



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
INGENIERIA ELECTRONICA**

TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

TÍTULO:

**Desarrollo de un Framework de MTConnect para Monitoreamiento Remoto de Máquinas de
Manufactura Aditiva RepRap Vía Internet**

Autor:

Juan Pablo Rodriguez Gasca

Director:

Ph.D. César A. Peña Cortés

Co-Director:

M.Sc. Efrain A. Rodriguez (Universidad de Brasilia, Brasil)

PAMPLONA-COLOMBIA

JULIO de 2019



**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA, SISTEMAS Y
TELECOMUNICACIONES
INGENIERÍA ELECTRÓNICA**

TRABAJO DE GRADO PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

TÍTULO:

**Desarrollo de un Framework de MTConnect para Monitoreamiento Remoto de Máquinas de
Manufactura Aditiva RepRap Vía Internet**

Autor:

Juan Pablo Rodriguez Gasca

Director:

Ph.D. César A. Peña Cortés

Co-Director:

M.Sc. Efrain A. Rodriguez (Universidad de Brasilia, Brasil)

JURADO CALIFICADOR:

Ing. Víctor Julio Vargas Sarmiento

Ing. Diego Alfonso Peláez Carrillo

PAMPLONA-COLOMBIA

JULIO de 2019

AGRADECIMIENTOS

Quiero agradecer primeramente a Dios por darme la fortaleza para llegar a este punto de mi vida donde me encuentro hoy y a mi familia por haberme apoyado para que pudiera alcanzar este logro, en especial mis padres Walter y Julia quienes han luchado incansablemente por sacarme adelante. Agradecer a mis hermanos Efrain, Mariannys y Lole por la ayuda y el apoyo que siempre me han brindado.

También quiero agradecer a mis amigos Oscar Almanza, Carlos Adrian, Edgar Medina, Julian Arias, Edwin Gonzales por el apoyo que me brindaron durante mi carrera en especial a Johana Henao por haberme aconsejado y ayudado para alcanzar tantas veces.

RESUMEN

El estándar MTConnect proporciona un puente para llevar los recursos de fabricación física en red a los entornos de fabricación en la nube. Sin embargo, hacer que los equipos de fabricación heredados cumplan con esta nueva norma sigue siendo una tarea difícil. De hecho, con respecto a los procesos de fabricación aditiva, este problema se encuentra en una etapa incipiente. Para abordar el problema, este trabajo presenta un marco compatible con MTConnect para el acceso a datos de planta y el monitoreo de las máquinas de fabricación aditiva RepRap basadas en los controladores de tecnología abierta Arduino. Se incorporó un canal de comunicación basado en el protocolo TCP / IP que utiliza un módulo Ethernet en el sistema de la máquina para ampliar las funcionalidades de las impresoras 3D RepRap para conectarse a Internet. También se desarrolló una aplicación de cliente web para validar el sistema mediante la supervisión de la máquina en tiempo real a través de Internet. La solución MTConnect propuesta permite recuperar datos de la posición de los ejes, la temperatura de la cabecera, la temperatura del extremo caliente, la extrusión de material y el tiempo transcurrido desde la aplicación web, lo que demuestra la viabilidad de operar en entornos de fabricación de nubes.

ABSTRACT

The MTConnect standard provides a bridge to bring networked physical manufacturing resources to cloud manufacturing environments. However, making legacy manufacturing equipment compliant with this new standard is still a difficult task. Indeed, regarding the additive manufacturing processes, this issue is in an incipient stage. To address the issue, this work introduces an MTConnect-compliant framework for shopfloor data access and monitoring of RepRap additive manufacturing machines based on Arduino open technology controllers. A communication channel based on TCP/IP protocol using an Ethernet module was incorporated into the machine system to extend the functionalities of RepRap 3D printers to connect to the Internet. Two MTConnect implementation architectures named Type 1 and Type 2 were proposed. A web-client application was also developed to validate the system by performing real-time machine monitoring over the Internet. The proposed MTConnect solution allows retrieving data of axes position, headbed temperature, hotend temperature, material extrusion, current layer and elapsed time from the web application, demonstrating feasibility to operate in cloud manufacturing environments.

CONTENIDO

RESUMEN.....	4
ABSTRACT	4
1. INTRODUCCIÓN.....	11
1.1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN	11
1.2. OBJETIVOS	13
1.2.1. OBJETIVO GENERAL.....	13
1.2.2. OBJETIVOS ESPECIFICOS	13
2. ESTADO DEL ARTE.....	15
2.1. EL NUEVO PANORAMA INDUSTRIAL – INDUSTRIA 4.0	15
2.1.1. La evolución de la máquina-herramienta.....	18
2.1.2. Sistemas ciber-físicos (CPS-Cyber-Physical Systems).....	19
2.1.3. Manufactura aditiva	20
2.1.4. El protocolo MTConnect.....	21
2.2. TRABAJOS RELACIONADOS	23
3. METODOLOGÍA DE DISEÑO DEL FRAMEWORK.....	24
3.3. Paso 3: Priorización de los RUs.....	25
3.4. Paso 4: Selección de los parámetros de diseño.	25
3.5. Paso 5: Relacionamiento entre URs y PDs.....	26
4. DESARROLLO DEL FRAMEWORK	28
4.1. ARQUITECTURA DEL PROTOCOLO MTCONNECT A DESARROLLAR	29
4.2. AGENTE MTCONNECT EMPLEADO.....	30
4.2.1. Agente MTConnect C++	30
4.2.2. Instalación y configuración del agente	31
4.2.3. Parámetros de configuración para el agente	34
4.2.4. Protocolo de comunicación entre el agente y el adaptador.....	36
4.3. DESARROLLO DEL ADAPTADOR MTCONNECT	36
4.3.1. Hardware del adaptador	36
4.3.2. Software del adaptador.	37
4.3.3. Configuración del ENC28J60	37
4.3.4. Comunicación entre el adaptador y el agente.	39
4.4. CLIENTE WEB PARA MTCONNECT	42

4.4.1.	Descripción de la interfaz	42
4.4.2.	Diagrama de flujo para el cliente web.	45
5.	RESULTADOS: EXPERIMENTOS Y VALIDACIÓN	48
5.1.	VELOCIDAD DE COMUNICACIÓN	50
5.1.1.	Comunicación Adaptador–Agente.	50
5.1.2.	Comunicación Agente-Cliente web	52
5.1.3.	Sincronización entre el adaptador y el agente	53
6.	CONCLUSIONES Y TRABAJOS FUTUROS	55
	REFERENCIAS BIBLIOGRÁFICAS	57
	ANEXOS.....	60

LISTA DE FIGURAS

Figura 1. Las cuatro revoluciones industriales. Adaptador de [6].....	16
Figura 2. Modelo de Arquitectura de Referencia para la Industria 4.0 (RAMI 4.0) [7].	17
Figura 3. Evolución de la máquina-herramienta.....	18
Figura 4. Representación del proceso FDM [27].....	21
Figura 5. Arquitectura de un sistema de manufactura complaciente con MTConnect.	22
Figura 6. Diagrama de Pareto para el análisis de RUs.	26
Figura 7. Matriz de casa de la calidad del QFD con relacionamientos entre RUs y PDs..	27
Figura 8. Estructura para comunicación de datos usando MTConnect.....	29
Figura 9. Modelo de información para el agente MTConnect.	30
Figura 10. Instalación del agente.....	31
Figura 11. Configuración del agente.	32
Figura 12. Configuración del agente.	32
Figura 13. Selección del modelo para el adaptador XML.	33
Figura 14. Administrador de servicios de Windows.	33
Figura 15. Archivo Agent.cfg editado.	34
Figura 16. Conexión ENC28J60 y ramps 1.4	37
Figura 17. Arquitectura de comunicación entre adaptador y agente.	37
Figura 18. Parámetros de configuración para el módulo ENC28J0.	38
Figura 19. Inicio del módulo Enc28j60.....	38
Figura 20. Configuración del timer 5.	40
Figura 21. Función ISR del timer 5.....	40
Figura 22. Funciones para obtener datos de los sensores y cargar mensajes en el buffer del módulo ethernet.	41
Figura 23. Interfaz de usuario para el cliente web.	42
Figura 24. Barra de navegación para la interfaz del cliente web.....	43
Figura 25. División detalle en la interfaz del cliente web.	43
Figura 26. División estado del proceso en la interfaz cliente web.....	43
Figura 27. División tele operación en el cliente web.....	44
Figura 28. División graficas en el cliente web.....	44
Figura 29. División datos en el cliente web.....	45
Figura 30. Diagrama de flujo para la programación del cliente.	46
Figura 31. Respuesta XML por el agente para el navegador web.	48
Figura 32. Respuesta XML sample devuelta por el agente para el navegador web.	49
Figura 33. Respuesta XML current devuelta por el agente para el navegador web.	49
Figura 34. Datos vistos desde el cliente web para la impresión de la pieza.	50
Figura 35. Paquete con PING del agente al adaptador.	50
Figura 36. Paquete con PONG del adaptador al agente.	51
Figura 37. Primer paquete de datos entre el agente y el adaptador.....	51
Figura 38. Segundo paquete de datos entre el adaptador y el agente.....	51
Figura 39. Grafica para la diferencia de tiempo entre datos para la comunicación adaptador-agente.	52

Figura 40. Solicitud del cliente web para agente vista desde el PC del cliente.	52
Figura 41. Respuesta del agente al cliente web vista desde el PC del cliente.	53
Figura 42. Fallo de sincronización entre los datos del agente y el cliente.	53

LISTA DE TABLAS

Tabla 1. Lista de requerimientos de usuario para el diseño del framework.....	25
Tabla 2. Parámetros de diseño seleccionados para el framework.....	26
Tabla 3. Especificaciones-meta y tendencias para las características del framework.	28
Tabla 4. Métodos y parámetros para la configuración del ENC28J60.	39
Tabla 5. Funciones para la comunicación con el agente mediante el módulo del ENC28J60.	41

1. INTRODUCCIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

En la última década, los enormes avances en tecnologías de información y comunicación (TICs) tales como Internet de las Cosas (IoT-Internet of Things), Sistemas Ciber-Físicos (CPS-Cyber-Physical Systems), Computación en la Nube (Cloud Computing), Datos Masivos (Big Data), Inteligencia Artificial, etc. han reformado la industria manufacturera y provocado la cuarta revolución industrial [1]. Tal reforma está cambiando principalmente los modelos de negocio de las compañías, yendo desde manufactura orientada-a-producción a manufactura orientada-a-servicio y desde producción de productos en masa a personalización de productos en masa, todo a fin de tornarlas más ágiles y costo-eficientes para mantenerse competitivas en el mercado global.

Este nuevo movimiento industrial ha captado importante atención de muchos gobiernos, los cuales han comenzado a desplegar sus planes de desarrollo tecnológico para tomar ventaja de lo que esto pueda brindar, por ejemplo: “Industry 4.0” en Alemania 2011; “Industrial Internet” en Estados Unidos 2012, “Made in China 2025” en China 2015; “Agenda para a Indústria 4.0” en Brasil 2017. Mientras tanto en Colombia, los ecos de cambio por la nueva revolución industrial apenas comienzan a percibirse. Consecuentemente, muchos grupos de investigación y líderes industriales de todo el mundo se han juntado a estas iniciativas de desarrollo con el objetivo en común de lograr sistemas de manufactura altamente inteligentes y autónomos, dando lugar a nuevos paradigmas tecnológicos como Cloud Manufacturing y Smart Manufacturing.

En este nuevo ambiente de manufactura Smart y Cloud, las máquinas de control numérico computarizado (CNC) son el elemento central, lo cual demanda mayores capacidades en términos de flexibilidad, interoperabilidad, conectividad, agilidad y alto nivel inteligencia. De acuerdo con Liu & Xu [2], esta nueva era industrial global lleva a la necesidad de desarrollar una nueva generación de máquinas CNC que sean más inteligentes, bien conectadas, ampliamente accesibles, más adaptables y más autónomas. No obstante, la conexión y acceso de las máquinas CNC vía Internet es un apremiante desafío que aún debe ser resuelto. En este sentido, la limitación que existe en la integración y gerenciamiento de la información de proceso, debido a la variedad de formatos de datos propietarios adoptados por los sistemas de adquisición de datos y fabricantes de controladores CNC, representa el principal obstáculo a ser enfrentado.

Afortunadamente, un nuevo protocolo de comunicación estándar, conocido como MTConnect (Machine Tool Connection) [3], ha sido recientemente desarrollado para promover el intercambio de datos de manera estandarizada entre equipos-dispositivos de manufactura y aplicaciones de software basadas en PC (Personal Computer) y/o en la Web. MTConnect representa el eslabón para conectar máquinas CNC a Internet, y eventualmente a los ambientes de Cloud Manufacturing.

Sin embargo, actualizar viejas máquinas CNC con una solución de MTConnect se torna en muchos casos una tarea complicada, sobre todo cuando se trata con máquinas basadas en tecnologías cerradas o licenciadas. En relación a esto, el caso de las máquinas CNC de

manufactura aditiva (AM-Additive Manufacturing, Impresión 3D) del tipo RepRap (acrónimo de Replicating Rapid Prototype) equipadas con controladores de arquitectura abierta (OAC-Open Architecture Control) son factibles para ser rápidamente actualizadas con MTConnect debido a que su concepción facilita su upgrade con nueva tecnología. Usualmente, estas máquinas de AM o impresoras 3D RepRap poseen controladores basados en tecnología abierta de Arduino, la cual es caracterizada por ser de fuente abierta (open source), de bajo costo, fácil uso, y de arquitectura de software y hardware extensibles. Este tipo de máquinas se ha tornado muy atractivo especialmente para pequeñas y medianas empresas, y laboratorios de investigación de universidades por su óptima relación entre precio y rendimiento. De hecho, un estudio estadístico ha mostrado que las máquinas RepRap son las más usadas por usuarios de la impresión 3D alrededor del mundo [4].

Con eso en mente, el presente trabajo introduce el desarrollo de un framework de sistema ciber-físico basado en MTConnect para monitoreamiento remoto en tiempo real de máquinas de manufactura aditiva RepRap equipadas con controlador de tecnología Arduino. Este trabajo se da en el contexto de una nueva revolución industrial que trae como importantes desafíos la producción sobre-demanda y la personalización de productos en masa. En un momento donde en Colombia los aires de esta nueva revolución industrial se están empezando a sentir, este trabajo pretende fomentar e motivar a las compañías, grupos de investigación y agencias gubernamentales al desarrollo de nuevas tecnologías asociadas a la Industria 4.0, que contribuyen a las empresas manufactureras a mejorar la eficiencia de los procesos y la calidad de los productos. Además, este trabajo desea significar un aporte al crecimiento del programa de Ingeniería Electrónica de la Universidad de Pamplona, al abarcar tópicos de gran relevancia actualmente como son IoT, CPS, Cloud Manufacturing y Manufactura Aditiva.

Adicionalmente, esta investigación hace parte de un trabajo colaborativo entre los laboratorios MMI (Mecánica, Mecatrónica e Industrial) de la Universidad de Pamplona y el LaDPRER (Laboratório de Desenvolvimento de Produto e Engenharia Reversa - <http://ladprer.alvarestech.com/>) de la Universidad de Brasilia. A su vez, este trabajo colaborativo hace parte del proyecto “Integrated and Intelligent Advanced Manufacturing Systems” liderado por el Profesor Alberto J. Alvares dentro del GIAI (Grupo de Inovação em Automação Industrial - <http://www.giai.unb.br>) en la Universidad de Brasilia. Proyecto que envuelve: sistemas avanzados de manufactura incluyendo manufactura aditiva e substractiva, manufactura híbrida, conformado incremental, y inspección en lazo cerrado; integración de la cadena digital de los procesos usando los estándares STEP-NC (Standard for Exchange of Product model data – Numerical Control) y QIF (Quality Information Framework); y, el desarrollo de sistemas ciber-físicos y sistemas de manufactura inteligentes basados en tecnologías de MTConnect, OPC UA (OPC Unified Architecture) y sistemas de inteligencia artificial.

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Concebir y desarrollar un framework de sistema ciber-físico para monitoreamiento remoto de máquinas de manufactura aditiva del tipo RepRap vía internet usando el protocolo MTConnect dentro del contexto de la Industria 4.0.

1.2.2. OBJETIVOS ESPECIFICOS

Para atender al objetivo general del presente trabajo, han sido planteados los siguientes objetivos específicos:

- Realizar un proceso de revisión bibliográfica asociando la fundamentación teórica de los diferentes tópicos y/o conceptos tratados y los principales trabajos relacionados.
- Adecuar una metodología para el desarrollo de aplicaciones de software para la sistematización del levantamiento de los requerimientos de diseño de la plataforma ciber-física.
- Desarrollar un adaptador MTConnect basado en software y hardware, de modo que pueda ser embarcado en una máquina de manufactura aditiva/impresora 3D con un controlador de tecnología abierta Arduino.
- Integrar el adaptador con un agente MTConnect para disponibilizar los servicios de monitoreamiento de la máquina a través de una interfaz de usuario en la Web.
- Validar la plataforma a través de la implementación y realización de pruebas de monitoreamiento sobre una impresora 3D.

1.3. METODOLOGÍA DE INVESTIGACIÓN

De acuerdo con Gil [5], el trabajo de investigación puede ser clasificado a partir de dos puntos de vista:

En cuanto a los objetivos – la investigación puede ser de tipo: exploratoria, descriptiva o explicativa. La investigación abordada en este trabajo encaja dentro del tipo de investigación exploratoria, ya que busca el desarrollo de un sistema ciber-físico basado en MTConnect para monitoreamiento remoto de impresoras 3D. El MTConnect es un nuevo protocolo de comunicación que habilita la conexión de máquinas CNC a la Internet para publicar datos de proceso sobre una aplicación web. En este trabajo, se explora el uso de dicho protocolo de comunicación para la adquisición de datos de proceso de manufactura aditiva desde Internet para aplicaciones de monitoreamiento remoto. Esto envuelve el desarrollo de un componente de software Adaptador de MTConnect, la integración entre el Adaptador y un software Agente de MTConnect, y la validación de todo el sistema a través de una aplicación web también desarrollada como parte de este trabajo. Como resultado final se tiene un prototipo de plataforma ciber-física para monitoreamiento remoto en manufactura aditiva usando máquinas del tipo RepRap.

En cuanto a los procedimientos técnicos - una investigación puede ser: bibliográfica, documental, experimental, de levantamiento o de investigación-acción. En relación a esta clasificación, la investigación descrita en este trabajo puede ser considerada como bibliográfica, dado que incluye una fundamentación teórica basada en una revisión de bibliografía sobre los principales tópicos envueltos no trabajo.

2. ESTADO DEL ARTE

Esta sección tiene como objetivo introducir al lector en los tópicos tratados y los principales trabajos relacionados.

2.1. EL NUEVO PANORAMA INDUSTRIAL – INDUSTRIA 4.0

Desde el momento en que Prometeo robó el fuego a los dioses en el tallo de una cañaheja y lo entregó a la humanidad, ésta no ha parado de usarlo para crear extraordinarias invenciones a lo largo de su historia.

Durante el transcurso de su evolución, el hombre ha creado, dominado y perfeccionado una gran variedad de tecnologías de fabricación con el fin de procesar la materia prima que la naturaleza le proporciona para su bienestar. El desarrollo de estas tecnologías le ha permitido trabajar sobre diferentes tipos de materiales incluyendo metales, madera, polímeros, cerámicas, etc. y con esto construir todo tipo de artefactos y herramientas. Consecuentemente, los avances tecnológicos en el campo de la manufactura y la producción, apoyados por los desarrollos en otras áreas como la física, química y medicina, han ocasionado cambios abruptos en términos económicos, sociales y políticos en todo el mundo.

