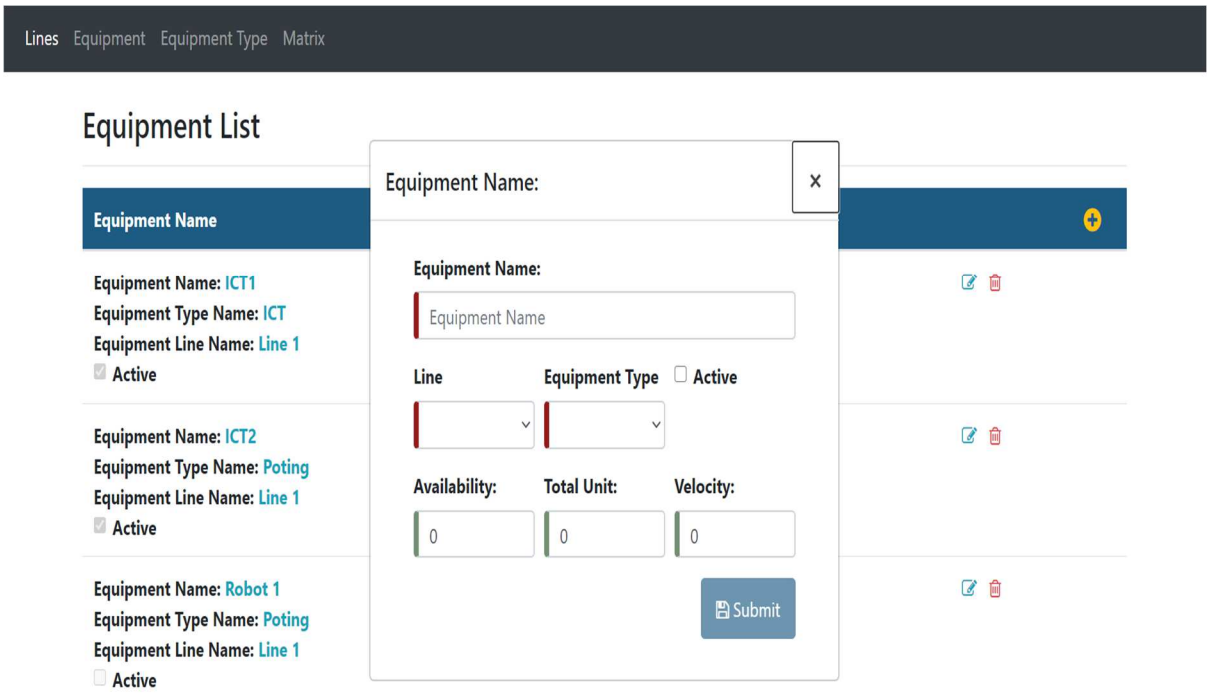


Figura 4-22 Ingresar un nuevo equipo

4.2.1.3.2 Editar equipo ya existente

En la siguiente figura 4-23 se muestra el proceso para editar una línea ya existente, se hace clic sobre icono marcado con un círculo rojo, posteriormente se lleva a cabo la edición del registro y dando clic submit

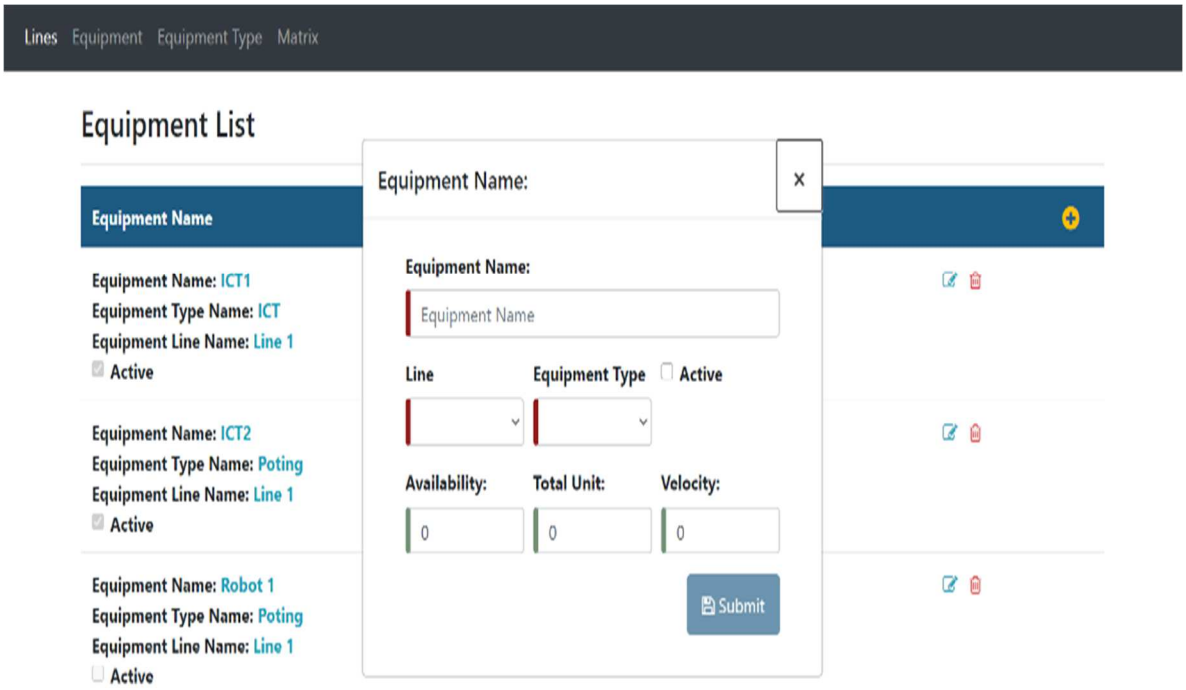


Figura 4-23 Editar un equipo ya existente

4.2.1.3.3 Eliminar equipo

En la siguiente figura 4-24 se muestra el proceso para eliminar un equipo, dando clic sobre el icono rojo de sesto de basura