Hoy en día, existe un consenso sobre cuatro grandes cambios sufridos por el sector de la manufactura y los sistemas de producción industrializados, y que han representado verdaderas revoluciones industriales. La Figura 1 ilustra una línea de tiempo de las cuatro revoluciones industriales sucedidas hasta la fecha y lo que caracteriza a cada una de ellas. La primera tuvo lugar entre finales del siglo XVIII e inicios del siglo XIX con el surgimiento de los primeros sistemas mecánicos impulsados a vapor y el primer telar basado en mecanismos. La segunda revolución, también conocida como revolución tecnológica, la cual data del periodo entre 1870 y 1914, es caracterizada por la introducción de la electricidad y los sistemas de producción en línea, además de significantes innovaciones en el campo de la manufactura tales como el establecimiento de la industria de máquinas-herramientas, el desarrollo de métodos de montaje de partes mecánicas intercambiables y nuevos procesos para manejo del acero. Ya la tercera de las revoluciones, tuvo su comienzo en la década de los 70s con el desarrollo de los materiales semiconductores y el incremento en la adopción de la microelectrónica en sistemas de automatización industrial. Esta tercera revolución también estuvo marcada por el surgimiento de los primeros mainframes, los primeros computadores personales y los inicios de Internet.

La cuarta revolución industrial, comúnmente referida como Industria 4.0, comenzó a finales de la década pasada con la iniciativa de estrategia alemana que objetiva el establecimiento de las fábricas inteligentes (Smart Factories) soportadas sobre pilares tecnológicos emergentes como los sistemas ciber-físicos, el internet de las cosas y la computación en la nube. En esta secuencia, varios países comenzaron a sumarse a esta causa entre los cuales pueden resaltarse EEUU, Francia, China, Japón, Brasil, entre otros. Industria 4.0 promueve la integración de sistemas avanzados de manufactura y tecnologías de información y comunicación para lograr sistemas de producción en red fundamentados sobre la hiper-conectividad y colaboración en tiempo real entre humanos, máquinas, sensores, interfaces hombre-máquina (HMI), etc.

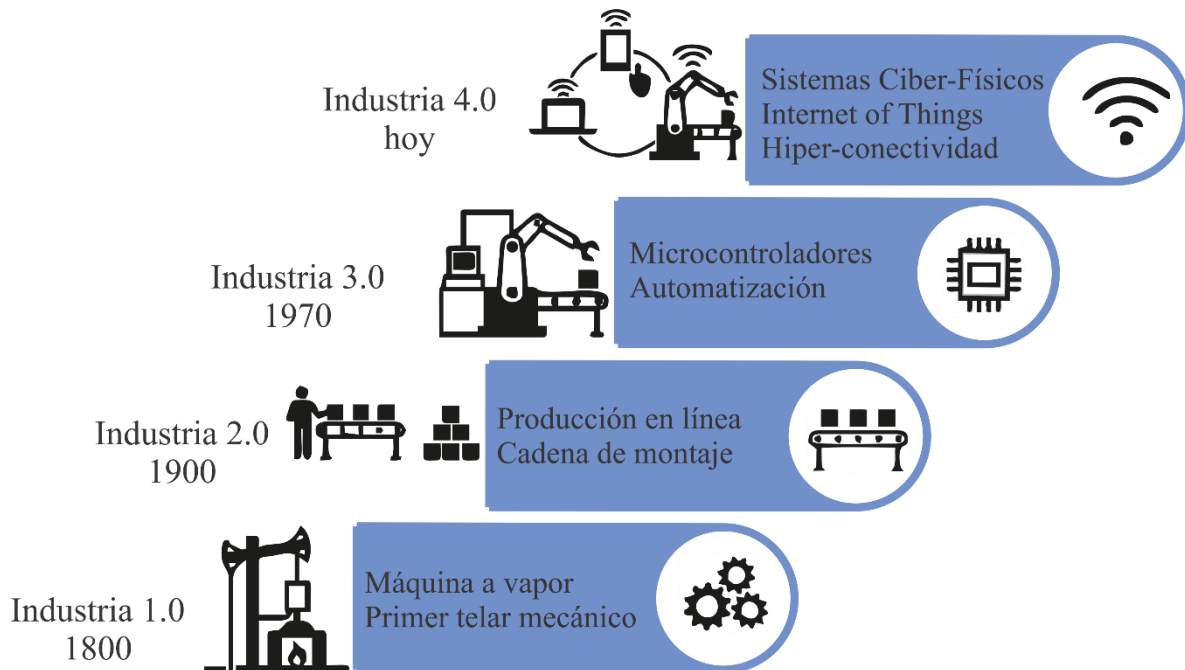


Figura 1. Las cuatro revoluciones industriales. Adaptador de [6]

Desde el punto de vista de la Industria 4.0, los sistemas de manufactura son llevados a un nivel superior de mayor inteligencia, flexibilidad, agilidad y autonomía aprovechando los potenciales beneficios ofrecidos por las TICs en orden a enfrentar las heterogéneas y dinámicas demandas del mercado global. Así, productos de mayor calidad y altamente personalizados, procesos más eficientes y sustentables, y nuevos modelos de negocio más rentables, son esperados con la llegada de esta nueva revolución que tendrá un impacto directo sobre la economía de los gobiernos, las empresas y por su supuesto en la de vida de las personas.

Debido al rápido esparcimiento e incremento en la aceptación de los nuevos conceptos tecnológicos promovidos por la Industria 4.0, recientemente ha sido presentado un nuevo modelo de arquitectura de referencia que proporciona una directriz para el entendimiento común de las tecnologías y estándares envueltos por la Industria 4.0. La Figura 2 ilustra tal modelo de referencia, denominado RAMI 4.0 (Reference Architecture Model Industrie 4.0) [7], el cual describe las interrelaciones entre las TICs, el ciclo de vida de los productos y las compañías/manufactureros a través de un mapa tridimensional formado por capas, cadena digital y niveles jerárquicos.

Las seis capas sobre el eje vertical (ver Figura 2) describen la descomposición de una máquina en sus propiedades de manera estructurada capa por capa, es decir, el despliegue virtual de una máquina. El eje horizontal izquierdo representa el ciclo de vida del producto, considerando principalmente su cadena digital, estandarizado por la norma IEC 62890 para el gerenciamiento de ciclo de vida. Posteriormente, se hace una distinción entre “Tipos” e “Instancias”, donde un “Tipo” pasa a ser una “Instancia” cuando constituye el producto propiamente dicho luego de salir de las fases de diseño y prototipado. Finalmente, sobre el eje horizontal derecho se indican los niveles jerárquicos normalizados por IEC 62264/61512, los cuales son estándares internacionales para los sistemas de control y tecnologías de información de las empresas.

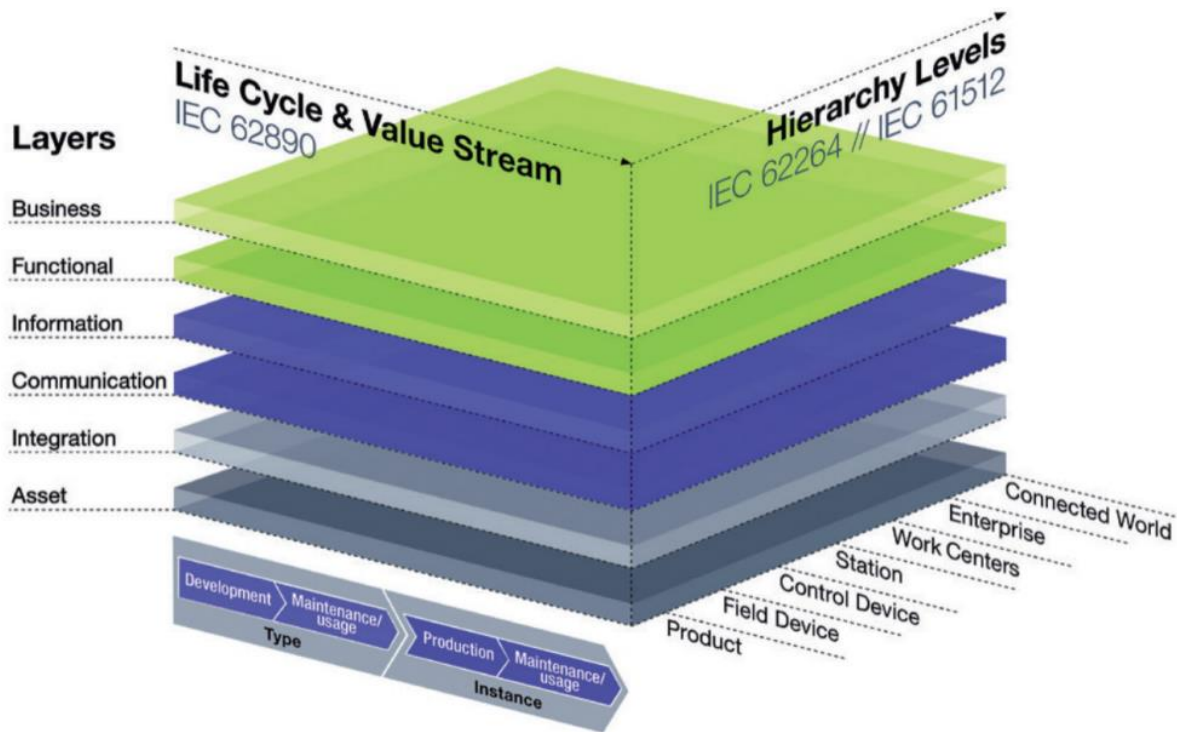


Figura 2. Modelo de Arquitectura de Referencia para la Industria 4.0 (RAMI 4.0) [7].

Sobre la dimensión de niveles jerárquicos del modelo RAMI 4.0, estándares industriales como MTConnect y OPC-UA (Open Platform Communications-Unified Architecture) proveen tanto los modelos de información como los protocolos de comunicación que permiten llevar datos de máquinas/dispositivos, productos y procesos desde el nivel de planta de la fábrica hasta el internet. Es decir, digitalizar los estados y comportamientos de todas las entidades físicas disponibles en la planta de la fábrica, y que son capaces de generar datos, para consolidar un sistema de producción ciber-físico, donde el modelo digital o ciber (Digital Twin) de la fábrica refleja el comportamiento de su equivalente físico.

Eventualmente, el modelo digital, idealmente soportado sobre la web, es usado para trabajar sobre monitoreamiento y operación remota de los dispositivos de campo, así como para optimizar las condiciones y parámetros del proceso aplicando técnicas de inteligencia artificial en pro de mejorar el rendimiento de las máquinas y los procesos, y aumentar la calidad de los productos con mayor capacidad de sustentabilidad y flexibilidad. Esto abre nuevas posibilidades de modelos de negocios basados en manufactura a distancia soportada en la nube (cloud manufacturing), donde los recursos físicos de manufactura pueden accedidos por los usuarios en diferentes localizaciones geográficas como servicios vía internet, habilitando sistemas de producción altamente colaborativos y de mayor rentabilidad para las empresas.

En este contexto, considerando que la máquina-herramienta es el elemento central de cualquier sistema de manufactura, cabe definir lo que es una máquina-herramienta ciber-física dentro del ambiente de manufactura en la nube de la Industria 4.0.

2.1.1. La evolución de la máquina-herramienta

Es sabido que las máquinas-herramienta han existido desde mucho antes de la primera revolución industrial; no obstante, estas han experimentado los cambios más significativos de su evolución precisamente con el acontecimiento de cada revolución industrial. Han pasado de ser enteramente operadas por humanos, a tener potentes controladores CNC capaces de realizar tareas complejas de manufactura de manera casi autónoma, lo cual ha mejorado sustancialmente la eficiencia de los procesos y la calidad de los productos fabricados. Una línea de tiempo de su evolución es ilustrada en la Figura 3.

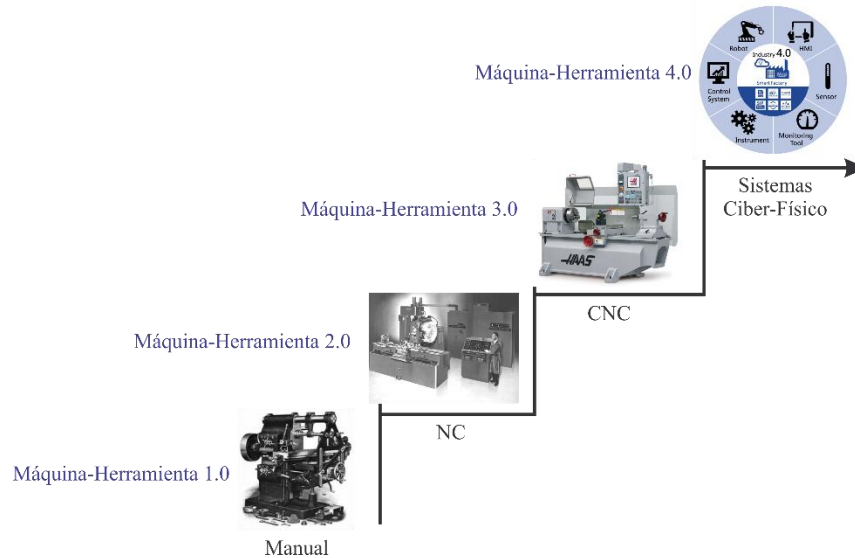


Figura 3. Evolución de la máquina-herramienta.

Durante el desarrollo de la primera revolución industrial, las primeras máquinas-herramientas ofrecidas para venta aparecieron en el mercado en Inglaterra a finales de 1700 – Máquinas-Herramientas 1.0. Permitieron que la producción de partes mecánicas intercambiables fuera una realidad. Estas primeras máquinas eran operadas manualmente por humanos, con algunas formas de asistencia mecánica para aumentar la precisión, por lo que exigían un alto grado de habilidad y experiencia del operario. El rápido avance de las máquinas-herramientas en términos de estandarización y calidad fue impulsado por la industria automotora que dominaba parte del desarrollo tecnológico de esa época.

Uno de los cambios más significantes en el sector de las máquinas-herramientas tuvo lugar a mediados del siglo XX con la incorporación de Control Numérico (NC) para control de actuación y movimiento – máquinas-herramientas 2.0. El primer controlador numérico fue desarrollado en 1940 en el MIT (Massachusetts Institute of Technology), el cual proporcionaba control automático del movimiento de la pieza en relación a la herramienta de corte en una máquina-herramienta [2]. Sin embargo, estos controladores tenían sus limitaciones principalmente en términos de capacidad de memoria para almacenamiento de programas al ser basados en cintas/tarjetas perforadas [8]. A pesar de esto, en comparación son las máquinas-herramientas convencionales operadas manualmente, los controladores numéricos de esa época significaron un enorme avance para el mejoramiento del rendimiento de los procesos de mecanizado, ahorrando tiempo y esfuerzo de trabajo, así como ayudando a mejorar la calidad del producto final.

Alrededor de los años 70s, el avance de los computadores ya había madurado en gran parte, lo cual auspició el nacimiento de los primeros controladores numéricos computarizados (CNC) para máquinas-herramientas – máquinas-herramientas 3.0. Estos controladores llegan con la capacidad para almacenar uno o más programas de un proceso, además de encargarse de la operación del sistema de movimiento y otras funciones de las máquinas [9]. Entonces, la introducción de máquinas-herramientas con controladores CNC transformó radicalmente el sector de la manufactura reduciendo dramáticamente la intervención de personas en el proceso al automatizar un gran puñado de las tareas de fabricación. Además, abrió nuevas posibilidades para la manufactura de partes con geometrías complejas como curvas y trayectorias tridimensionales. Esta fase de las máquinas-herramientas también motivó al desarrollo de los sistemas computacionales para diseño, planeamiento y manufactura auxiliadas por computador (CAD-CAPP-CAM, Computer Aided Design-Process Planning-Manufacturing) [10].

Hoy en día, la llegada de una nueva revolución industrial trae consigo la necesidad de progreso por parte de las máquinas-herramientas actuales; es decir, la versión de máquinas-herramientas 4.0 de la Industria 4.0. Estas nuevas máquinas deberán, esencialmente, aumentar su capacidad de conexión sobre internet y tener un alto nivel de autonomía e inteligencia para tomar decisiones sobre la tarea que están realizando.

Liu y Xu [2] han introducido un concepto para la Máquina-Herramienta 4.0, la cual debe ser más inteligente, bien conectada, ampliamente accesible, y más autónoma y adaptativa. Ellos argumentan que esto puede ser logrado con la integración e implementación de tecnologías de sistemas ciber-físicos, IoT y computación en la nube en máquinas CNC. Así, con la máquina-herramienta 4.0, los sistemas de producción ciber-físicos en la industria se harán posible para alcanzar objetivos como la producción sobre-demanda y la personalización de productos en masa.

Resumiendo, la máquina-herramienta dentro del panorama de la Industria 4.0 pasa a ser un sistema ciber-físico cuyo núcleo es la máquina CNC en sí. Por lo tanto, cabe definir el concepto de sistema ciber-físico en este trabajo, lo cual es expuesto a continuación.

2.1.2. Sistemas ciber-físicos (CPS-Cyber-Physical Systems)

Los sistemas ciber-físicos hacen referencia a la integración de recursos computacionales, redes de comunicación y procesos físicos, proporcionando la abstracción, el modelamiento, el diseño y las técnicas de análisis para tratar tanto la información generada de los procesos como la dinámica física de su comportamiento [11]–[13]. Estos sistemas, que en esencia son formados por componentes de software y hardware interaliados, son capaces de sensor, actuar y comunicarse a través de las redes de comunicación [14]. En los últimos años, los CPSs se han beneficiado del alto crecimiento y disponibilidad de nuevos sensores, sistemas de adquisición de datos y redes de computadores para hacer más factible su implementación en el ambiente industrial propuesto por la Industria 4.0.

La combinación de CPSs con sistemas de producción, logística y servicios en las prácticas industriales actuales, transformará las fábricas de hoy en día en fábricas inteligentes de la Industria 4.0 (*Smart Factories*) con un significativo potencial económico [15]. En este contexto, las máquinas-herramientas juegan un rol de suma importancia al ser el corazón de los sistemas de manufactura en la producción ciber-física ya que se han de transformar en máquinas-herramientas ciber-físicas (CPMT-*Cyber-Physical Machine Tools*). Estas máquinas son la siguiente evolución de las máquinas-herramientas actuales, o sea, la aplicación de CPSs en la manufactura [2].

Así, la CPMT es constituida por la máquina-herramienta en sí, los procesos de manufactura, recursos computacionales y redes de comunicación, donde computaciones embebidas monitorean y controlan los procesos de manufactura, con lazos de retroalimentación en los cuales los procesos afectan las computaciones y viceversa [2].

El desarrollo de una CPMT lleva a la necesidad de un modelo de información comprensivo para representar tanto la estructura de la máquina, incluyendo todos los dispositivos disponibles para generar datos, como el status en tiempo real de cada uno de sus componentes críticos aprovechando las ventajas de los datos en tiempo real proporcionados por los sistemas de adquisición de datos. Además del modelo de información, es necesario definir los protocolos de comunicación que permiten transportar los datos recuperados desde la máquina al internet de manera eficiente, transparente, consistente y segura.

En este punto, como fue expuesto antes en la introducción de este documento, el problema que existe está relacionado a la inmensa variedad de protocolos de comunicación existentes los cuales varían dependiendo del fabricante de la máquina herramienta, lo cual dificulta en primera instancia la actualización de las actuales máquinas herramientas con nueva tecnología, y en segundo lugar entorpecen la correcta integración, gerenciamiento e intercambio de información entre diferentes dispositivos y equipos de manufactura dentro de una planta de fábrica.

Actualmente, dos nuevos estándares MTConnect y OPC-UA están luchando por resolver los problemas de falta de estandarización en las comunicaciones industriales. En este trabajo, se ha decidido implementar el estándar MTConnect con el fin de desarrollar CPMTs aplicadas a procesos de manufactura aditiva usando máquina de fuente abierta basadas en el proyecto RepRap.

2.1.3. Manufactura aditiva

Mientras tanto, Manufactura Aditiva (*AM-Additive Manufacturing*), también referida como Impresión 3D o Prototipado Rápido (*RP-Rapid Prototyping*), ha explotado en popularidad en los últimos años y es ahora considerada como una de las mega-tendencias tecnológicas de la Industria 4.0. La razón es debido a sus capacidades únicas de permitir la fabricación de objetos físicos tridimensionales con geometrías complejas que antes eran casi imposibles de lograr con procesos de manufactura subtractiva tradicionales (fresado, torneado...). Además, minimiza el desperdicio de material y no requiere de herramientas sofisticadas de alto costo [16]. De esta manera, la AM ofrece potenciales ventajas para el proceso de desarrollo de los nuevos productos incluyendo reducción del tiempo de lanzamiento al mercado, personalización en masa y optimización de la cadena de suministro, lo cual representa beneficios para las empresas en términos del tríptico costo,

tiempo y calidad.

Desde sus inicios en los años 1980s, los procesos de AM han experimentado un considerable progreso y expansión. Inicialmente fueron reservados sólo a la creación de modelos prototipo de productos; sin embargo, gracias a los avances que ha en materiales, velocidad, calidad y utilidades de máquina, hoy en día permiten la fabricación de partes totalmente funcionales y con alto valor agregado [17]. Más de 870 máquinas 1500 tipos de materiales de AM están ahora disponibles en el mercado [18]. Su expansión cubre un importante rango de aplicaciones que tales como la ingeniería aeronáutica y automotora, ingeniería biomédica, arquitectura, dispositivos electrónicos, etc. [19].

De acuerdo con la norma ISO/ASTM 52900 [20], los procesos de AM son clasificados en siete categorías dependiendo del tipo de material usado y de la técnica de manufactura. No obstante, la categoría que ha gozado de mayor popularidad debido a su simplicidad de uso y bajo costo, es el Modelamiento por Deposición Fundida (FDM-Fused Deposition Modeling) [21], tornándose atractiva sobre todo para pequeñas y medianas empresas, como para investigadores y aficionados. Una representación de este proceso se muestra en la Figura 4. El proceso de FDM consiste en empujar un filamento (usualmente de material termoplástico) a través de un mecanismo extrusor formado por un motor y un sistema de engranes. El material es extrudido por una boquilla caliente (hot-end) acoplada al extrusor con un orificio de diámetro menor que el filamento, y depositado capa-a-capa sobre una base plana hasta formar el objeto. En este campo, la mayor parte de los esfuerzos han sido enfocados principalmente al desarrollo de nuevos materiales [22], optimización de parámetros de proceso [23], sustentabilidad [24] y diseño de máquinas [25], [26]. Sin embargo, hasta el momento pocos estudios han sido direccionados al tema de monitoreamiento remoto de sistemas de AM a través de internet, principalmente debido a la falta de implementación estándares que definan tanto el modelo de transferencia de información como los protocolos de comunicación, lo cual puede ser abordado con el uso del estándar MTConnect.

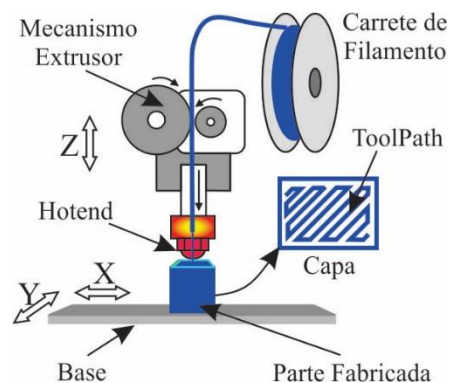


Figura 4. Representación del proceso FDM [27].

2.1.4. El protocolo MTConnect

MTConnect ha sido desarrollado por la Asociación para Tecnologías de Manufactura (AMT- *Association for Manufacturing Technology*) como un estándar de comunicación de fuente abierta, *royalty-free* y de sólo lectura que objetiva incrementar la interoperabilidad entre equipos-dispositivos de manufactura y aplicaciones de software a través de reportes de

datos vía Internet. El estándar define un modelo semántico para mejorar las capacidades de adquisición de datos de equipos-dispositivos de manufactura y aplicaciones de software basado sobre las tecnologías más prevalentes como XML (*eXtensible Markup Language*) y HTTP (*Hypertext Transfer Protocol*). De esta forma, proporcionar un lenguaje común de alto nivel entendible por cualquier dispositivo, equipo o sistema de manufactura es su principal objetivo.

Una arquitectura básica complaciente con MTConnect es ilustrada en la Figura 5, la cual consiste, esencialmente, de un componente Adaptador y un componente Agente de MTConnect. El Adaptador de MTConnect es básicamente un traductor que habilita a una entidad de manufactura para hablar MTConnect. Técnicamente, el Adaptador puede ser tanto un componente de software o un combo de software y hardware, que puede o no estar embebido en el sistema del equipo-dispositivo de manufactura. Su función consiste en coleccionar datos de máquina y de proceso generados en la planta de la fábrica, convertirlos a un simple diccionario basado en texto (SHDR-*Simple Hierarchical Data Representation*, como, por ejemplo: “*Timestamp|name|value|name|value*”), y entregarlo al el Agente MTConnect.

La esencia del MTConnect es el Agente, el cual es un programa responsable por recibir los datos desde el Adaptador, formatearlos en una estructura de datos en XML y disponibilizarlos para cualquier cliente web que pueda solicitarlos en una nube de manufactura. El Agente aloca los datos en un *buffer* de almacenamiento con capacidad suficiente para soportar hasta que una aplicación los requiera. Entonces, el agente permite que aplicaciones externas accedan a los datos respondiendo a requerimientos HTTP a través de un documento XML. Por ejemplo, a un requerimiento “*probe*” el agente responde con un conjunto de datos relacionados a los ítems de datos que pueden ser recuperados desde la máquina. Otros tipos de requerimientos tales como “*current*” y “*sample*” retornan los valores de cada ítem en un instante de tiempo y los datos dentro de un periodo de tiempo, respectivamente. Otros como “*asset*” y “*error*” también están definidos en la norma.

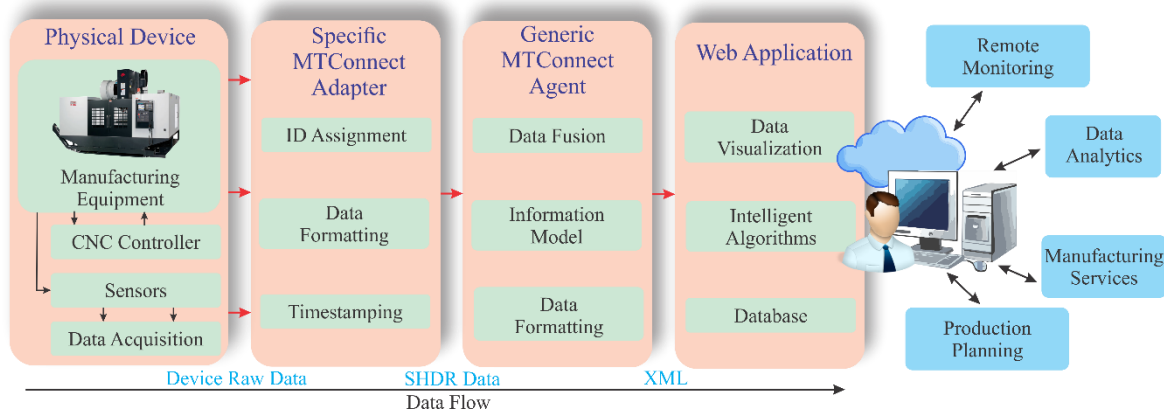


Figura 5. Arquitectura de un sistema de manufactura complaciente con MTConnect.

2.2. TRABAJOS RELACIONADOS

Desde su lanzamiento en 2008 [3], MTConnect ha sido adoptado cada vez más por los fabricantes en sus fábricas como una técnica efectiva para realizar un proceso estandarizado de recolección y monitoreo de datos con un ahorro significativo de tiempo y costos. Actualmente, cada vez más recursos de manufactura son compatibles con el estándar, incluyendo las máquinas de fresado y torneado CNC, robots industriales, etc. Hoy en día, el estándar MTConnect se considera un facilitador para desarrollar la próxima generación de máquinas-herramientas cibernéticas en la nueva era de la Industria 4.0 [2].

Muchos trabajos han llevado a cabo el desarrollo de sistemas de supervisión y monitoreamiento basados en la web utilizando MTConnect para la recopilación de datos y el análisis de máquinas de mecanizado CNC [16]–[18]. Las capacidades de MTConnect han demostrado ser de gran importancia, principalmente en el monitoreo de datos relacionados con el consumo de energía [19], [20] y el mantenimiento predictivo basado en condición de los sistemas de avanzados de manufactura [21].

El uso de MTConnect para monitoreamiento de máquina y análisis de datos de planeamiento de procesos basado sobre datos del modelo del producto de STEP-NC [22], [23] ha sido abordado por [24], [25]. STEP Tools, Inc. está avanzando en el desarrollo de un subproceso digital que conecta el diseño del producto, la fabricación y la inspección en torno a un Digital Twin utilizando STEP-NC, MTConnect y QIF (Quality Information Framework) [26]. En los laboratorios de ingeniería mecánica de la Universidad de Brasilia, el equipo del profesor Alvares ha desarrollado un framework de sistema cibernético para el monitoreamiento y la teleoperación de un centro de torneado CNC equipado con un controlador Fanuc que utiliza los protocolos MTConnect y OPC-UA, envolviendo datos de planeamiento de proceso de STEP-NC [27].

Con respecto a los procesos de manufactura aditiva, pocos trabajos se pueden encontrar en este trecho de investigación. Como parte del proyecto técnico entre NCDMM, AMT, MTConnect Institute y MAYA Design, se desarrolló una aplicación de software MTConnect Adaptador-Agente escrita en Python para que el fabricante de impresoras 3D comerciales Makerbot examine una posible implementación de MTConnect en sus máquinas [28]. Liu et al. [29] propusieron la implementación del protocolo MTConnect RESTful para monitorear impresoras 3D basadas de código abierto en una red cibernética de manufactura en la nube. Su solución es un combo Adaptador-Agente MTConnect implementado en una impresora 3D con un controlador de tecnología Raspberry. Una evolución de ese trabajo ha sido presentado en [30]. En otro trabajo, [31] propuso un método para construir Digital Twins de un tipo de impresora 3D utilizando un modelo de información basado en MTConnect. Estas soluciones se implementaron en máquinas con controlador de tecnología Raspberry. Recientemente, Stratasys también decidió incorporar en sus máquinas soluciones de sistemas compatibles con MTConnect para la Industria 4.0 [32].

A este punto, es claro que hay una falta de desarrollo con respecto a la manufactura aditiva con MTConnect, principalmente con máquinas de código abierto. Para abordar tal problema, este trabajo trata con la aplicación del protocolo MTConnect en máquinas de manufactura aditiva RepRap equipadas con controladores de tecnología Arduino, destinado a aplicaciones de producción de manufactura ciber-física.

3. METODOLOGÍA DE DISEÑO DEL FRAMEWORK

En esta sección se trata el abordaje metodológico adoptado para el diseño del framework. Lo que se propone en este trabajo es un prototipo de CPS que permita el monitoreamiento remoto de máquinas de AM vía internet. Esto envuelve el desarrollo, adecuación e integración de diferentes componentes tecnológicos tanto de software como de hardware que constituyen el framework. De acuerdo con Rodríguez et al. [25], el diseño y desarrollo de sistemas tecnológicos que envuelven la complejidad de diferentes dominios como mecánicos, electrónicos y computacionales, requiere el uso de una metodología para soportar y sistematizar la toma de decisiones que lleven a un producto de calidad. Una metodología que considere los requerimientos del usuario puede ser muy conveniente.

En este sentido, se ha decidido adoptar la metodología del Despliegue de la Función de Calidad (QFD-*Quality Function Deployment*), la cual es una metodología que permite definir atributos y/o características del producto siendo desarrollado con base en los requerimientos del usuario. El QFD fue propuesto por el profesor Akao [45] en Japón a finales de los 60s y desde entonces ha sido ampliamente aplicado al desarrollo de diversos tipos de productos incluyendo productos mecatrónicos [25], [46], [47] y de software [48]–[50]. Hoy en día, empresas como Toyota, Mitsubishi y Hewlett-Packard aplican QFD en el diseño y desarrollo de sus productos. Mayores detalles sobre esta metodología pueden ser encontrados en [45]. En este trabajo, la aplicación del QFD para el desarrollo del framework de MTConnect que se propone es descrita a través de los siguientes pasos:

3.1. Paso 1: Identificación de los usuarios.

Este primer paso de la aplicación de la metodología QFD trata con la identificación de los posibles usuarios del producto (el framework de MTConnect para AM). En este caso se ha optado por asumir una clasificación para los CPSs en dos categorías, las cuales son de bajo costo e industriales. El framework de MTConnect desarrollado en este trabajo como un prototipo de CPS encaja en la primera categoría, por lo que sus usuarios han sido identificados en primera instancia como los miembros del equipo de investigación del LaDPRER donde será implementado junto a otros trabajos. Eventualmente, pueden ser usuarios los investigadores externos trabajando en el área que pueden interesarse en usar el framework, ya que el material será disponibilizado libremente sobre la web. Entonces, para el desarrollo del trabajo se tuvo en cuenta la opinión de los seis colaboradores trabajando sobre este tópico en el LaDPRER incluyendo el propio autor.

3.2. Paso 2: definición de los requerimientos de usuario.

La definición de los requisitos de usuario (RUs) ha sido llevada a cabo con base en la experiencia de los colaboradores del LaDPRER y en la revisión de literatura relacionada. Este análisis se ha hecho considerando principalmente la idea de obtener un producto de alto rendimiento y de bajo costo. Un total de doce RUs fueron consolidados para el diseño del framework de MTConnect que debe indispensablemente permitir monitorear una impresora 3D RapRap vía internet. Los doce RUs son listados en la Tabla 1. En el lenguaje del QFD, los RUs pueden ser referidos como “”QUEs””.

Requerimientos de usuario	Descripción	Importancia Relativa
RU-1	Mostrar datos del proceso de impresión como temperaturas, posiciones, filamento, etc.	5
RU-2	Mostrar ítems de datos disponibles de la máquina	4
RU-3	Reportar status de la máquina en tiempo real	5
RU-4	Permitir conexión de máquinas RepRap a internet	5
RU-5	Buena estabilidad de la comunicación	4
RU-6	Mantener seguridad de los datos	5
RU-7	Tener un cliente web compatible	3
RU-8	Interfaz de usuario amigable	3
RU-9	Bajo costo de adecuación en impresoras 3D RepRap	4
RU-10	Que almacene los datos recolectados y los permita descargar a manera de reporte en formato CSV o XLSX	3
RU-11	Que esté disponible en la web para uso común (<i>Open Source</i>)	2
RU-12	Simple de implementar en laboratorio	3

Tabla 1..

3.3. Paso 3: Priorización de los RUs.

Este paso consiste en atribuir una importancia y peso relativo a los RUs en orden a priorizar aquellos que tienen un mayor efecto en la satisfacción del usuario final. Para ello fue realizado un análisis con el equipo de diseño, donde cada requerimiento fue comparado con sus pares para atribuirle un valor de importancia relativa entre 1 y 5, donde 1 significa poco importante y cinco significa de gran importancia o indispensable. Las importancias relativas de los RUs para el diseño del framework han sido colocadas respectivamente sobre la columna de la derecha en la Tabla 1.

Por otro lado, los pesos relativos son calculados como la suma ponderada de las importancias en cada uno de los RUs. A partir de eso, un análisis más elaborado que permite un mejor entendimiento de sobre los RUs que representan prioridad en el diseño es logrado a partir del diagrama de Pareto mostrado en la Figura 6. Este diagrama permite visualizar los RUs en orden de importancia de mayor a menor en sentido izquierda a derecha del gráfico.

De acuerdo con el principio de Pareto y con base en la Figura 6, para obtener una satisfacción mínima del 80% de los RUs establecidos para el diseño del framework, deben ser priorizados y cumplidos obligatoriamente los RUs con una importancia relativa mayor a 3. Analizando el grafico, solo el requerimiento RU-11 no representa una prioridad en el diseño del framework. Esto puede ser tomado como que para los integrantes del equipo diseñador que representan los usuarios del producto no es relevante disponibilizar el material desarrollado y basta con realizar publicación de artículos científicos reportando el trabajo realizado.

3.4. Paso 4: Selección de los parámetros de diseño.

Este paso consiste en desplegar los RUs en parámetros de diseño (PDs), los cuales representan conceptos más físicos del producto que pueden ser mejor entendidos por el

ingeniero o diseñador. En otras palabras es una transición de lo abstracto hacia lo concreto, representando la primera toma de decisión en la fase de diseño en busca de definir características del producto final. En este trabajo los PDs fueron seleccionados basados sobre la experiencia del equipo del autor y sobre revisión de literatura. Un total de once PDs pensados para satisfacer los RUs establecidos fueron seleccionados para el diseño del framework listados en la Tabla 2. En lenguaje del QFD, los PDs suelen ser llamados “COMOs”.

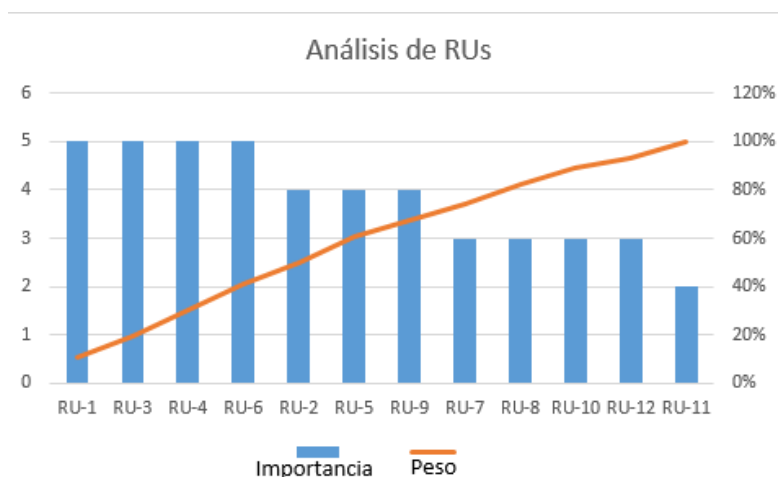


Figura 6. Diagrama de Pareto para el análisis de RUs.