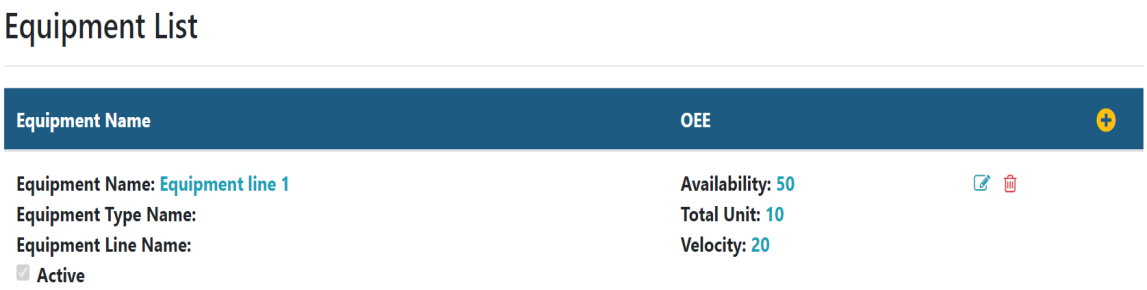


Figura 4-24 Eliminar un equipo existente

4.2.1.4 Módulo Matrix

En este módulo de matriz de estados se muestra los diferentes pasos para añadir, editar y eliminar un segmento de flujo en la aplicación.

Se Selecciona la opción Matrix (Fig. 4-25)

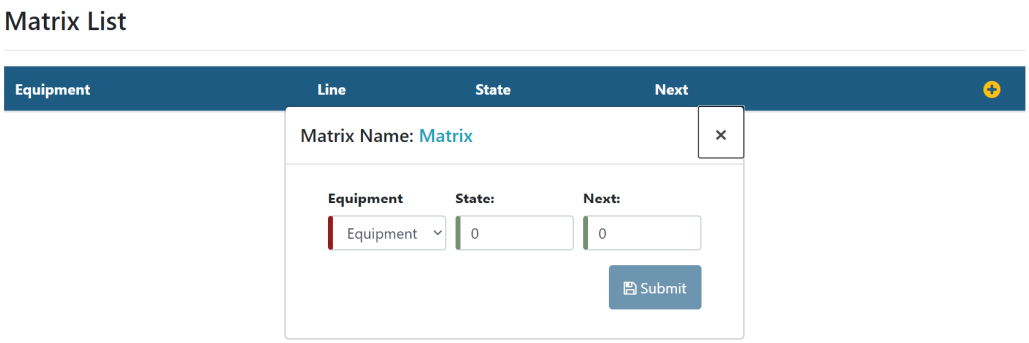


Figura 4-25 Lista de matriz de estados

4.2.1.4.1 Añadir equipo a la matriz de estado

En la siguiente figura 4-26 se muestra el proceso para añadir un elemento a la matriz de estado, se hace clic sobre icono con una cruz amarilla, se debe de introducir los siguientes datos:

A continuación, introduzca los parámetros:

- Equipment
- State
- Next

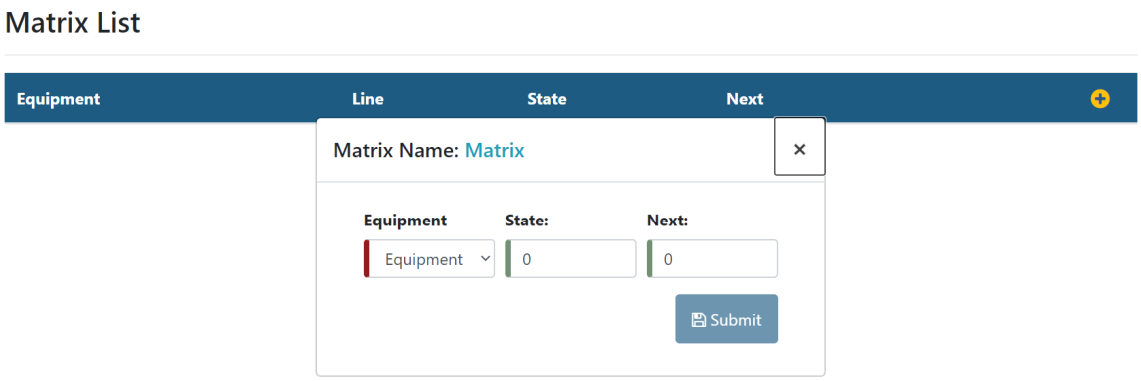


Figura 4-26 ingresar elemento a la matriz de estado

4.2.1.4.2 Editar equipo en la matriz de estado existente

En la siguiente figura 4-27 se muestra el proceso para editar un elemento de la matriz de estado, se hace clic sobre icono azul con un lápiz y hoja, posteriormente se lleva a cabo la edición del registro y dando clic submit

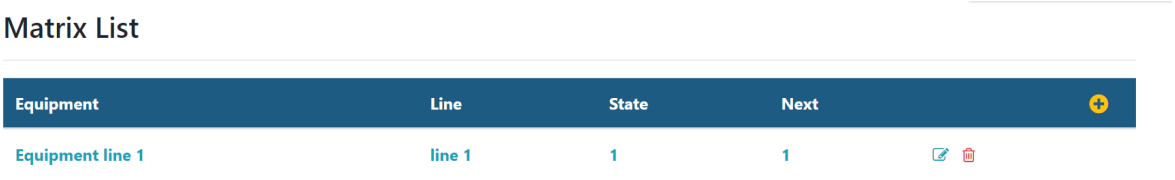


Figura 4-27 Editar elemento de la matriz ya existente

4.2.1.4.3 Eliminar elemento de la matriz de estado

En la siguiente figura 4-28 se muestra el proceso para eliminar un elemento de la matriz de estados, dando clic sobre el icono rojo de sesto de basura