Parámetros de diseño	Descripción
PD-1	Protocolo de información estándar para recolección y publicación de datos de impresión 3D vía internet.
PD-2	Modelo de información de la máquina
PD-3	Envío de información en tiempo real
PD-4	Adicionar un canal de comunicación basado en TCP/IP
PD-5	Protocolo de comunicación de solo lectura
PD-6	Aplicación web con capacidad de hacer requerimientos de información sobre la máquina y el proceso de impresión 3D
PD-7	Interfaz de gráfica de usuario
PD-8	Costo de desarrollo del framework
PD-9	Base de datos
PD-10	Material disponible en repositorio en la web
PD-11	Configuraciones accesibles y documentadas

Tabla 2. Parámetros de diseño seleccionados para el framework.

3.5. Paso 5: Relacionamiento entre URs y PDs.

El proceso de identificar las relaciones entre RUs y PDs es realizado de manera subjetiva por el diseñador y busca determinar la influencia de los PDs en los RUs. El grado de relacionamiento es fijado en tres niveles: relación fuerte (con valor de 5), relación moderada (con valor de 3) y relación débil (con valor de 1). El análisis de relacionamiento se ha hecho a través de la matriz QFD mostrada en la Figura 7. Uno de los resultados de este análisis

son las importancias técnicas de los PDs colocadas sobre la parte inferior de la matriz QFD. Se puede observar que los DPs marcados como DP-1, DP-2, DP-4 y DP-6 son los que tienen la mayor importancia técnica debido a que afectan en gran medida el cumplimiento de la mayoría de los RUs. Por lo tanto, esto quiere decir que debe ser enfocado mayor cuidado en estos parámetros de diseño durante el desarrollo.

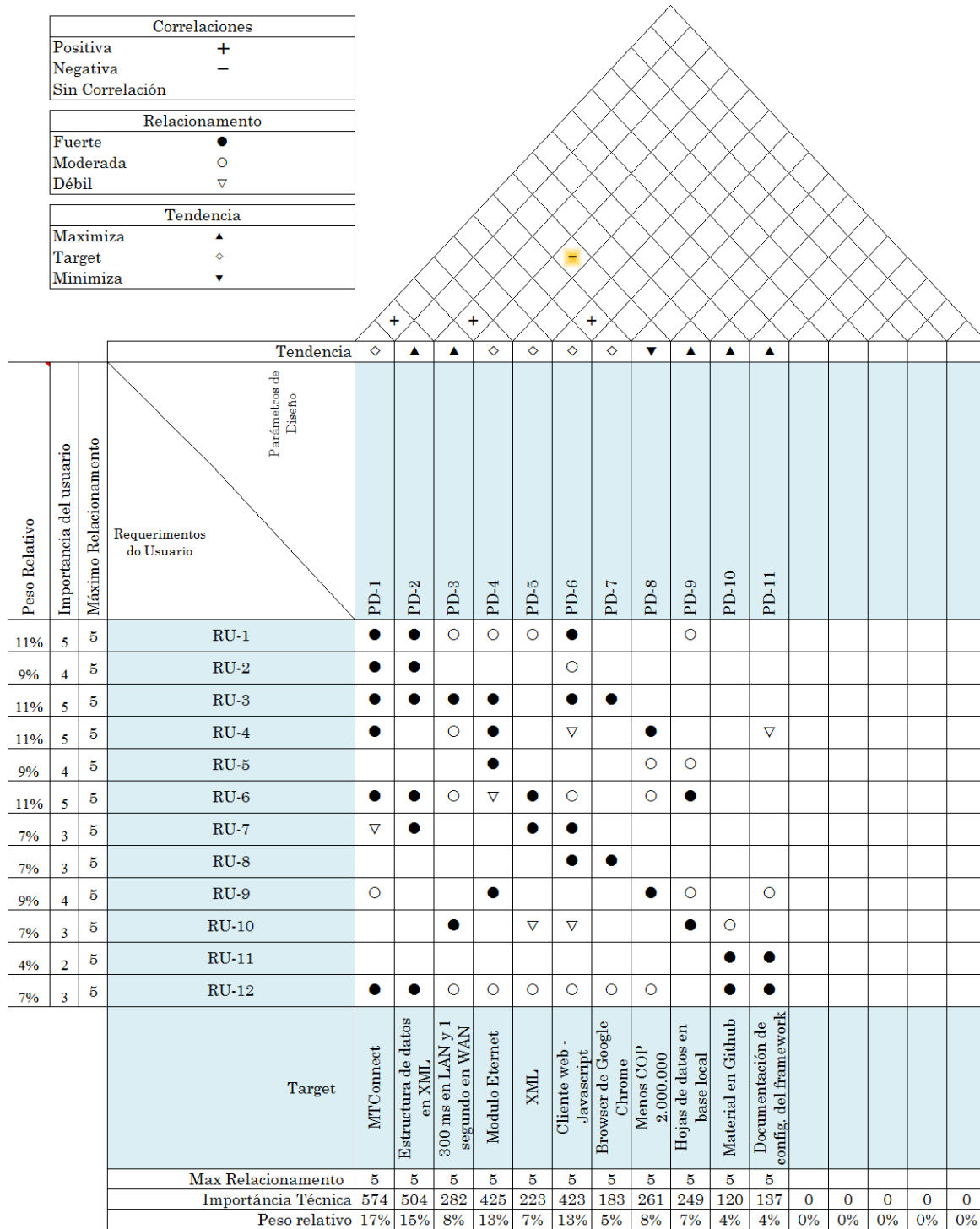


Figura 7. Matriz de casa de la calidad del QFD con relacionamientos entre RUs y PDs.

Otro de los resultados en el análisis de la matriz QFD es la obtención de las especificaciones-meta del producto final, las cuales fueron desplegadas a partir de los PDs seleccionados. Las especificaciones-meta del framework de MTConnect también están localizadas sobre la parte inferior de la matriz QFD de la Figura 7. Estas hacen referencia al uso del estándar MTConnect, usando una estructura de datos para representar la máquina en XML, con reportes de datos con un *delay* de 300 milisegundos en una conexión de red local y 1 segundo en conexión de red externa. Además, el uso de un módulo Ethernet para habilitar la conexión de las máquinas a internet, la seguridad de documentos de solo lectura en XML, un cliente web específico programado en Javascript para el monitoreamiento usando el browser de google Chrome. El costo del proyecto fue estimado a COP 2.000.000 considerando el caso de tener que construir la impresora 3D; sin embargo, en realidad en este caso que se contaba con las máquinas disponibles del LaDPRER, el costo de proyecto se reduce a tan solo COP 50.000 referentes al valor del módulo Ethernet acoplado al Arduino. Por el momento el framework almacena los datos en hojas de datos en base de datos local, pero puede ser extendido a una base de datos en la nube. Finalmente, la documentación se requiere disponible en un repositorio en la web para uso común.

Para mayor claridad las especificaciones-meta han sido listadas en la Tabla 3 junto a sus respectivos sentidos de tendencia. Tales especificaciones-meta representan las características que deben aparecer al final del desarrollo del framework.

Especificaciones-Meta	Tendencia
MTConnect	Target
Estructura de datos en XML	Maximiza
300 ms en LAN y 1 segundo en WAN	Maximiza
Modulo Ethernet	Target
XML	Target
Cliente web en Javascript	Target
Browser de Google Chrome	Target
Menos de COP 2.000.000	Minimiza
Hojas de datos en base local	Maximiza
Material en Github	Maximiza
Documentación de configuraciones del framework	Maximiza

Tabla 3. Especificaciones-meta y tendencias para las características del framework.

4. DESARROLLO DEL FRAMEWORK

4.1. ARQUITECTURA DEL PROTOCOLO MTCONNECT A DESARROLLAR

Los grandes retos de la cuarta revolución industrial es la adquisición de los datos en tiempo real para máquinas de diferentes fabricantes. El MTConnect fue creado para solucionar este problema permitiendo la comunicación con las máquinas de una forma estandarizada. Una implementación del protocolo MTConnect consta de un adaptador el cual es encargado de tomar los datos de la maquina y pasarlos al agente en un formato el cual este entienda, el agente se encarga de servir los datos a los clientes web que los soliciten en un formato XML (eXtensible Markup Language), este tipo de formato es muy utilizado actualmente para transferir información entre aplicaciones desarrolladas con diferentes lenguajes, la comunicación entre adaptador y agente se da mediante el protocolo TCP/IP (Transmission Control Protocol/Internet Protocol) empleando solicitudes HTTP(Hypertext Transfer Protocol) el cual es el principal protocolo utilizado por el internet para comunicar.

En la Figura 8 se indica con un gráfico cómo es el flujo de los datos utilizando el MTConnect, perdiendo de izquierda a derecha se encuentra la máquina con los datos para monitorear, estos datos pertenecen a los sensores y están formateados según el fabricante. El adaptador se toma estas formateándolos en un lenguaje que entienda el agente el cual puede ser formato de barras como se muestra en recuadro azul de la Figura 8. El agente toma los suministrados por el adaptador para responder a los clientes que soliciten dichos datos en lenguaje estándar XML por medio del protocolo HTTP. Los clientes interpretan esta información para mostrarlas al usuario ya sea en graficas o tablas dependiendo de la aplicación.

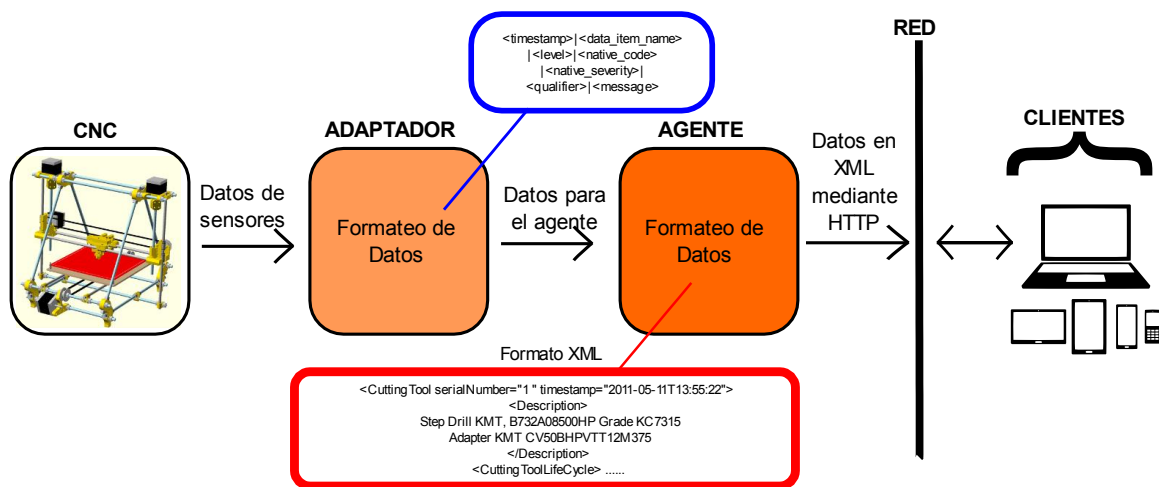


Figura 8. Estructura para comunicación de datos usando MTConnect.

El modelo de información definido según el estándar MTConnect para este trabajo se muestra en la Figura 9, esta ilustración muestra el modelo de información construido para máquina RepRapa utilizada en este trabajo. La jerarquía del modelo siempre comienza con la estructura del elemento etiquetado como "\ MTConnectDevices" que contiene las

subelementos estructurales "\ Devices" y "\ Header" (Figura 9). "\ Devices" abarca todas las máquinas agregadas al sistema, donde cada máquina está representada por el elemento estructural "\ Device" dentro de la jerarquía del modelo. Un "\Device" está compuesto por "\ Description", "\ DataItems " y "\Components". "\ Descripción" proporciona información sobre la máquina, mientras que "\ DataItems "contienen un "\DataItem" relacionando la disponibilidad de la máquina. En la misma jerarquía de nivel, "\Components" incluye los datos de la máquina referente a las posiciones del eje (eje x, eje y, eje z y extrusora).

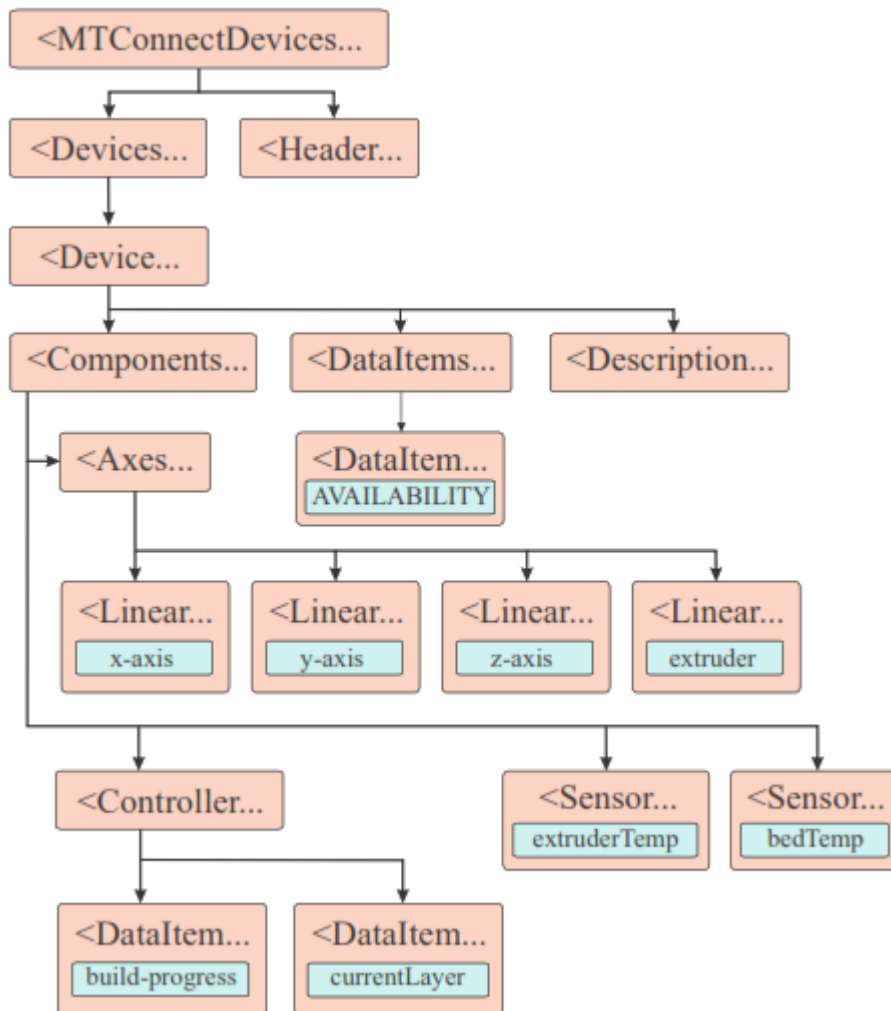


Figura 9. Modelo de información para el agente MTConnect.

4.2. AGENTE MTCONNECT EMPLEADO

4.2.1. Agente MTConnect C++

Un agente MTConnect es el encargado de recibir los datos del adaptador y formatearlos en un documento XML para ser entregados a los clientes web por medio del protocolo HTTP. Para este trabajo se utilizó el MTConnect C++ Agent en la versión 1.4.0 el cual funciona

como un servicio en y provee un desarrollo completo del servidor HTTP que solicita el estándar MTConnect, instalado el agente es necesario solamente especificar la descripción XML del dispositivo y la dirección IP del dispositivo donde solicitara los datos.

4.2.2. Instalación y configuración del agente

Los archivos de instalación del agente están disponibles en el repositorio de Github en la siguiente dirección: <https://github.com/mtconnect/cppagent>, la instalación se llevó a cabo en el sistema operativo Windows 7 x64 bits.

Al extraer los archivos del comprimido .zip en ruta cppagent-master\install\Windows\src se encuentra el ejecutable MTConnectAgentInstall1.2.exe, al aceptar los términos de licencia y dar clic en siguiente se puede escoger la ruta de instalación preferiblemente en el disco C:/ del sistema operativo Windows 7 como se muestra en la Figura 9.

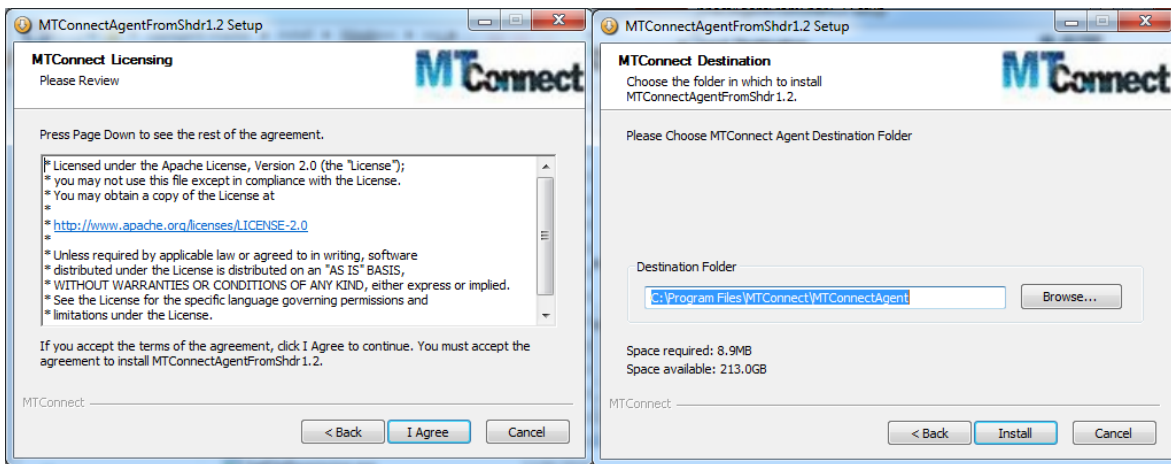


Figura 10. Instalación del agente.

Luego de que termine la instalación se configuro el nombre del servicio y el puerto por el cual se comunicaría el agente vía HTTP, no es necesario cambiar la configuración por defecto funciono bien al instalar como se muestra en la Figura 10.

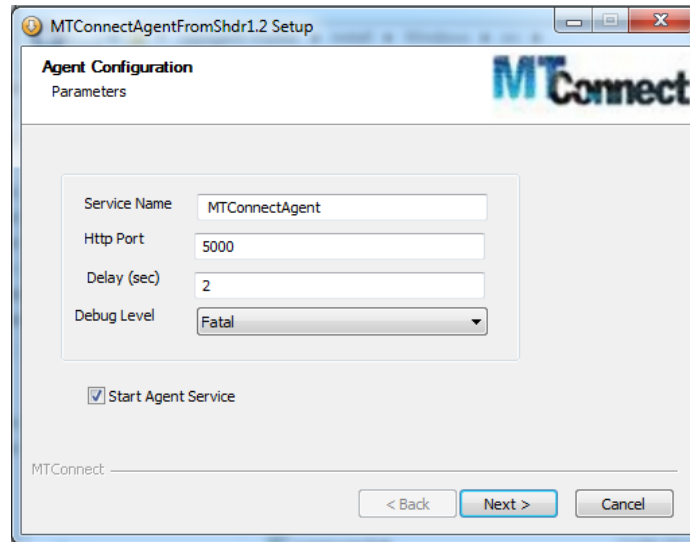


Figura 11. Configuración del agente.

Al terminar la instalación del agente se abre la ventana para configurar el adaptador entre el agente y la máquina, aquí se seleccionó el vendedor del controlador, el tipo de maquina y la versión como se muestra en la Figura 11.

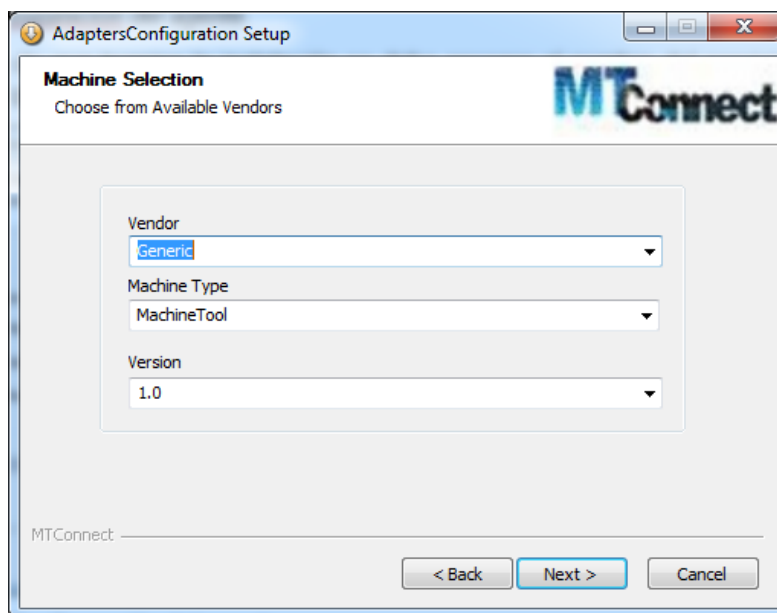


Figura 12. Configuración del agente.

En la siguiente ventana se seleccionó la descripción del modelo XML para el dispositivo, la IP y puerto de comunicación que utilizará el adaptador para comunicarse con la maquina como se muestra en la Figura 12.

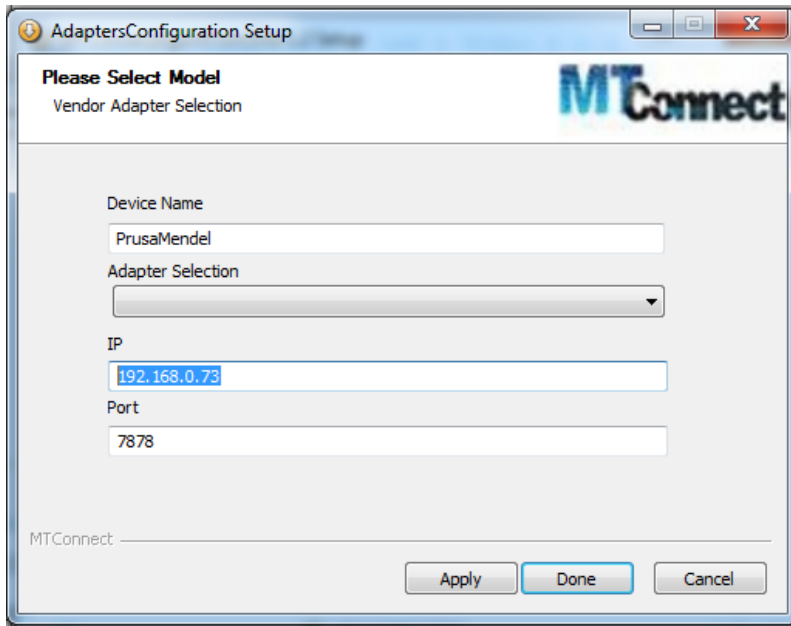


Figura 13. Selección del modelo para el adaptador XML.

En el administrador de servicios de Windows se crea un servicio con el nombre dado durante la instalación como se muestra en la Figura 13.

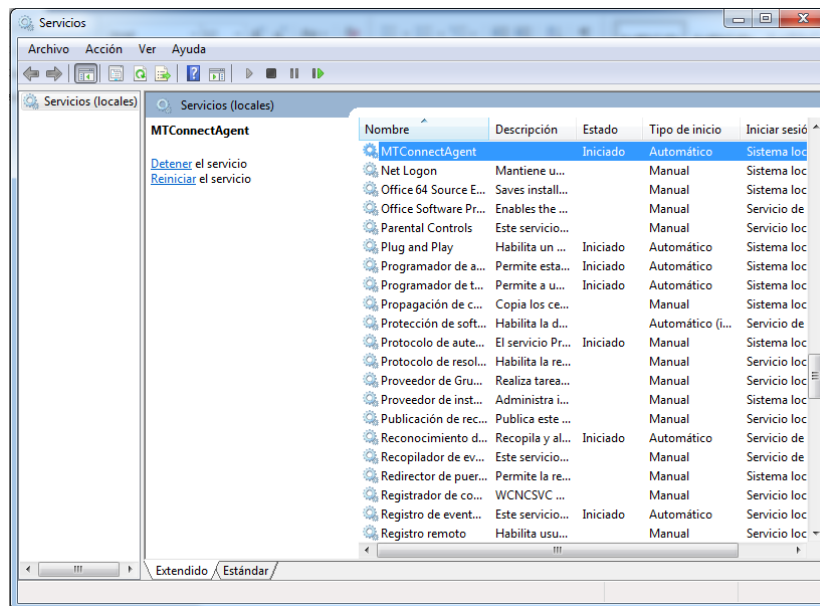


Figura 14. Administrador de servicios de Windows.

En la ruta de instalación del agente se crea un archivo llamado Agent.cfg, este archivo contiene los parámetros de configuración para el agente, en la Figura 14 se muestra el archivo luego de abrirlo con un editor de texto.

```

ServiceName = MTConnectAgent
Devices = PrusaMendel.xml
Port = 5000
CheckpointFrequency=5000
AllowPut=true
Adapters
{
    PrusaMendel
    {
        Host=192.168.10.73
        Port=7878
        DeviceType=Faunc - CNC4axis
        FilterDuplicates=yes
        ReconnectInterval=5
        RealTime=true
    }
}
# Logger Configuration
logger_config
{
    logging_level = debug
    output = cout
}

```

Figura 15. Archivo Agent.cfg editado.

4.2.3. Parámetros de configuración para el agente

Parámetros de configuración de nivel superior:

- ServiceName- Cambia el nombre del servicio al instalar o quitar el servicio. Esto permite que múltiples agentes se ejecuten como servicios en la misma máquina.
Predeterminado: MTConnect Agent
- Devices- El archivo XML para cargar que especifica los dispositivos y se suministra como resultado de una solicitud de sondeo. Si no se encuentra la clave, se intentan los valores predeterminados.
Valores predeterminados: probe.xml o Devices.xml
- Port - El número de puerto al que se enlaza el agente para las solicitudes.
Predeterminado: 5000
- CheckpointFrequency- Los puntos de control de frecuencia se crean en el flujo. Esto se usa para corriente con el argumento. Este es un elemento de configuración avanzada y no debe cambiarse a menos que comprenda el funcionamiento interno del agente.
Predeterminado: 1000
- AllowPut - Permitir HTTP, PUT o POST de valores de elementos de datos o activos.
Predeterminado: falso

Elementos de configuración del adaptador:

- **Adapters-** Comienza una lista de bloques de dispositivos. Si no se especifican los adaptadores y el archivo de dispositivos solo contiene un dispositivo, se creará una entrada de dispositivo predeterminada con un adaptador ubicado en el host local y el puerto 7878 asociado con el dispositivo en el archivo de dispositivos.
Predeterminado: localhost 5000 asociado con el dispositivo predeterminado
- **Host -** El host en el que se encuentra el adaptador.
Predeterminado: localhost
- **Port -** El puerto para conectar al adaptador.
Predeterminado: 7878
- **FilterDuplicates-** Si el valor es yes, filtra todos los valores duplicados para los elementos de datos. Esto es para admitir adaptadores que no están realizando un filtrado duplicado adecuado.
Predeterminado: no
- **ReconnectInterval-** La cantidad de tiempo entre los intentos de reconexión del adaptador. Esto es útil para la implementación de adaptadores de alto rendimiento donde la disponibilidad necesita ser rastreada casi en tiempo real. El tiempo se especifica en milisegundos (ms).
Predeterminado: 10000
- **RealTime -** Aumente la prioridad de subprocesos de este adaptador para que los eventos se manejen más rápido.
Predeterminado: falso

Elementos de configuración logger_config:

En la carpeta de instalación del agente se crea un archivo llamado Agent.log el cual lleva un registro de los eventos ejecutados por el servicio del agente.

- **logger_config -** La sección de configuración de registro.
- **logging_level-** El nivel de registro: trace, debug, info, warn, error, o fatal.
Predeterminado: info
- **output-** El archivo de salida o flujo. Si utiliza un archivo, especifique como: "file <filename>". cout y cerr se pueden usar para especificar la salida estándar y los flujos de error estándar. Por defectos al mismo directorio que el ejecutable.
Predeterminado: archivoadapter.log

Para más parámetros de configuración del agente, adaptador y logger_config ir al enlace de Github: <https://github.com/mtconnect/cppagent>

4.2.4. Protocolo de comunicación entre el agente y el adaptador

El agente y el adaptador tienen un latido que se asegura de que cada uno responda para manejar adecuadamente las desconexiones de manera oportuna. La frecuencia de Heartbeat es establecida por el adaptador y respetada por el agente. Cuando el agente se conecta al adaptador, primero envía * PING y luego espera la respuesta * PONG <timeout>, donde <timeout> se especifica en milisegundos. Así que si se dan las siguientes comunicaciones:

Agente: * PING

Adaptador: *PONG 10000

Si el adaptador no responde PONG, el agente asume que el adaptador no puede participar en el protocolo de latido y usa el tiempo heredado especificado anteriormente.

4.3. DESARROLLO DEL ADAPTADOR MTCONNECT

Un adaptador MTConnect puede estar formado por software y hardware, permitiendo lograr comunicar los datos de la maquina con el agente y que esté los entregue en un lenguaje entendible a los clientes web. Para este trabajo se desarrolló un adaptador tanto de software como hardware, la parte del software esta embebida en el firmware SPRINTER que controla la impresora 3D y el hardware es el módulo ENC28J60 el cual permite una comunicación física vía ethernet con la computadora donde se encuentra el agente instalado esperando los datos.

4.3.1. Hardware del adaptador

La comunicación utilizada entre el agente y el adaptador se da por medio del protocolo TCP/IP y solicitudes HTTP, es por esto que para la parte física del adaptador se utilizó el módulo de Arduino ENC28J60, basado en su hoja de datos el ENC28J60 adopta el estándar IEEE 802.3 compatible para usarse con ethernet y una interfaz SPI para la comunicación con el controlador de la máquina. Se empleo la librería para Arduino EtherCad la cual provee una serie de funciones que permiten leer los paquetes de datos que llegan del agente por el módulo y enviar los datos de los sensores formateados por el adaptador hacia el agente a través del manejo del protocolo HTTP. En la Figura 15 se muestra la conexión del módulo ethernet al controlador Arduino de la maquina por medio de shield ramps 1.4.

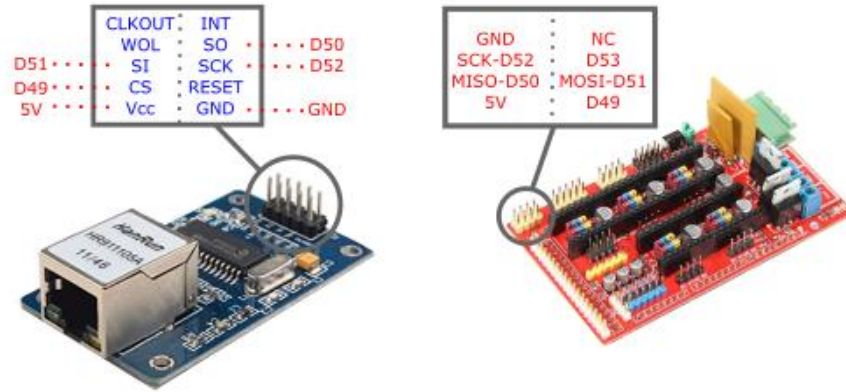


Figura 16. Conexión ENC28J60 y ramps 1.4

4.3.2. Software del adaptador.

El protocolo de comunicación entre el agente y el adaptador funciona con un Heartbeat o latido que mantiene viva la comunicación se cerciora de cada uno responda para manejar las desconexiones entre ambos, la frecuencia de estas respuestas la da el adaptado. El parte con el software del adaptador esta embebida dentro del firmware Sprinter que es el código que controla la máquina, se usó el timer 5 del Arduino mega para manejar las respuestas al Heartbeat y enviar datos al agente. En la Figura 16 se muestra un esquema de la comunicación entre adaptador y agente para este trabajo.

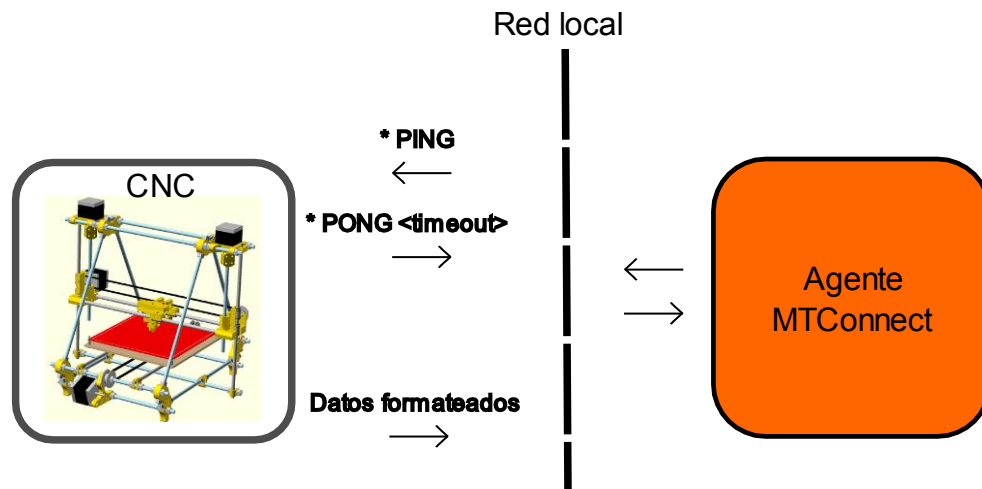


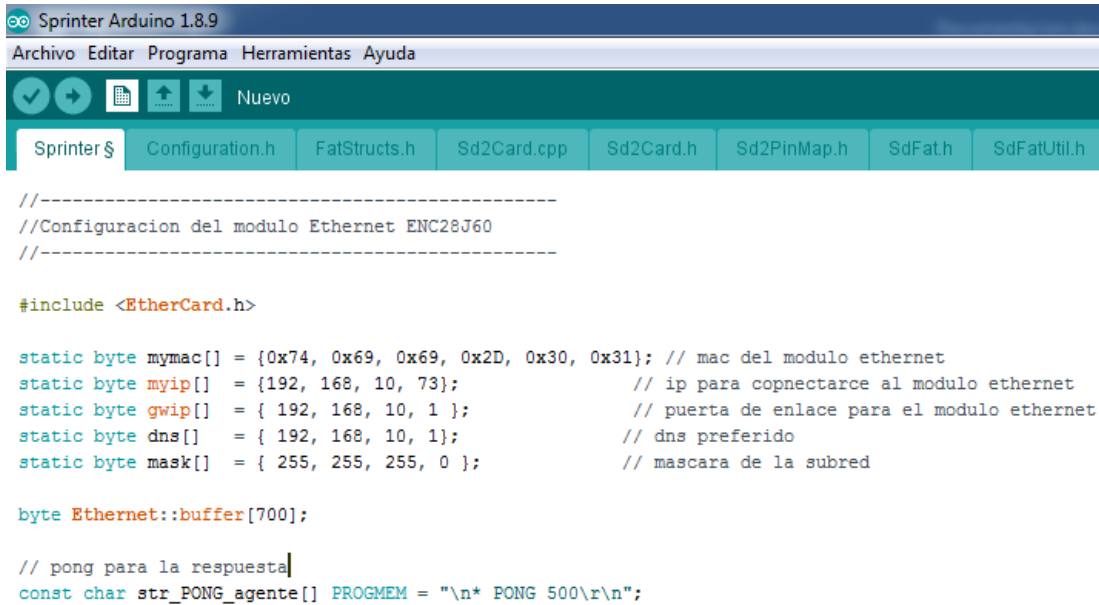
Figura 17. Arquitectura de comunicación entre adaptador y agente.

4.3.3. Configuración del ENC28J60

La configuración del módulo ethernet se llevó a cabo utilizando la librería EtherCad la cual mediante la función Begin carga los parámetros para que este pueda conectarse a la red ethernet donde se encuentra el agente.

En la Figura 17 se muestra los parámetros de configuración para modulo ethernet la variable mymac es la configuración física que tomara el módulo dentro de la red que se conectó, myip es la dirección IP que el módulo utiliza para conectarse a la red de ethernet, los otros

valores son la puerta de enlace, el DNS (Domain Name System), máscara de la sub red, el tamaño del buffer para almacenar los datos que se envían y reciben a través del módulo y un arreglo de caracteres que es usado para responder el Heartbeat del agente.



```

//-----
//Configuracion del modulo Ethernet ENC28J60
//-----

#include <EtherCard.h>

static byte mymac[] = {0x74, 0x69, 0x69, 0x2D, 0x30, 0x31}; // mac del modulo ethernet
static byte myip[] = {192, 168, 10, 73}; // ip para conectarse al modulo ethernet
static byte gwip[] = { 192, 168, 10, 1 }; // puerta de enlace para el modulo ethernet
static byte dns[] = { 192, 168, 10, 1}; // dns preferido
static byte mask[] = { 255, 255, 255, 0 }; // mascara de la subred

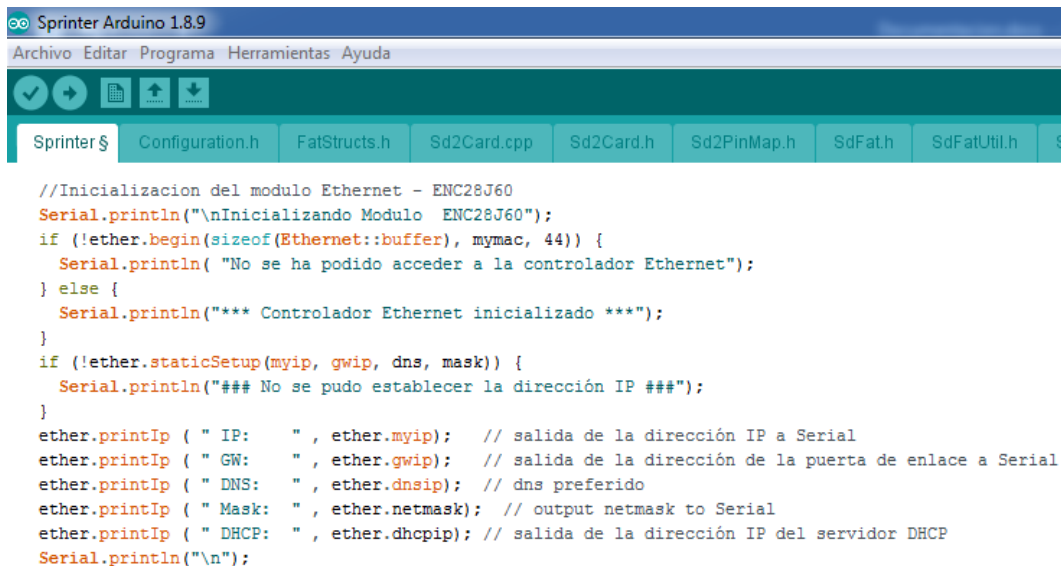
byte Ethernet::buffer[700];

// pong para la respuesta
const char str_PONG_agente[] PROGMEM = "\n* PONG 500\r\n";

```

Figura 18. Parámetros de configuración para el módulo ENC28J0.