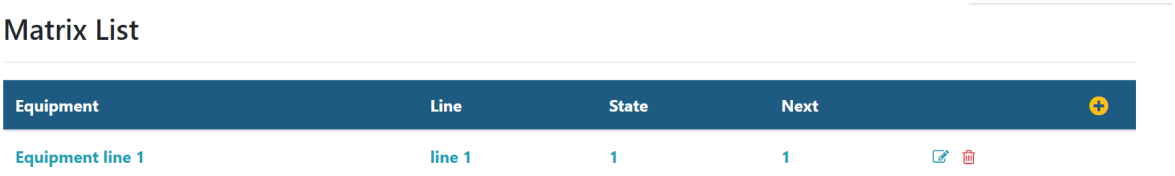


Figura 4-28 Eliminar tipo de equipo

4.2.2 Implementación de comunicación entre Robot y Modbus

Para realizar la comunicación entre el robot y los servicios web fue necesario utilizar Node Red, para poder tomar las lecturas de los dispositivos de entradas y salidas y evaluarlos a lo largo del proceso para ser enviados a la base de datos.

4.2.2.1 Enviar datos al Servidor

A continuación, se muestra cómo se elabora el diagrama para obtener los datos y ser evaluados por Node Red.

En la siguiente figura 4-29 se muestra donde se debe colocar el URL para que se envíen los datos procesados a través del controlador y el robot.

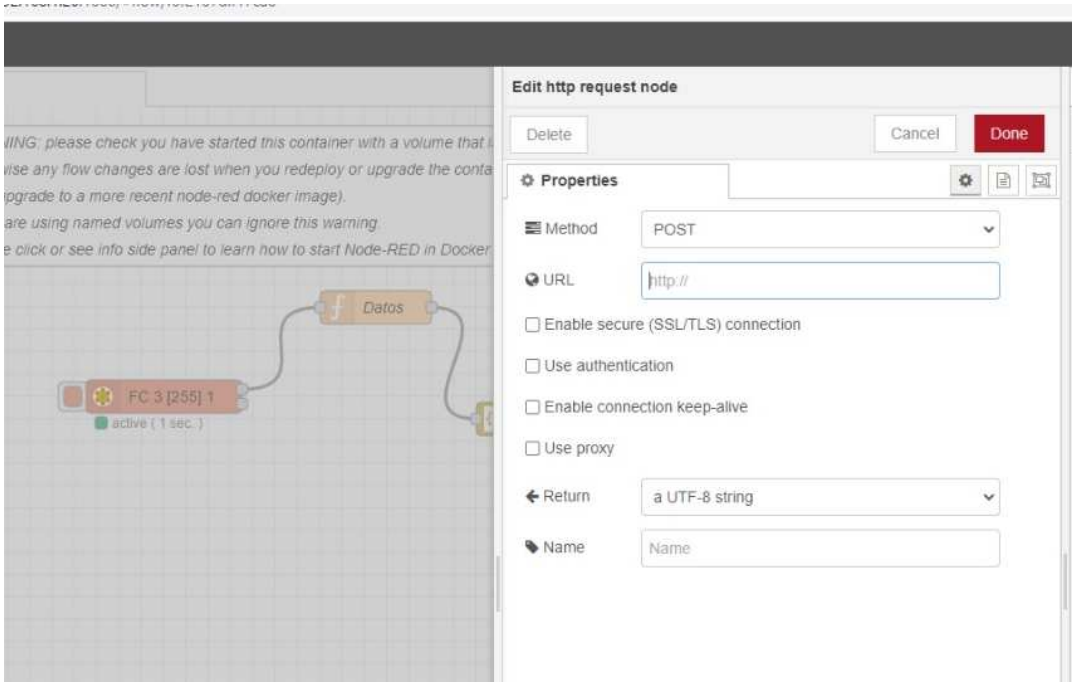


Figura 4-29 Post hacia el servidor

CAPÍTULO 5. RESULTADOS Y CONCLUSIONES

5.1 Introducción

En este capítulo se corre una simulación que permite obtener resultados en una celda de producción automatizada con el fin de determinar el OEE de un equipo de ICT el cual tiene como objetivo producir 300 unidades por turno de 7:00AM a 4:00PM y contar con una capacidad de 36 unidades por hora. El objetivo de correr esta simulación es evaluar su OEE por hora y turno. En esta simulación se utilizó el prototipo construido para este propósito.

5.2 Resultados

Al inicio de este proyecto se expuso la necesidad de realizar el desarrollo del cálculo de OEE de manera automática usando robots colaborativos, debido que en la actualidad es difícil de determinar usando los métodos manuales, para ello se desarrolló una aplicación con la característica que se pueda colocar en ella el tiempo de trabajo en horas (turno), capacidad total (cantidad a producir) y capacidad por hora.

En el capítulo de análisis se realizó y se determinó los casos de usos funcionales para determinar el comportamiento de la aplicación y a partir de ahí se obtuvo un modelo conceptual el cual fue desarrollado y aplicado en el capítulo de desarrollo.

Una vez obtenido el modelo conceptual se tomó como base para construir el diagrama de clases y con este fue posible codificar en C#.

Para demostrar la funcionalidad de esta aplicación se utilizó una estación de ICT en la cual se tienen las siguientes características:

- Disponibilidad 7:00AM a 4:00PM
- Capacitad total de 300 unidades
- Capacidad de 36 unidades por hora

Se realizó la recopilación de la información mediante la aplicación de OEE obteniendo como resultado los siguientes datos

5.2.1 Disponibilidad

La aplicación recopiló información de disponibilidad (Tabla 5.1), en la tabla se muestra el tiempo efectivo del equipo en minutos y el porcentaje de disponibilidad por hora en el turno el cual promedia 48.02 minutos de tiempo efectivo y una disponibilidad del 80%.