La Figura 18 muestra el script que inicia el módulo, el inicio se hace llamando a la función Begin a la cual se le pasan como parámetros el buffer para almacenar los datos, la dirección MAC (Media Access Control) y el pin CS para controlar la comunicación ISP (Serial Peripheral Interface), la configuración de IP estática se hace utilizando la función staticSetup pasando como parámetros los mostrados en la Figura 15.



```

//Inicializacion del modulo Ethernet - ENC28J60
Serial.println("\nIniciando Modulo ENC28J60");
if (!ether.begin(sizeof(Ethernet::buffer), mymac, 44)) {
  Serial.println("No se ha podido acceder a la controlador Ethernet");
} else {
  Serial.println("*** Controlador Ethernet inicializado ***");
}
if (!ether.staticSetup(myip, gwip, dns, mask)) {
  Serial.println("### No se pudo establecer la dirección IP ###");
}
ether.printIp ( " IP:   ", ether.mypip); // salida de la dirección IP a Serial
ether.printIp ( " GW:   ", ether.gwip); // salida de la dirección de la puerta de enlace a Serial
ether.printIp ( " DNS:  ", ether.dnsip); // dns preferido
ether.printIp ( " Mask: ", ether.netmask); // output netmask to Serial
ether.printIp ( " DHCP: ", ether.dhcpip); // salida de la dirección IP del servidor DHCP
Serial.println("\n");

```

Figura 19. Inicio del módulo Enc28j60.

La lista de métodos y parámetros usados para la configuración del módulo ethernet están descritos en la Tabla 4.

Método	Descripción
<code>#include <EtherCard.h></code>	Adicionar los archivos de la librería EtherCad.h al código del Sprinter.
<code>static byte mymac[] = { , , , , };</code>	Arreglo de tipo estático con los bytes de la mac que se le asignara al módulo Enc28j60.
<code>static byte myip[] = { , , };</code>	Arreglo de tipo estático con los bytes de la ip que tomara el módulo Enc28j60 en caso de estar disponible, esta debe estar configurada en el rango de la red que se conecte la máquina.
<code>static byte gwip[] = { , , };</code>	Arreglo de tipo estático con los bytes de la puerta de enlace que tomara el módulo Enc28j60, esta debe ser la misma de la red en donde se conecte la máquina.
<code>static byte dns[] = { , , };</code>	Arreglo de tipo estático con los bytes del dns preferido para módulo Enc28j60. (Se puede omitir esta configuración)
<code>static byte mask[] = { , , , };</code>	Arreglo de tipo estático con los byte de la máscara de subred tomara el módulo Enc28j60, esta debe estar según el rango de la red en donde se conecte la máquina.
<code>byte Ethernet::buffer[];</code>	Buffer de datos para almacenar los datos que se enviaran y recibirán a través del modulo Enc28j60.
<code>const char str_PONG_agente[] PROGMEM = "\n* PONG 500\r\n";</code>	Mensaje de respuesta para el protocolo que mantiene viva la comunicación entre el agente y el adaptador. (Heartbeat).
<code>ether.begin(size of buffer,mac, CS);</code>	Inicio de la interfaz de red: tamaño del buffer de datos, mac para modulo enc28j60 y pin CS de la placa donde se conecta.
<code>ether.staticSetup(myip, gwip, dns, mask);</code>	Configuración de la interfaz de red con ip estatica: ip, puerta de enlace, dns, mascara de subred.

Tabla 4. Métodos y parámetros para la configuración del ENC28J60.

4.3.4. Comunicación entre el adaptador y el agente.

La respuesta PONG y los datos de los sensores de la maquina son enviados al agente de manera constante utilizando una interrupción por tiempo, la interrupción de tiempo empleada fue el timer 5 del controlador de la maquina Arduino Mega. En la Figura 19 se muestra la configuración del timer 5 a una frecuencia de 80.23 Hz.

```

Sprinter Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Sprinter $
TCCR5A = 0;
TCCR5B = 0;
OCR5A = 781;
TCCR5B |= (1 << WGM52);
TCCR5B |= (1 << CS10);
TCCR5B |= (1 << CS12);
TIMSK5 |= (1 << OCIE5A);

```

Figura 20. Configuración del timer 5.

El timer 5 tiene asociada una función de rutina para la interrupción llamada ISR, esta función lo que hace es verificar si hay algún paquete de datos por leer en el buffer del módulo ethernet, en el caso que si este responde con PONG para mantener viva la comunicación entre el agente y el adaptador y luego envía los datos de los sensores al agente. En la Figura 20 se muestra la ISR para el timer 5.

```

Sprinter Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Subir
Sprinter $
//-----
//funcion para el timer5
//-----
ISR(TIMERS5_COMPA_vect) //esta es la funcion de la rutina de la interrupcion. Lo que está
{
  word pos = ether.packetLoop(ether.packetReceive());
  // check if valid tcp data is received
  if (pos) {
    char* data = (char *) Ethernet::buffer + (pos + 2);
    if (strncmp("PING", data, 4) == 0) {
      ether.httpServerReplyAck(); // enviar ack a la solicitud
      memcpy_P(ether.tcpOffset(), str_PONG_agente, sizeof str_PONG_agente); // enviar el
      ether.httpServerReply_with_flags(sizeof str_PONG_agente - 1, TCP_FLAGS_PSHACK_V);
      delay(5);
      ether.httpServerReply_with_flags(EnviarDatos(msgDatos()), TCP_FLAGS_PSHACK_V);
      delay(5);
    }
  }
}

```

Figura 21. Función ISR del timer 5.

La función msgDatos() devuelve una cadena con los datos de los sensores de la maquina en el formato de barras aceptada por el agente y la función enviarDatos(String str_MTC_datos) recibe como parámetro un cadena para pasarla al buffer de datos devolviendo la posición del cursores en el buffer. En la Figura 21 se muestra el código de estas dos funciones.


```

Sprinter Arduino 1.8.9
Archivo Editar Programa Herramientas Ayuda
Subir
Sprinter $
String msgDatos() {
  const String str_MTC_datos = "Xact|" + String(current_position[X_AXIS], 1) +
    "|Yact|" + String(current_position[Y_AXIS], 1) +
    "|Zact|" + String(current_position[Z_AXIS], 1) +
    "|Extruder|" + String(current_position[E_AXIS], 1) +
    "|extruder temp sensor|" + String(analog2temp(current_raw)) +
    "|bed temp sensor|" + String(analog2tempBed(current_bed_raw)) +
    "\r\n";

  return str_MTC_datos;
}

static word enviarDatos(String str_MTC_datos) {
  BufferFiller bfill = ether.tcpOffset();
  const char strbuf[str_MTC_datos.length()];
  str_MTC_datos.toCharArray(strbuf, str_MTC_datos.length() + 1);
  bfill.emit_raw(strbuf, strlen(strbuf));

  return bfill.position();
}

```

Figura 22. Funciones para obtener datos de los sensores y cargar mensajes en el buffer del módulo ethernet.

La descripción de los métodos usados para la comunicación entre el agente y el adaptador están descritas en la Tabla 5.

Método	Descripción
<code>ether.packetLoop(ether.packetReceive());</code>	Analizar los paquetes de datos recibidos, en caso de tener paquetes recibidos se devolverá el desplazamiento de la carga útil TCP en el buffer de datos o cero si no hay datos.
<code>char* data = (char *) Ethernet::buffer + pos;</code>	Lectura de los datos TCP en el buffer de datos hasta la posición válida.
<code>memcpy_P(ether.tcpOffset(), str_PONG_agente, sizeof str_PONG_agente);</code>	Añade texto con formato en la memoria flash del controlador
<code>ether.httpServerReply_with_flags(sizeof str_PONG_agente - 1, TCP_FLAGS_PSHACK_V);</code>	Envía un paquete de datos a través del módulo ethernet con banderas TCP determinadas.
<code>ether.tcpOffset()</code>	Ubica el puntero al inicio de la carga TCPk

Tabla 5. Funciones para la comunicación con el agente mediante el módulo del ENC28J60.

El firmware Sprinter modificado está disponible para ser descargado en el repositorio de GitHub con el siguiente enlace: <https://github.com/juangasca95/Adaptador-MTcoonec---Prusa-Mendel-i2>

4.4. CLIENTE WEB PARA MTCONNECT.

Un cliente web es una aplicación que se conecta al agente por medio de peticiones HTTP y recibe información del estado de las máquinas en un documento con formato XML. Para este trabajo se realizó un cliente web desarrollado con html5 para interfaz del usuario y JavaScript para hacer las solicitudes HTTP al agente, leer el documento XML enviado como respuesta con los datos de la máquina, graficar y exportar una tabla con la información del estado de la máquina. En la Figura 22 se muestra una imagen completa de la interfaz que se desarrollada.

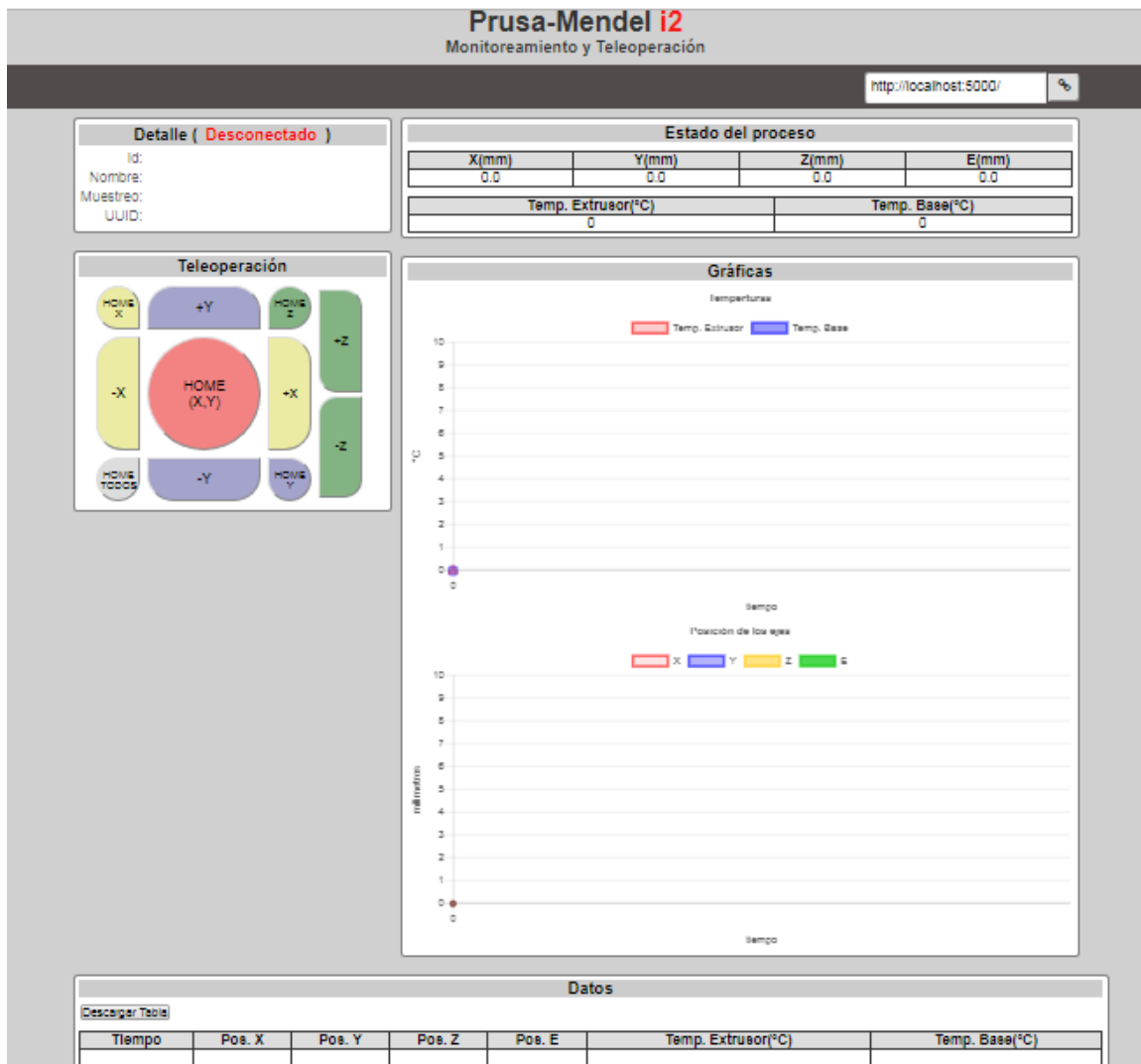


Figura 23. Interfaz de usuario para el cliente web.

4.4.1. Descripción de la interfaz

En la barra de navegación está ubicado un campo de texto para ingresar la dirección URL de donde se encuentra el agente con la información de la máquina, el botón que le sigue

en la parte derecha al del campo de texto inicia la conexión con el agente, si la conexión es exitosa se muestra una etiqueta en la parte izquierda de la maquina donde se lleva el tiempo de conexión entre el cliente y el agente, en caso que no tenga éxito la conexión se muestra una alerta indicando el error de conexión. En la Figura 23 se muestra la barra de navegación de la maquina tras una conexión exitosa.

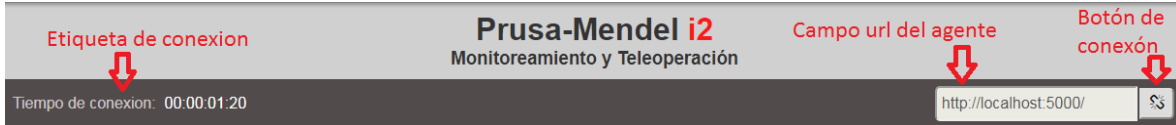


Figura 24. Barra de navegación para la interfaz del cliente web.

El cuerpo de la interfaz esta dividido en cinco partes: detalle, estado del proceso, tele operación, gráficas y datos.

- **Detalle:** en esta división se muestra el estado de conexión de la maquina y los atributos Id, nombre, muestreo y UUID obtenidos del documento XML enviados como respuesta por el agente. En la Figura 24 se muestra la división detalle al conectarse con un agente correctamente.



Figura 25. División detalle en la interfaz del cliente web.

- **Estado del proceso:** en esta división se muestra los datos que llegan del agente tras cada solicitud HTTP a este. Los datos que se muestran aquí son posición de los ejes XYZ, del extrusor E, temperatura del extrusor y de la base. En la Figura 25 se muestra la división estado del proceso al conectarse con un agente correctamente.

Estado del proceso			
X(mm)	Y(mm)	Z(mm)	E(mm)
189.0	150.0	80.0	53.0
Temp. Extrusor(°C)		Temp. Base(°C)	
20		21	

Figura 26. División estado del proceso en la interfaz cliente web.

- **Tele operación:** en esta división están los botones que permitirán la tele operación de la máquina de forma remota, esta parte se encuentra en construcción. En la Figura 26 se muestra la división tele operación.

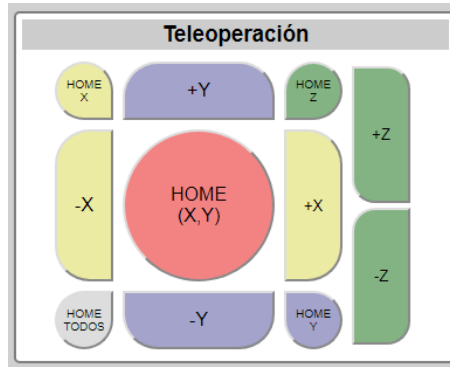


Figura 27. División tele operación en el cliente web.

- Graficas:** en esta división se muestran dos gráficas, la gráfica superior es para los valores de temperatura en el extrusor y la base, la gráfica inferior es para los valores de las posiciones de los ejes XYZ y el extrusor E. en la Figura 27 se muestra la división graficas.

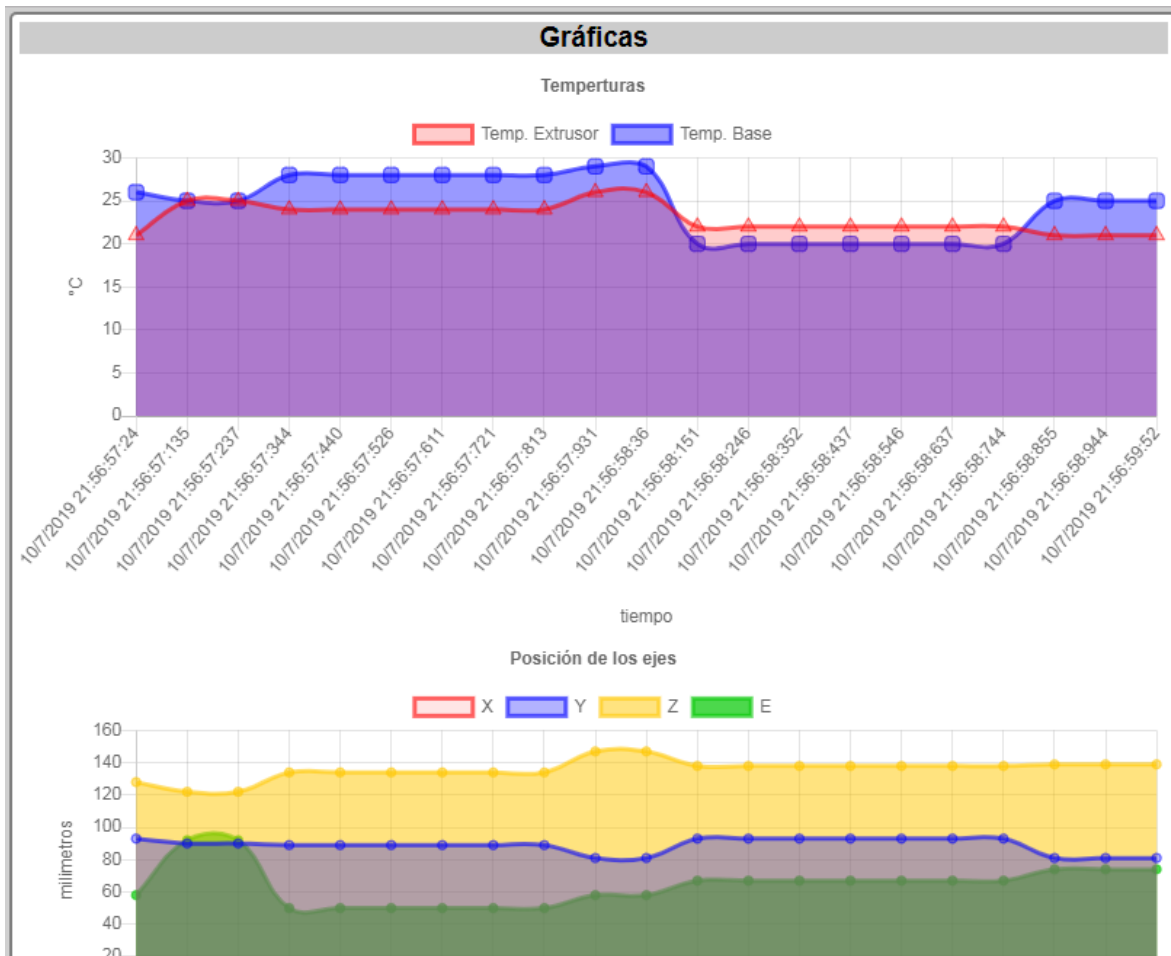


Figura 28. División graficas en el cliente web.