Tabla 5.1 Disponibilidad en minutos por turno

Hora	Disponibilidad real en minutos	Disponibilidad teórica en minutos	%Disponibilidad real	%Disponibilidad Teórica
07:00 AM - 08:00AM	48.78	60	81%	100%
08:00 AM - 09:00AM	47.65	60	79%	100%
09:00 AM - 10:00AM	48.9	60	82%	100%
10:00 AM - 11:00AM	46.82	60	78%	100%
11:00 AM - 12:00PM	48.08	60	80%	100%
12:00 AM - 13:00PM	48.32	60	81%	100%
13:00 PM - 14:00PM	48.97	60	82%	100%
14:00 PM - 15:00PM	46.43	60	77%	100%
15:00 PM - 16:00PM	48.23	60	80%	100%
Promedio	**48.02**		**80%**	

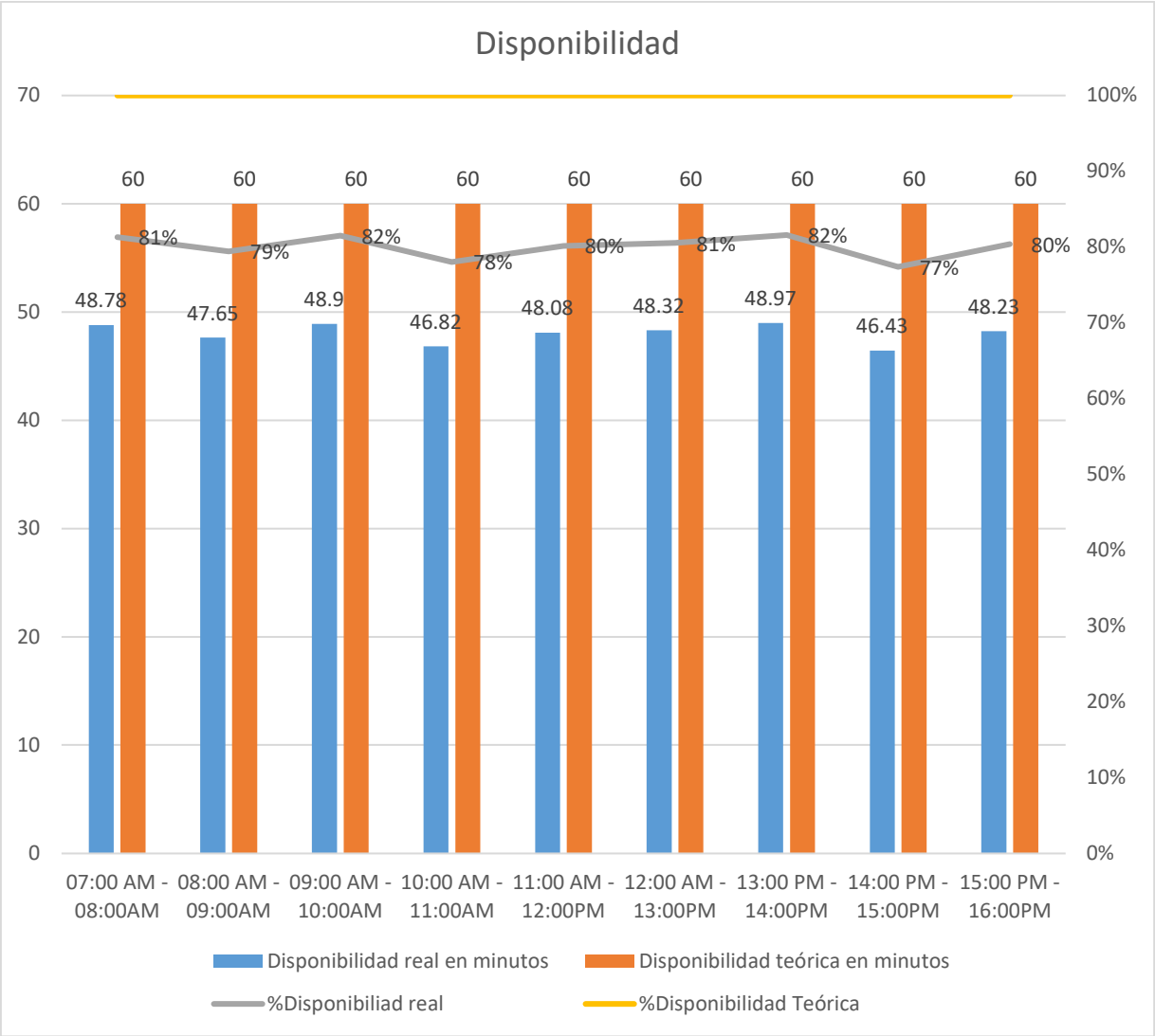


Figura 5-1 Gráfica de Disponibilidad

En la Figura 5-1 se observa que la barra color naranja cuenta con 60 minutos de disponibilidad teórica, en la barra color azul se aprecia el tiempo disponible real por hora, la línea color amarilla representa el 100% de la disponibilidad teórica y la línea color gris representa el porcentaje real de disponibilidad por hora, dando como promedio de 48.02 minutos efectivos por hora y una disponibilidad total de 80% por turno.

5.2.2 Rendimiento

La aplicación recopiló información de rendimiento (Tabla 5.2), en la tabla se muestra rendimiento de las unidades producidas por hora y el porcentaje de rendimiento del equipo. En promedio el rendimiento real es de 32.89 unidades por hora y el porcentaje de rendimiento es de 94%.

Tabla 5.2 Rendimiento por hora en un turno

Hora	Rendimiento real	Rendimiento teórico	%Rendimiento real	%Rendimiento Teórico
07:00 AM - 08:00AM	34	36	94%	100%
08:00 AM - 09:00AM	32	36	89%	100%
09:00 AM - 10:00AM	34	36	94%	100%
10:00 AM - 11:00AM	32	36	89%	100%
11:00 AM - 12:00PM	33	36	92%	100%
12:00 AM - 13:00PM	33	36	92%	100%
13:00 PM - 14:00PM	33	36	92%	100%
14:00 PM - 15:00PM	32	36	89%	100%
15:00 PM - 16:00PM	33	36	92%	100%
Promedio	32.89		91%	

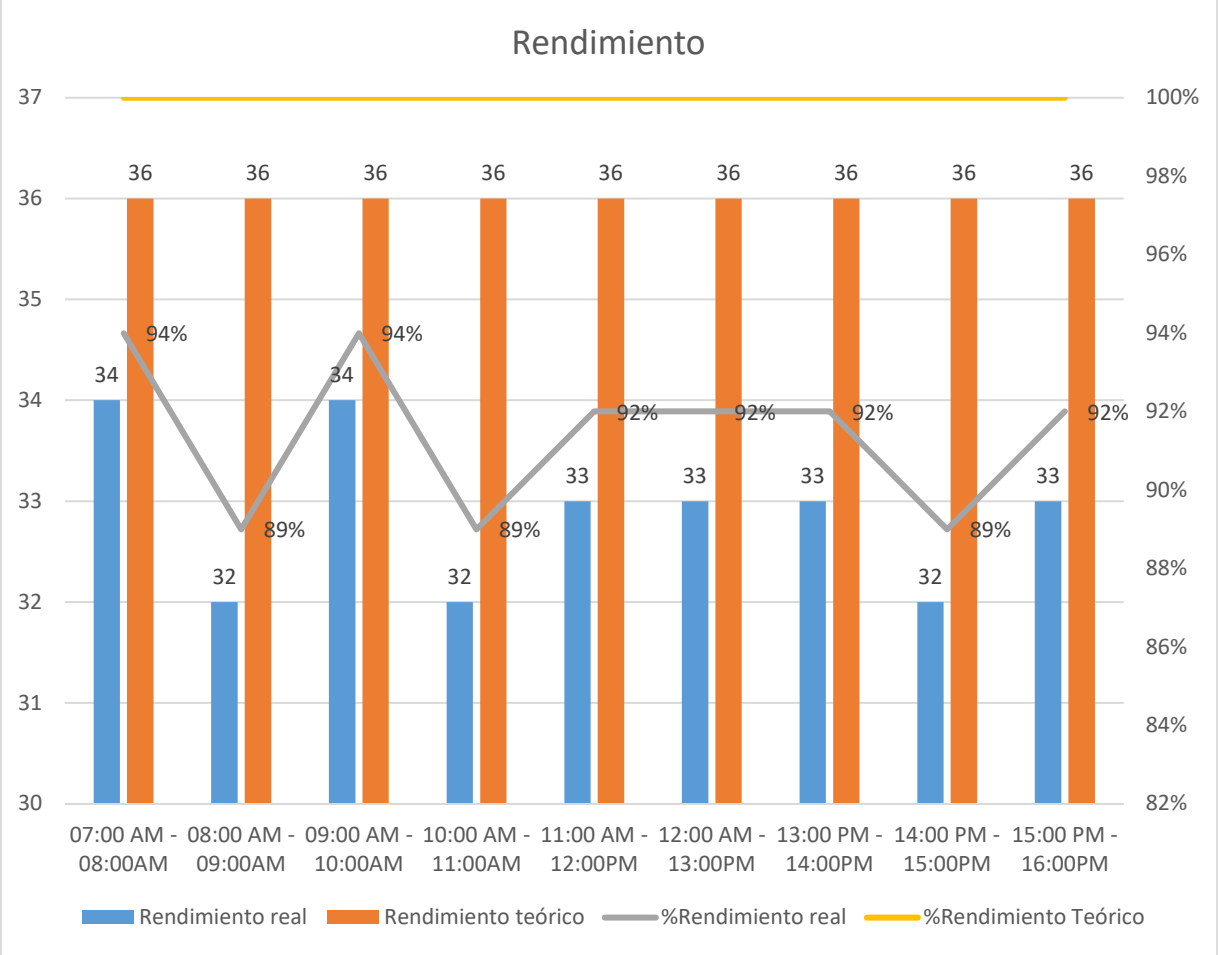


Figura 5-2 Gráfica de Rendimiento

En la Figura 5-2 se observa que la barra color naranja tiene como objetivo que se produzcan 36 unidades por hora teóricamente, en la barra color azul se aprecia el rendimiento real por hora, la línea color amarilla representa el 100% del rendimiento teórico y la línea color gris representa el porcentaje real del rendimiento por hora, dando como promedio de 32.98 unidades producidas por cada 60 minutos y un rendimiento de 91% por turno.

5.2.3 Calidad

La aplicación recopiló información de calidad (Tabla 5.3), en la tabla se muestra calidad de las unidades producidas por hora y el porcentaje de calidad del equipo, en promedio la calidad real es de 31.56 unidades conformantes por hora y porcentaje de calidad es del 96%.