- **Datos:** en esta división se muestran los datos en una tabla leídos por el cliente del documento XML suministrados por el agente y ordenados por tiempo de llegada, en la esquina superior izquierda de la división se encuentra un botón que permite exportar los datos a un documento Excel con extensión .xls. en la Figura 25 se muestra la división de datos para la interfaz del cliente web.

Datos						
Descargar Tabla						
Tiempo	Pos. X	Pos. Y	Pos. Z	Pos. E	Temp. Extrusor(°C)	Temp. Base(°C)
19/6/2019 11:50:59.626	135	119	143	88	25	22
19/6/2019 11:50:58.624	134	137	135	89	20	25
19/6/2019 11:50:57.629	121	127	100	95	24	21
19/6/2019 11:50:56.624	149	111	118	86	20	25
19/6/2019 11:50:55.627	118	130	122	97	22	23
19/6/2019 11:50:54.627	125	112	105	86	20	26
19/6/2019 11:50:53.626	116	110	125	94	27	21
19/6/2019 11:50:52.626	141	128	119	94	26	25
19/6/2019 11:50:51.628	117	125	104	92	21	25

Figura 29. División datos en el cliente web.

4.4.2. Diagrama de flujo para el cliente web.

Un diagrama de flujo se usa para visualizar las actividades que ocurren dentro de un proceso, sistema o algoritmo informático. Se utilizó un diagrama de flujo para representar el proceso interno que ejecuta el cliente web desarrollado para este trabajo desde el momento de dar clic en el botón de conectar hasta finalizar la conexión con la URL de donde se originan los datos. En la Figura 29 muestra el diagrama de flujo para la programación del cliente web.

Al momento de cargar la página HTML con el cliente web este ejecuta la función `cargarVariables()` y la función `cargarGrafica()`.

- **cargarVariables():** Esta función carga todas las etiquetas de la página HTML del cliente en variables globales las cuales serán usadas para mostrar los datos y mensajes por las funciones que más adelante se ejecuten en la interfaz gráfica.
- **cargarGraficas():** Crea las gráficas a partir de las etiquetas canvas que están en la página HTML del cliente.

Cuando el usuario para el cliente web introduce la URL en el cual se encuentra el agente escuchando las peticiones HTTP para responder con un documento XML se ejecuta la función `conectar()`.

- **conectar():** Esta función es la encargada de iniciar el hilo que solicita los datos al agente y a su vez de finalizarlo dependiendo del valor de la etiqueta span que funciona como tooltip text para el botón conectar, si tiene un valor igual a conectar inicia la solicitud periódica de datos al agente, si tiene un valor igual a desconectar finaliza la solicitud de datos al agente.

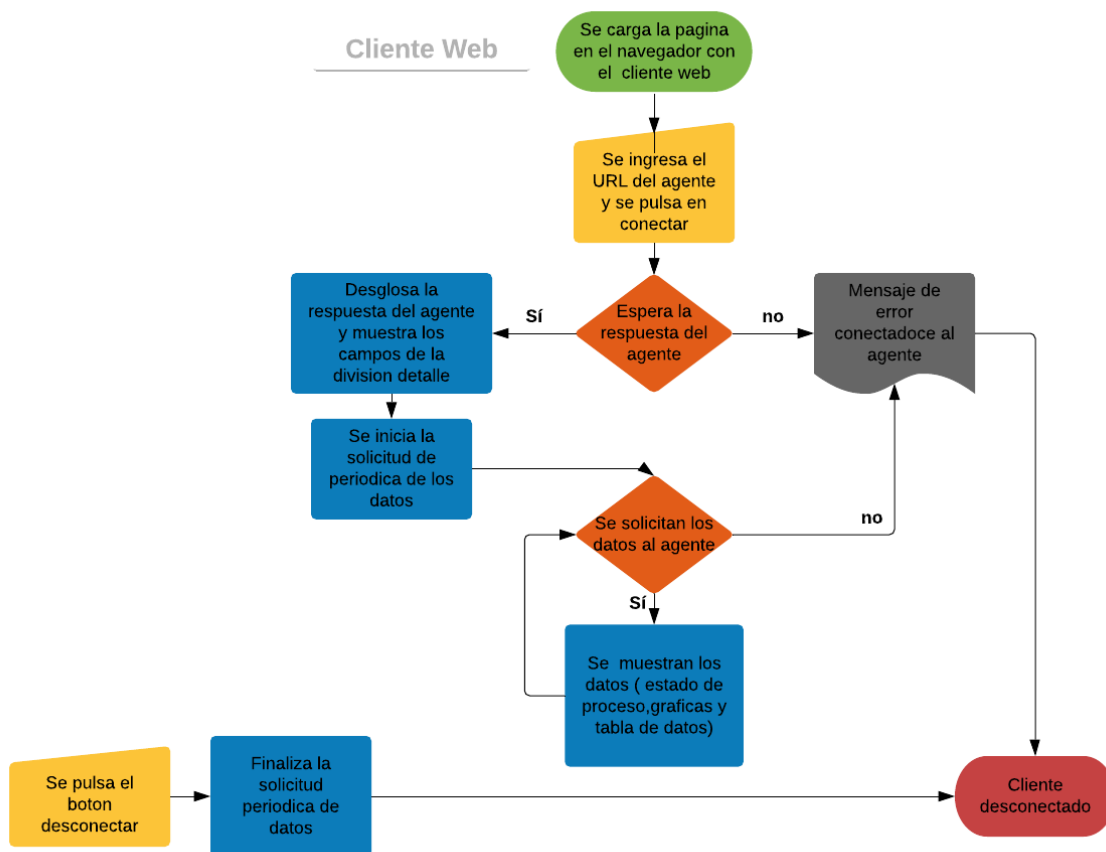


Figura 30. Diagrama de flujo para la programación del cliente.

La forma en que se solicitan los datos al agente es utilizando un objeto XMLHttpRequest, este objeto JavaScript permite interactuar con servidores mediante una URL sin tener que recargar la página por completo. La función conectar() internamente crea un objeto de este tipo solicitando los datos al URL que el usuario del cliente web proporciono, a este objeto se le carga el parametro onreadystatechange con una función que se ejecutara cada vez que el parámetro readyState del objeto cambia. Dentro de la función pasada al onreadystatechange se evalúa la respuesta del del URL al cual se solicitaron los datos, si esta es correcta y además es un documento XML, se desglosa el documento por el nombre de las etiquetas para poder cargar los valores que se muestran en la división detalle de la interfaz del cliente.

- **iniciarSolicitudDatos():** Agrega una función llamada solicitarDatos() al método de JavaScript setInterval(), este método ejecutara esta función periódicamente según el parámetro de milisegundos pasado.
- **iniciarGraficaDatos():** Agrega una función llamada graficarDatos() al método de JavaScript setInterval(), este método ejecutara esta función periódicamente según el parámetro de milisegundos pasado.
- **solicitarDatos():** Al igual que la función conectar utiliza un objeto XMLHttpRequest para comunicarse con el agente solicitando los datos para mostrarlos en la división

Estado del proceso, graficados por la función `graficarDatos()` y agregados a la tabla de la división Datos.

- **graficarDatos():** Esta función grafica los datos de posiciones y temperaturas que la función `solicitarDatos()` obtiene del agente por medio de la librería para JavaScript `Chart.js` y de ejecutar la función `agregarDatosTabla()`, cuando los datos en la gráfica superan la cantidad máxima de 20, el primer dato es borrado de la gráfica agregando el siguiente dato entrante.
- **agregarDatosTabla():** esta función se encarga de agregar los datos a la tabla que se encuentra en la división Datos del cliente web provenientes del agente, esta función agrega una nueva fila a la tabla y luego las celdas donde estarán los datos de posición y temperatura de la máquina.

Las funciones para desconectarse del agente finalizando las solicitudes de datos y la graficas de estos son `finalizarSolicitudDatos()` y `finalizarGraficaDatos()`, esta función utiliza el método de JavaScript `clearInterval()` pasando como parámetro las variables de control para los métodos `setInterval()` que se crean para cada función al ser lanzados estos métodos grafican.

5. RESULTADOS: EXPERIMENTOS Y VALIDACIÓN

Los resultados fueron tomados a partir de las validaciones que se llevaron a cabo en los laboratorios de robótica de la Universidad de Pamplona (UP) y los laboratorios de mecánica de la Universidad de Brasilia (UNB), Brasil. Se empleo una impresora 3D modelo Prusa i2 a la cual se le cargo el firmware modificado con el adaptador inmerso, conectado a un servidor proporcionado por la UNB y monitoreando el proceso de fabricación de una pieza a través de un cliente web desarrollado en HTML y JavaScript ejecutándose desde un navegador de un PC remoto.

El primer paso fue cargar el Firmware Sprinter modificado con el adaptador a la máquina, luego conectando este a un pc el cual tiene instalado el agente MTConnect y este PC al servidor que proporciona una dirección IP global para lograr la conexión desde un PC externo a la red local donde se encuentra el agente. Para comprobar el servicio del agente en el PC funcionara correctamente se ejecutó en el navegador la ruta <http://localhost:5000> el cual devolvió como respuesta un XML con la estructura dada para la máquina como se muestra en la Figura 30, en la Figura 31 se muestra la respuesta devuelta por el agente ejecutando el URL <http://localhost:5000/sample> en el navegador web y en la Figura 32 se muestra la respuesta devuelta por el agente ejecutando el URL <http://localhost:5000/current> en el navegador web.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<MTConnectDevices xmlns:m="urn:mtconnect.org:MTConnectDevices:1.3" xmlns="urn:mtconnect.org:MTConnectDevices:1.3"
xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.3 http://www.mtconnect.org/schemas/MTConnectDevices_1.3.x"
  <Header creationTime="2019-07-09T15:18:47Z" sender="Juan-Rodriguez" instanceId="1562681966" version="1.3.0.22" a
  <Devices>
    <Device id="PrusaMendel" iso841Class="6" name="PrusaMendel" sampleInterval="10" uuid="PrusaMendel001">
      <Description manufacturer="RepRap">Prusa Mendel</Description>
      <DataItems>
        <DataItem category="EVENT" id="PrusaMendelavail" type="AVAILABILITY"/>
        <DataItem category="EVENT" id="PrusaMendel_asset_chg" type="ASSET_CHANGED"/>
        <DataItem category="EVENT" id="PrusaMendel_asset_rem" type="ASSET_REMOVED"/>
      </DataItems>
      <Components>
        <Axes id="PrusaMendel-axes" name="Axes">
          <Components>
            <Linear id="PrusaMendel-X" name="X">
              <DataItems>
                <DataItem category="SAMPLE" id="PrusaMendel-x" name="Xact" nativeUnits="MILLIMETER" subType="ACTUAL"
              </DataItems>
            </Linear>
            <Linear id="PrusaMendel-Y" name="Y">
              <DataItems>
                <DataItem category="SAMPLE" id="PrusaMendel-y" name="Yact" nativeUnits="MILLIMETER" subType="ACTUAL"
              </DataItems>
            </Linear>
            <Linear id="PrusaMendel-Z" name="Z">
              <DataItems>
                <DataItem category="SAMPLE" id="PrusaMendel-z" name="Zact" nativeUnits="MILLIMETER" subType="ACTUAL"
              </DataItems>
            </Linear>
            <Linear id="PrusaMendel-E" name="E">
              <DataItems>
                <DataItem category="SAMPLE" id="PrusaMendel-e" name="Extruder" nativeUnits="MILLIMETER" subType="AC"
              </DataItems>
            </Linear>
          </Components>
        </Axes>
      </Components>
    </Device>
  </Devices>
```

Figura 31. Respuesta XML por el agente para el navegador web.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<HTConnectStreams xmlns:mtconnect="urn:mtconnect.org:HTConnectStreams:1.3" xmlns="urn:mtconnect.org:HTConnectStreams:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.org:HTConnectStreams:1.3 http://www.mtconnect.org/schemas/HTConnectStreams_1.3.xsd">
  <Header creationTime="2019-07-09T16:27:09Z" sender="Juan-Rodríguez" instanceId="1562691256" version="1.3.0.22" bufferSize="131072" nextSequence="101" firstSeqLastSequence="215"/>
  <Streams>
    <DeviceStream name="PrusaMendel" uuid="PrusaMendel001">
      <ComponentStream component="Device" name="PrusaMendel" componentId="PrusaMendel">
        <Events>
          <AssetChanged dataItemId="PrusaMendel_asset_chg" timestamp="2019-07-09T16:54:16.401197Z" sequence="6" assetType="">UNAVAILABLE</AssetChanged>
          <AssetRemoved dataItemId="PrusaMendel_asset_rem" timestamp="2019-07-09T16:54:16.401197Z" sequence="7" assetType="">UNAVAILABLE</AssetRemoved>
          <Availability dataItemId="PrusaMendelavail" timestamp="2019-07-09T16:54:16.401197Z" sequence="8">UNAVAILABLE</Availability>
        </Events>
      </ComponentStream>
      <ComponentStream component="Linear" name="E" componentId="PrusaMendel-E">
        <Samples>
          <Position dataItemId="PrusaMendel-e" timestamp="2019-07-09T16:54:16.401197Z" name="Extruder" sequence="2" subType="ACTUAL">UNAVAILABLE</Position>
          <Position dataItemId="PrusaMendel-e" timestamp="2019-07-09T16:54:16.611209Z" name="Extruder" sequence="15" subType="ACTUAL">0.0</Position>
          <Position dataItemId="PrusaMendel-e" timestamp="2019-07-09T17:23:50.637678Z" name="Extruder" sequence="84" subType="ACTUAL">UNAVAILABLE</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Linear" name="X" componentId="PrusaMendel-X">
        <Samples>
          <Position dataItemId="PrusaMendel-x" timestamp="2019-07-09T16:54:16.401197Z" name="Xact" sequence="3" subType="ACTUAL">UNAVAILABLE</Position>
          <Position dataItemId="PrusaMendel-x" timestamp="2019-07-09T16:54:16.611209Z" name="Xact" sequence="12" subType="ACTUAL">0.0</Position>
          <Position dataItemId="PrusaMendel-x" timestamp="2019-07-09T17:23:50.637678Z" name="Xact" sequence="85" subType="ACTUAL">UNAVAILABLE</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Linear" name="Y" componentId="PrusaMendel-Y">
        <Samples>
          <Position dataItemId="PrusaMendel-y" timestamp="2019-07-09T16:54:16.401197Z" name="Yact" sequence="4" subType="ACTUAL">UNAVAILABLE</Position>
          <Position dataItemId="PrusaMendel-y" timestamp="2019-07-09T16:54:16.611209Z" name="Yact" sequence="13" subType="ACTUAL">0.0</Position>
          <Position dataItemId="PrusaMendel-y" timestamp="2019-07-09T17:23:50.637678Z" name="Yact" sequence="86" subType="ACTUAL">UNAVAILABLE</Position>
        </Samples>
      </ComponentStream>
    </DeviceStream>
  </Streams>
</HTConnectStreams>
```

Figura 32. Respuesta XML sample devuelta por el agente para el navegador web.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<HTConnectStreams xmlns:mtconnect="urn:mtconnect.org:HTConnectStreams:1.3" xmlns="urn:mtconnect.org:HTConnectStreams:1.3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="urn:mtconnect.org:HTConnectStreams:1.3 http://www.mtconnect.org/schemas/HTConnectStreams_1.3.xsd">
  <Header creationTime="2019-07-09T18:56:21Z" sender="Juan-Rodríguez" instanceId="1562698567" version="1.3.0.22" bufferSize="131072" nextSequence="20" firstSeqLastSequence="19"/>
  <Streams>
    <DeviceStream name="PrusaMendel" uuid="PrusaMendel001">
      <ComponentStream component="Device" name="PrusaMendel" componentId="PrusaMendel">
        <Events>
          <AssetChanged dataItemId="PrusaMendel_asset_chg" timestamp="2019-07-09T18:56:07.185477Z" sequence="6" assetType="">UNAVAILABLE</AssetChanged>
          <AssetRemoved dataItemId="PrusaMendel_asset_rem" timestamp="2019-07-09T18:56:07.185477Z" sequence="7" assetType="">UNAVAILABLE</AssetRemoved>
          <Availability dataItemId="PrusaMendelavail" timestamp="2019-07-09T18:56:07.185477Z" sequence="8">UNAVAILABLE</Availability>
        </Events>
      </ComponentStream>
      <ComponentStream component="Linear" name="E" componentId="PrusaMendel-E">
        <Samples>
          <Position dataItemId="PrusaMendel-e" timestamp="2019-07-09T18:56:07.405489Z" name="Extruder" sequence="15" subType="ACTUAL">0.0</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Linear" name="X" componentId="PrusaMendel-X">
        <Samples>
          <Position dataItemId="PrusaMendel-x" timestamp="2019-07-09T18:56:07.405489Z" name="Xact" sequence="12" subType="ACTUAL">0.0</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Linear" name="Y" componentId="PrusaMendel-Y">
        <Samples>
          <Position dataItemId="PrusaMendel-y" timestamp="2019-07-09T18:56:07.405489Z" name="Yact" sequence="13" subType="ACTUAL">0.0</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Linear" name="Z" componentId="PrusaMendel-Z">
        <Samples>
          <Position dataItemId="PrusaMendel-z" timestamp="2019-07-09T18:56:07.405489Z" name="Zact" sequence="14" subType="ACTUAL">0.0</Position>
        </Samples>
      </ComponentStream>
      <ComponentStream component="Sensor" name="BedSensor" componentId="bedSensor">
        <Samples>
          <Temperature dataItemId="bedTemp" timestamp="2019-07-09T18:56:16.985037Z" name="bed temp sensor" sequence="19">22</Temperature>
        </Samples>
      </ComponentStream>
    </DeviceStream>
  </Streams>
</HTConnectStreams>
```

Figura 33. Respuesta XML current devuelta por el agente para el navegador web.

Para conectarse con el cliente web este se ejecutó en el navegador web la IP suministrada por el servidor para el pc con el agente instalado seleccionando el puerto 5000 para la conexión <http://IpPcAgente:5000> este dio como resultado los datos de impresión provenientes de la máquina que ejecutaba el adaptador como se muestra en la Figura 33. El proceso de monitoreo del cliente web para la impresión de la pieza pueden ser visto en los videos a través de los siguientes links.

<https://www.youtube.com/watch?v=55VilJ53d5I>

<https://www.facebook.com/monkeywit/videos/322142985350947/>

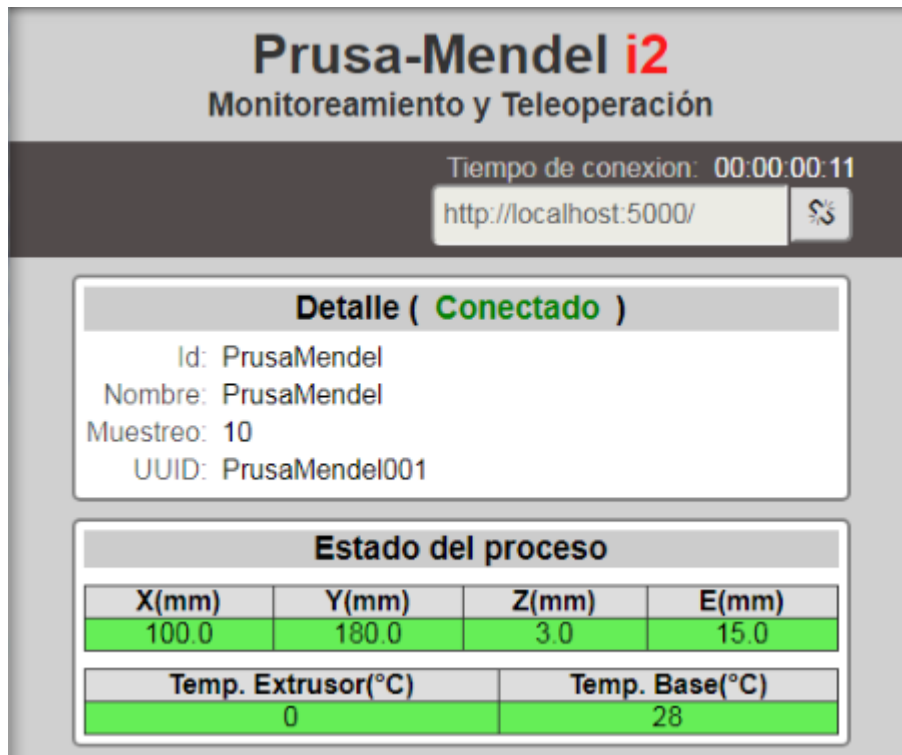


Figura 34. Datos vistos desde el cliente web para la impresión de la pieza.

5.1. VELOCIDAD DE COMUNICACIÓN

Para verificar la velocidad de comunicación entre el agente y el cliente web fue usado el software Wireshark, este software es un analizador de protocolos de red que permite la visualización a bajo nivel los paquetes TCP/IP entrantes y saliente de la tarjeta de red del PC, proporcionando información importante como el tiempo relativo entre un paquete y otro. Aplicando un filtro para la IP del agente en el software se midió el tiempo de respuesta entre el cliente y el agente, así como entre el agente y el adaptador.

5.1.1. Comunicación Adaptador–Agente.

En la figura 34 y 35 se muestra una imagen que ilustra la comunicación entre el agente y el adaptador para el manejo Heartbeat entre ambos.

Time	Source	Destination	Protocol	Length	Info
16696	1585.145609	192.168.10.72	TCP	54	49824 → 7878 [ACK] Seq=1331 Ack=23181 Win=16274 Len=0
16697	1585.598043	192.168.10.72	TCP	61	49824 → 7878 [PSH, ACK] Seq=1331 Ack=23181 Win=16274 Len=7
16698	1585.622573	192.168.10.73	TCP	60	7878 → 49824 [ACK] Seq=23181 Ack=1338 Win=1024 Len=0
16699	1585.628867	192.168.10.73	TCP	67	7878 → 49824 [PSH, ACK] Seq=23181 Ack=1338 Win=1024 Len=13
16700	1585.647062	192.168.10.73	TCP	163	7878 → 49824 [PSH, ACK] Seq=23194 Ack=1338 Win=1024 Len=109
16701	1585.647139	192.168.10.72	TCP	54	49824 → 7878 [ACK] Seq=1338 Ack=23303 Win=16152 Len=0
0000	74 69 69 2d 30 31 c8 0a a9 e0 f0 e0 08 00 45 00		tii-01...E	
0010	00 2f 4e a2 40 00 80 06 16 45 c0 a8 0a 48 c0 a8		./N:@...	E...H..	
0020	0a 49 c2 a0 1e c6 48 3a d8 9e 00 00 64 8d 50 18		-I...H:d-P	
0030	3f 92 a0 d3 00 00 2a 20 50 49 4e 47 0a		?.....*	PING-	

Figura 35. Paquete con PING del agente al adaptador.

No.	Time	Source	Destination	Protocol	Length	Info
16696	1585.145609	192.168.10.72	192.168.10.73	TCP	54	49824 → 7878 [ACK] Seq=1331 Ack=23181 Win=16274 Len=0
16697	1585.598043	192.168.10.72	192.168.10.73	TCP	61	49824 → 7878 [PSH, ACK] Seq=1331 Ack=23181 Win=16274 Len=7
16698	1585.622573	192.168.10.73	192.168.10.72	TCP	60	7878 → 49824 [ACK] Seq=23181 Ack=1338 Win=1024 Len=0
16699	1585.628867	192.168.10.73	192.168.10.72	TCP	67	7878 → 49824 [PSH, ACK] Seq=23181 Ack=1338 Win=1024 Len=13
16700	1585.647062	192.168.10.73	192.168.10.72	TCP	163	7878 → 49824 [PSH, ACK] Seq=23194 Ack=1338 Win=1024 Len=109
16701	1585.647139	192.168.10.72	192.168.10.73	TCP	54	49824 → 7878 [ACK] Seq=1338 Ack=23303 Win=16152 Len=0