Tabla 5.3 Calidad por hora en un turno

Hora	Calidad real	Calidad teórica	% Calidad Real	% Calidad teórica	Fallas
07:00 AM - 8:00AM	34	34	100%	100%	0
08:00 AM - 9:00AM	32	32	100%	100%	0
09:00 AM -10:00AM	31	34	91%	100%	3
10:00 AM - 11:00AM	30	32	94%	100%	2
11:00 AM - 12:00PM	32	33	97%	100%	1
12:00 AM - 13:00PM	31	33	94%	100%	2
13:00 PM - 14:00PM	31	33	94%	100%	2
14:00 PM - 15:00PM	31	32	97%	100%	1
15:00 PM - 16:00PM	32	33	97%	100%	1
Promedio	31.56		96%		

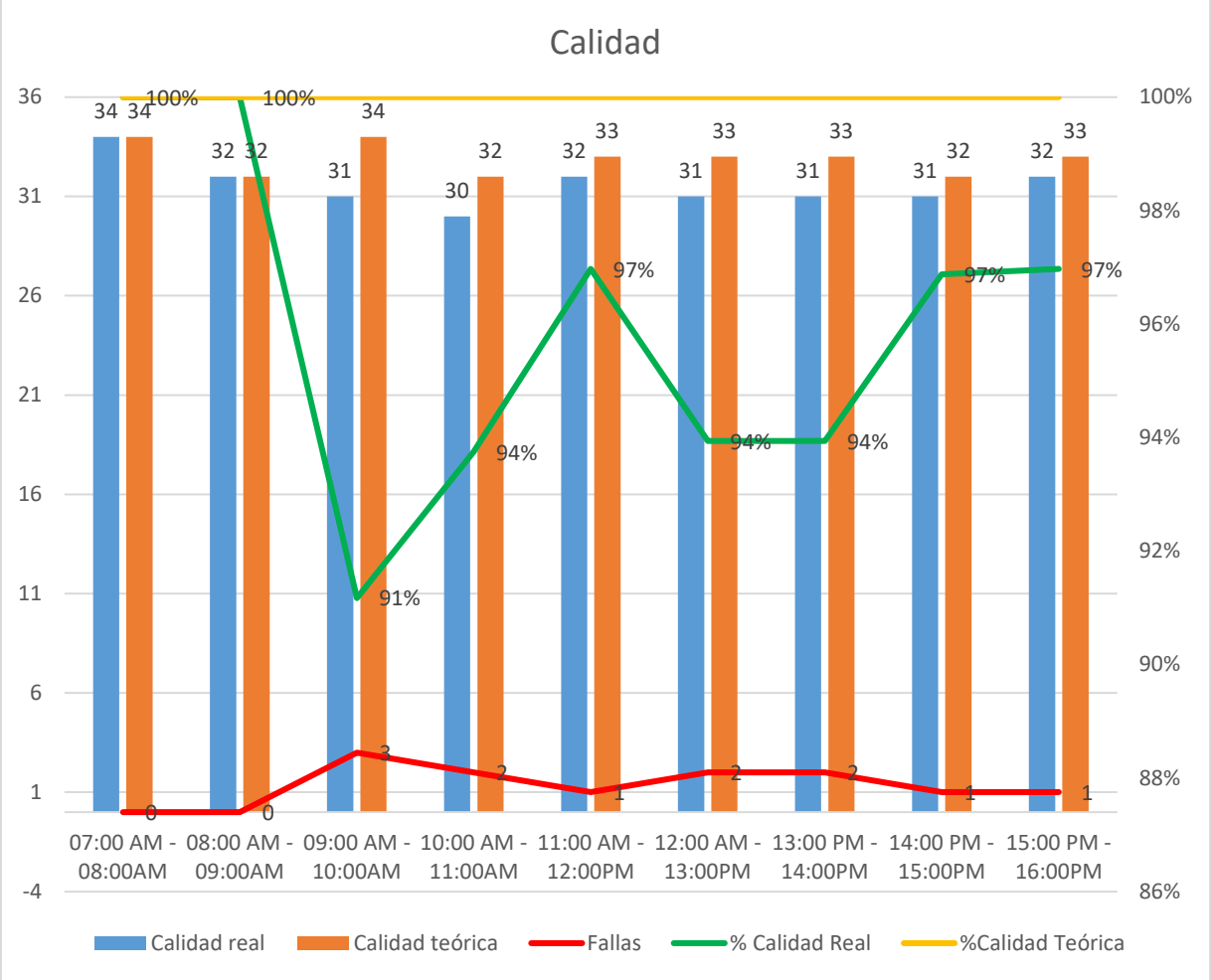


Figura 5-3 Gráfica de Calidad

En la Figura 5-3 se muestra la barra color naranja que se refiere a las unidades probadas por el equipo de ICT donde en teoría deben de ser 100% de unidades conformantes, en la barra color azul se aprecia la calidad real por hora, la barra color naranja representa el total de unidades probadas, la línea color amarilla representa el 100% de calidad teórico y la línea color verde representa el porcentaje real de la calidad por hora, la línea roja representa las fallas por horas en equipo de ICT. Dando como promedio de 31.56 unidades producidas de calidad y 1.33 de unidades falladas por cada 60 minutos y una calidad del 96% por turno de las cuales se produjeron 12 piezas con fallas.

5.2.4 Cálculo del OEE para un equipo de ICT

Con los datos obtenidos los cuales se han recopilado en la Tabla 5.4 por turno por hora y en cada hora de la disponibilidad, rendimiento y calidad se procede hacer el cálculo del OEE por hora.

Tabla 5.4 Tabla de resultado de OEE por Hora

Hora	Disponibilidad	Rendimiento	Calidad	OEE
07:00 AM - 08:00AM	81%	94%	100%	77%
08:00 AM - 09:00AM	79%	89%	100%	71%
09:00 AM - 10:00AM	82%	94%	91%	70%
10:00 AM - 11:00AM	78%	89%	94%	65%
11:00 AM - 12:00PM	80%	92%	97%	71%
12:00 AM - 13:00PM	81%	92%	94%	69%
13:00 PM - 14:00PM	82%	92%	94%	70%
14:00 PM - 15:00PM	77%	89%	97%	67%
15:00 PM - 16:00PM	80%	92%	97%	71%
Promedio	80%	91%	96%	70.11%

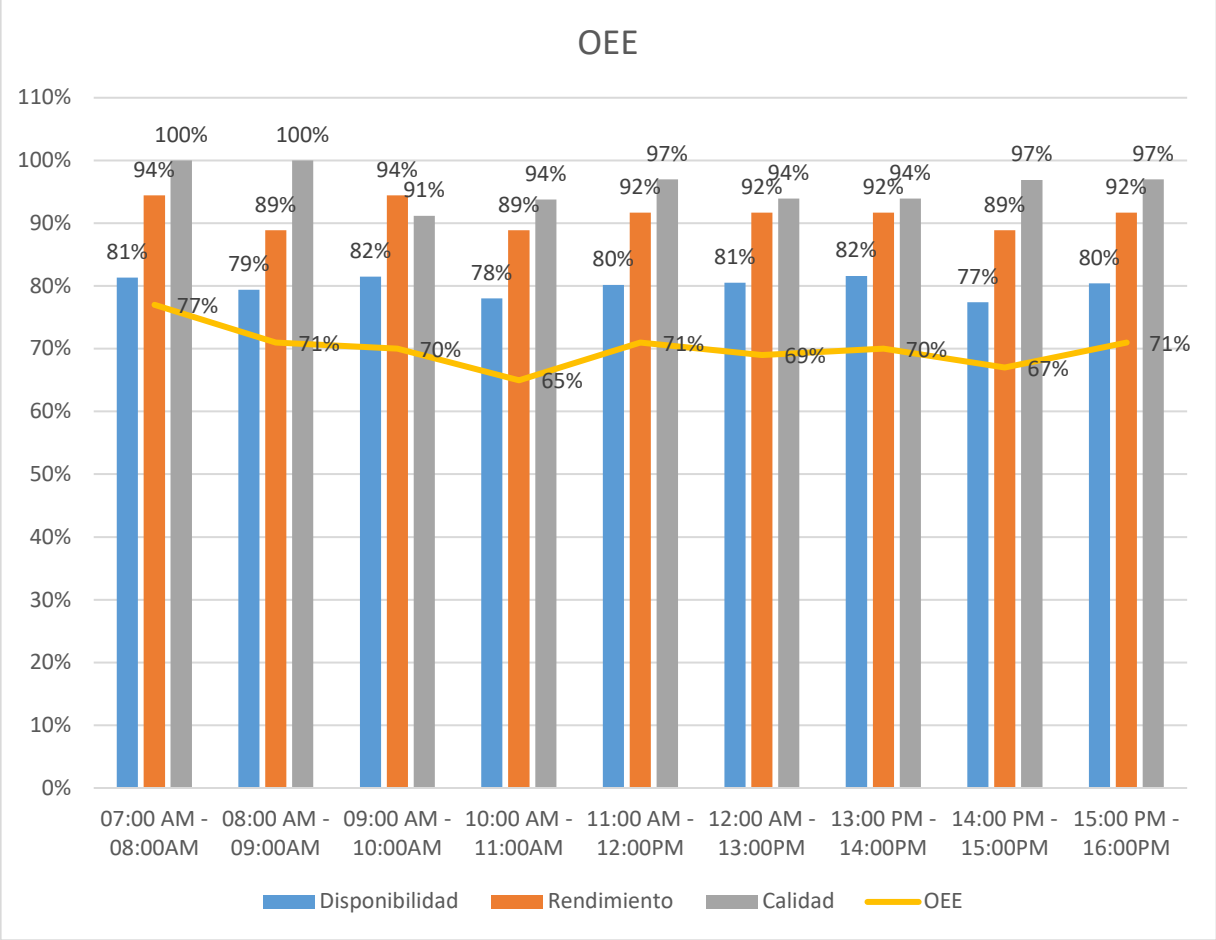


Figura 5-4 Gráfica de OEE por Hora

En la Figura 5.4 se muestra la barra color azul que representa el porcentaje de disponibilidad real del equipo ICT por hora, en la barra color naranja representa el porcentaje de rendimiento real del equipo por hora, en la barra de color gris representa el porcentaje real de calidad del equipo y por último la línea color amarillo representa el OEE del equipo por hora, en promedio se tiene que el equipo tiene un 80% de disponibilidad, un rendimiento de 91% y una calidad del 96% por turno.

5.3 Conclusiones

Después de implementar y llevar acabo el cálculo del OEE para la celda de producción automatizada, se observa que se puede aplicar este módulo en cualquier tipo de celda, siempre y cuando tenga robots colaborativos.

El modelo que se construyó para su aplicación facilita al usuario armar una línea de producción de manera sencilla y simple, esto a su vez también permite que sea adaptable en gran medida a otras celdas y equipos no automatizados siempre y cuando reúnan las condiciones de poderse comunicar a través de los servicios web.

Se concluye que en promedio la disponibilidad es del 80%, el rendimiento es del 94% y la calidad del 96%, (ver tabla 5.4) y que da como resultado un OEE del **70.11%** en promedio el cual tiene un **calificativo de regular**. Significa que la celda de producción tiene pérdidas económicas debido a que no se cumple con la cantidad de piezas conformantes por hora con respecto a lo planeado, sin embargo, es aceptable si se está en proceso de mejora. Se puede decir que el proceso debe mejorar en su disponibilidad y no tener lapsos en los que el equipo no se utilice por avería o por falta de material.

Se observa una oportunidad de mejora con respecto al rendimiento del equipo el cual no trabaja al tiempo estimado asignado por hora, sin embargo, si este tiempo se incrementa, cabe la posibilidad que se vea afectada la calidad de las unidades a producir.

En la actualidad no existe una herramienta con la capacidad de determinar el OEE en tiempo real en celdas automatizadas con robots colaborativos.

5.4 Trabajos futuros

Los datos obtenidos por el software se pueden almacenar en un Big Data el cual tiene como función principal procesar y analizar grandes volúmenes de datos, estos análisis proporcionan y generan información muy valiosa para la toma de decisiones. Con la información generada puede realizarse tambien análisis predictivo, prescriptivo, descriptivo y diagnóstico.

REFERENCIAS BIBLIOGRÁFICAS

Álvarez, H. R. (2018). Stochastic model for Overall Equipment Effectiveness: practical considerations. *Revista Ontare*, 53-86.

Amaya, A., & Rojas, J. (2016). *PROTOCOLOS Y ESTÁNDARES DE TRANSMISIÓN UTILIZADOS POR LOS INSTRUMENTOS INDUSTRIALES.* Barcelona: Inacap.

Aroraa, G., & Dash, T. (2018). *Building RESTful Web Services with .NET Core.*

Baeza, R. (2016). *Plan de mejora en línea de fabricación de producto específico en empresa manufacturera.* Valparaiso: Universidad Tecnológica de Chile.

Camps, R., Casillas, L. A., & Dolors, C. (2005). *Bases de datos.* Cataluña: UOC.

Carretero, J., & Garcia, F. (2004). *Problemas resueltos de programación en lenguaje C++.*

Carretero, J., Garcia, F., & Garcia, J. (2004). *Problemas resueltos de programación en lenguaje C++.*

Cruelles Ruiz, J. A. (2010). *La teoría de la medición del despilfarro.*

Del Val, J. L. (2014). *Industria 4.0: la transformación digital de la industria.* Deusto: Coddinforme.

Dimitrov, K. (2018). *Cyber Defence in Industry 4.0 Systems and Related Logistics and IT Infrastructures.*

Flores, L. (22 de 03 de 2017). *¿CUÁNTOS EMPLEOS EN MÉXICO SERÁN REEMPLAZADOS POR ROBOTS?* Obtenido de expansion.mx:

https://expansion.mx/carrera/2017/03/21/cuantos-empleos-en-mexico-podrian-ser-reemplazados-por-robots

Fowler , M. (1996). *Analysis Patterns: Reusable Object Models .*

Gervais, L. (2017). *Aprender la programación orientada a objetos con el lenguaje C#.*

Greiner, M. (2015). *Overall Equipment Effectiveness (OEE). Approaches for Improvement.*

Guerín, B. (2015). *Gestión de proyectos informáticos: Desarrollo, análisis y control.* Barcelona: Editions Eni.

Gutierrez, C. (2011). *Casos prácticos de UML.*

Hermann, M., Pentek, T., & Otto, B. (2015). Design Principles for Industrie 4 Scenarios: A Literature Review.

Hugon, J. (2015). *C# 6: Desarrolle aplicaciones Windows con Visual Studio 2015.*

HUGON, J. (2015). *C# 6: Desarrolle aplicaciones Windows con Visual Studio 2015.*

ISA–The Instrumentation, S. a. (2001). *ANSI/ISA–95.00.02–2001.* North Carolina: ANSI/ISA–95.00.02–2001.

Kamarul Bahrin, M., Othman, F., Nor Hayati, N., & Muhamad, F. (Jun de 2016). Industry 4.0: a review on industrial automation and robotic.

Kenneth Kennedy, R. (2017). *Understanding, Measuring, and Improving Overall Equipment Effectiveness: How to Use OEE to Drive Significant Process Improvement.*

Landscheidta, S., & Kansb, M. (Noviembre de 2016). Method for Assessing the Total Cost of Ownership of Industrial Robots.

Larman, C. (2003). *UML y patrones : una introducción al análisis y diseño orientado a objetos y al proceso unificado.*

Liebrecht, C., Jacob, A., Kuhnle, A., & Lanza, G. (Marzo de 2017). Multi-criteria Evaluation of Manufacturing Systems 4.0 under Uncertainty.

Linares González, V. (2018). *diagnosis de averías y mantenimiento correctivo de sistemas de automatización industrial.*

Márques, M. (2002). *Bases de datos orientadas a objetos.* Barcelona: UAPA.

Miragliotta, G., Sianesi, A., Convertini, E., & Distante, R. (6 de September de 2018). Data driven management in Industry 4.0: a method to measure Data Productivity.

Mocq, F. (2016). *Raspberry Pi 2.* Barcelona: Eni.

OLLIVIER, S. (2016). *AngularJS: Desarrolle hoy las aplicaciones web de mañana.*

Østergaard, E. (11 de diciembre de 2018). Universal Robots ha vendido 27.000 unidades en sus primeros 10 años en automoción. *Autorevista.com*.

Pelegrí, J. (2018). Cobots vs. Robots industriales: ¿cuáles son las diferencias? *Universal Robots*.

PUTIER , S. (2015). C# 6 y Visual Studio 2015 - Los fundamentos del lenguaje.

Putier, S. (2015). C# 6 y Visual Studio 2015 - Los fundamentos del lenguaje.

Riquelme, R. (8 de 7 de 2017). *México, uno de los más expuestos al trabajo autómata*. Obtenido de https://www.eleconomista.com.mx/empresas/Mexico-uno-de-los-mas-expuestos-al-trabajo-automata-20170708-0003.html

Rita, G., Luca, G., Francesco, L., & Rimini, B. (Julio de 2017). On the Analysis of Effectiveness in a Manufacturing Cell: A Critical Implementation of Existing Approaches.

Soria, B. (13 de 10 de 2016). *Guanajuato, con oportunidades de negocio por 4 mil mdd*. Obtenido de Vanguardia-Industrial: https://www.vanguardia-industrial.net/guanajuato-con-oportunidades-de-negocio-por-4-mil-mdd/

Steenkampa, L., Hagedorn-Hansen, D., & Oosthuizenc, G. (Febrero de 2017). Visual Management System to Manage Manufacturing Resources.