```

0000 c8 0a a9 e0 f0 e0 74 69 69 2d 30 31 08 00 45 00 .....ti i-01..E.
0010 00 35 4e a2 40 00 40 06 56 3f c0 a8 0a 49 c0 a8 ....@.@. ....I..
0020 0a 48 1e c6 c2 a0 00 00 64 8d 48 3a d8 a5 50 18 .H.....d:H:..P.
0030 04 00 7e e3 00 00 0a 2a 20 50 4f 4e 47 20 35 30 .....* PONG 50
0040 30 0d 0a

```

Figura 36. Paquete con PONG del adaptador al agente.

De las marcas de tiempo en ambas Figura 34 y 35 se observa que los tiempos de respuesta entre PING y PONG son de 30.824 milisegundos aproximadamente. El tiempo de respuesta entre un paquete con los datos de los sensores y el siguiente se muestra en la Figura 36 y 37.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.063443	192.168.10.73	192.168.10.72	TCP	156	7878 → 62385 [PSH, ACK] Seq=14 Ack=8 Win=1024 Len=102
5	0.063546	192.168.10.72	192.168.10.73	TCP	54	62385 → 7878 [ACK] Seq=8 Ack=116 Win=15720 Len=0
6	0.512002	192.168.10.72	192.168.10.73	TCP	61	62385 → 7878 [PSH, ACK] Seq=8 Ack=116 Win=15720 Len=7
7	0.541670	192.168.10.73	192.168.10.72	TCP	60	7878 → 62385 [ACK] Seq=116 Ack=15 Win=1024 Len=0
8	0.546766	192.168.10.73	192.168.10.72	TCP	67	7878 → 62385 [PSH, ACK] Seq=116 Ack=15 Win=1024 Len=13
9	0.565048	192.168.10.73	192.168.10.72	TCP	156	7878 → 62385 [PSH, ACK] Seq=129 Ack=15 Win=1024 Len=102
10	0.565158	192.168.10.72	192.168.10.73	TCP	54	62385 → 7878 [ACK] Seq=15 Ack=231 Win=15605 Len=0
11	1.018004	192.168.10.72	192.168.10.73	TCP	61	62385 → 7878 [PSH, ACK] Seq=15 Ack=231 Win=15605 Len=7
12	1.042192	192.168.10.73	192.168.10.72	TCP	60	7878 → 62385 [ACK] Seq=231 Ack=22 Win=1024 Len=0

```

0000 c8 0a a9 e0 f0 e0 74 69 69 2d 30 31 08 00 45 00 .....ti i-01..E.
0010 00 8e 09 cc 40 00 40 06 9a bc c0 a8 0a 49 c0 a8 ....@.@. ....I..
0020 0a 48 1e c6 f3 b1 00 00 8c a2 4b 2f b0 4b 50 18 .H.....K/.KP.
0030 04 00 fd df 00 00 7c 62 75 69 6c 64 20 70 72 6f .....|b uild pro
0040 67 72 65 73 73 7c 30 25 7c 58 61 63 74 7c 30 2e gress|0% |Xact|0.
0050 30 7c 59 61 63 74 7c 30 2e 30 7c 5a 61 63 74 7c 0|Yact|0 .0|Zact|
0060 30 2e 30 7c 45 78 74 72 75 64 65 72 7c 30 2e 30 0.0|Extr uder|0.0
0070 7c 65 78 74 72 75 64 65 72 20 74 65 6d 70 20 73 |extrude r temp s
0080 65 6e 73 6f 72 7c 30 7c 62 65 64 20 74 65 6d 70 ensor|0| bed temp
0090 20 73 65 6e 73 6f 72 7c 32 30 0d 0a sensor| 20..

```

Figura 37. Primer paquete de datos entre el agente y el adaptador.

No.	Time	Source	Destination	Protocol	Length	Info
4	0.063443	192.168.10.73	192.168.10.72	TCP	156	7878 → 62385 [PSH, ACK] Seq=14 Ack=8 Win=1024 Len=102
5	0.063546	192.168.10.72	192.168.10.73	TCP	54	62385 → 7878 [ACK] Seq=8 Ack=116 Win=15720 Len=0
6	0.512002	192.168.10.72	192.168.10.73	TCP	61	62385 → 7878 [PSH, ACK] Seq=8 Ack=116 Win=15720 Len=7
7	0.541670	192.168.10.73	192.168.10.72	TCP	60	7878 → 62385 [ACK] Seq=116 Ack=15 Win=1024 Len=0
8	0.546766	192.168.10.73	192.168.10.72	TCP	67	7878 → 62385 [PSH, ACK] Seq=116 Ack=15 Win=1024 Len=13
9	0.565048	192.168.10.73	192.168.10.72	TCP	156	7878 → 62385 [PSH, ACK] Seq=129 Ack=15 Win=1024 Len=102
10	0.565158	192.168.10.72	192.168.10.73	TCP	54	62385 → 7878 [ACK] Seq=15 Ack=231 Win=15605 Len=0

```

0000 c8 0a a9 e0 f0 e0 74 69 69 2d 30 31 08 00 45 00 .....ti i-01..E.
0010 00 8e 09 cc 40 00 40 06 9a bc c0 a8 0a 49 c0 a8 ....@.@. ....I..
0020 0a 48 1e c6 f3 b1 00 00 8c a2 4b 2f b0 4b 50 18 .H.....K/.KP.
0030 04 00 fd df 00 00 7c 62 75 69 6c 64 20 70 72 6f .....|b uild pro
0040 67 72 65 73 73 7c 30 25 7c 58 61 63 74 7c 30 2e gress|0% |Xact|0.
0050 30 7c 59 61 63 74 7c 30 2e 30 7c 5a 61 63 74 7c 0|Yact|0 .0|Zact|
0060 30 2e 30 7c 45 78 74 72 75 64 65 72 7c 30 2e 30 0.0|Extr uder|0.0
0070 7c 65 78 74 72 75 64 65 72 20 74 65 6d 70 20 73 |extrude r temp s
0080 65 6e 73 6f 72 7c 30 7c 62 65 64 20 74 65 6d 70 ensor|0| bed temp
0090 20 73 65 6e 73 6f 72 7c 32 30 0d 0a sensor| 20..

```

Figura 38. Segundo paquete de datos entre el adaptador y el agente.

De las marcas de tiempo de la figura 36 y 37 se observa el tiempo entre un paquete y el siguiente enviado por el adaptador es de 501.605 milisegundos aproximadamente. Se tomaron 15 datos seguidos para evaluar la diferencia de tiempos entre un paquete de datos y el siguiente sacando un valor promedio de tiempo de 501.528 milisegundos, gráfica de datos vs tiempo se muestra en la Figura 38.



Figura 39. Grafica para la diferencia de tiempo entre datos para la comunicación adaptador-agente.

5.1.2. Comunicación Agente-Cliente web

La comunicación entre agente y el cliente maneja un tiempo base de 100 milisegundos, este tiempo viene dado por la función `setInterval()` de JavaScript la cual se ejecuta periódicamente. La velocidad de comunicación entre el agente y el cliente web puede ser afectada por la distancia de donde se conecte el cliente y la velocidad de la red mediante la que se conectó. En la Figura 39 se muestran el paquete TCP/IP del cliente web analizado con Wireshark enviado al agente solicitando los datos y en la Figura 40 se muestra el paquete de respuesta del agente al cliente, este paquete contiene el documento XML con los datos.

No.	Time	Source	Destination	Protocol	Length	Info
5	1.228062	192.168.0.14	192.168.0.12	TCP	66	50543 → 5000 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
8	1.325885	192.168.0.12	192.168.0.14	TCP	66	5000 → 50543 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 S
9	1.326117	192.168.0.14	192.168.0.12	TCP	54	50543 → 5000 [ACK] Seq=1 Ack=1 Win=17520 Len=0
10	1.327996	192.168.0.14	192.168.0.12	HTTP	327	GET / HTTP/1.1 Solicitud del cliente al agente
11	1.338446	192.168.0.12	192.168.0.14	TCP	1514	5000 → 50543 [ACK] Seq=1 Ack=274 Win=65536 Len=1460 [TCP segment of
12	1.343484	192.168.0.12	192.168.0.14	TCP	1514	5000 → 50543 [ACK] Seq=1461 Ack=274 Win=65536 Len=1460 [TCP segment
13	1.343618	192.168.0.14	192.168.0.12	TCP	54	50543 → 5000 [ACK] Seq=274 Ack=2921 Win=17520 Len=0
14	1.345948	192.168.0.12	192.168.0.14	HTTP/X..	140	HTTP/1.0 200 OK
15	1.346106	192.168.0.14	192.168.0.12	TCP	54	50543 → 5000 [ACK] Seq=274 Ack=3008 Win=17432 Len=0
16	1.347994	192.168.0.14	192.168.0.12	TCP	54	50543 → 5000 [FIN, ACK] Seq=274 Ack=3008 Win=17432 Len=0
17	1.351022	192.168.0.12	192.168.0.14	TCP	54	5000 → 50543 [ACK] Seq=3008 Ack=275 Win=65536 Len=0


```

0000 e8 39 df 3e e9 90 e4 f8 9c 82 10 9f 08 00 45 00  9 >.....E.
0010 01 39 4e ca 40 00 80 06 29 8a c0 a8 00 0e c0 a8  9N@.....).....
0020 00 0c c5 6f 13 88 c3 86 4f 11 9b d1 26 92 50 18  .....o.....O...&P.
0030 11 1c c3 6f 00 00 47 45 54 20 2f 20 48 54 54 50  .....o...GE T / HTTP
0040 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 39 32 2e  /1.1..Host: 192.
0050 31 36 38 2e 30 2e 31 32 3a 35 30 30 30 0d 0a 43  168.0.12 :5000..C
0060 6f 6e 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 70 2d  onnection: keep-
0070 61 6c 69 76 65 0d 0a 55 73 65 72 2d 41 67 65 6e  alive..U ser-Agen
0080 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28  t: Mozilla/5.0 (
0090 57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 31 3b 20  Windows NT 6.1;

```

Figura 40. Solicitud del cliente web para agente vista desde el PC del cliente.

```

5 1.228062 192.168.0.14 192.168.0.12 TCP 66 50543 → 5000 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=
8 1.325885 192.168.0.12 192.168.0.14 TCP 66 5000 → 50543 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=2
9 1.326117 192.168.0.14 192.168.0.12 TCP 54 50543 → 5000 [ACK] Seq=1 Ack=1 Win=17520 Len=0
10 1.327996 192.168.0.14 192.168.0.12 HTTP 327 GET / HTTP/1.1
11 1.338446 192.168.0.12 192.168.0.14 TCP 1514 5000 → 50543 [ACK] Seq=1 Ack=274 Win=65536 Len=1460 [TCP segment
12 1.343484 192.168.0.12 192.168.0.14 TCP 1514 5000 → 50543 [ACK] Seq=1461 Ack=274 Win=65536 Len=1460 [TCP segm
13 1.343618 192.168.0.14 192.168.0.12 TCP 54 50543 → 5000 [ACK] Seq=274 Ack=2921 Win=17520 Len=0
14 1.345948 192.168.0.12 192.168.0.14 HTTP/x... 140 HTTP/1.0 200 OK Respuesta del agente al cliente
15 1.346106 192.168.0.14 192.168.0.12 TCP 54 50543 → 5000 [ACK] Seq=274 Ack=3008 Win=17432 Len=0
16 1.347994 192.168.0.14 192.168.0.12 TCP 54 50543 → 5000 [FIN, ACK] Seq=274 Ack=3008 Win=17432 Len=0
17 1.351022 192.168.0.12 192.168.0.14 TCP 54 5000 → 50543 [ACK] Seq=3008 Ack=275 Win=65536 Len=0
0000 e4 f8 9c 82 10 9f e8 39 df 3e e9 90 08 00 45 00 .....9 ->....E.
0010 05 dc 20 a1 40 00 00 06 53 10 c0 a8 00 0c c0 a8 ..@...S.....
0020 00 0e 13 88 c5 6f 9b d1 26 92 c3 86 50 22 50 10 .....o...&...P"p
0030 01 00 f2 b3 00 00 48 54 54 50 2f 31 2e 30 20 32 .....HT TP/1.0 2
0040 30 30 20 4f 4b 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 00 OK..Content-L
0050 65 6e 67 74 68 3a 20 32 39 34 31 0d 0a 43 6f 6e 65 6e 67 74 68 3a 20 32 39 34 31 0d 0a 43 6f 6e
0060 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 2f content-Type: text/
0070 78 6d 6c 0d 0a 0d 0a 3c 3f 78 6d 6c 20 76 65 72 xml:....<?xml ver
0080 73 69 6f 6e 3d 22 31 2e 30 22 20 65 6e 63 6f 64 sion="1.0" encod
0090 69 6e 67 3d 22 55 54 46 2d 38 22 3f 3e 0a 3c 4d ing="UTF-8"?><M

```

Figura 41. Respuesta del agente al cliente web vista desde el PC del cliente.

De las Figuras 39 y 40 se observa que el tiempo de respuesta de una solicitud de datos por parte del cliente al agente tarda al aproximadamente 15.622 milisegundos, este tiempo puede variar según la velocidad que tenga la conexión a internet y el tamaño de los datos con que responda el agente.

5.1.3. Sincronización entre el adaptador y el agente

Al reiniciar la computadora donde se encuentra el agente instalado y conectar el adaptador este se pierde muchas veces, causando que los datos de la maquina no sean capturados por el agente y como resultado el cliente web no muestre incorrectos como se muestra en la figura 41.

Time of connection: 00:00:00:20 <http://localhost:5000/>

Detalle (Conectado)

Id: PrusaMendel
Nombre: PrusaMendel
Muestreo: 10
UUID: PrusaMendel001

Estado del proceso

X(mm)	Y(mm)	Z(mm)	E(mm)
UNAVAILABLE	UNAVAILABLE	UNAVAILABLE	UNAVAILABLE

Temp. Extrusor(°C)	Temp. Base(°C)
UNAVAILABLE	UNAVAILABLE

Figura 42. Fallo de sincronización entre los datos del agente y el cliente.

A continuación, se establecieron una serie de pasos para la correcta sincronización entre el adaptador y el agente:

- Primero se debe detener el servicio del agente en la computadora.
- Reiniciar el controlador de la maquina y conectarla al software el cual se use para carga el código G a la máquina, esta conexión siempre reinicia el controlador nuevamente, por eso es necesario que se haga antes de echar andar nuevamente el servicio del agente.
- Iniciar el servicio del agente.

5.2. Publicaciones

Parte del trabajo desarrollado será próximamente publicado en la revista IFAC PapersOnline de Elsevier, el artículo para esta publicación se muestra en el anexo C. Todos los artículos de las reuniones de la IFAC se publican, en asociación con Elsevier, el editor de la IFAC, en la serie de procedimientos de la IFAC-PapersOnLine alojada en el servicio web ScienceDirect.

6. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo ha introducido el desarrollo de un método para conectar máquinas de manufactura aditiva del tipo RepRap a internet usando el protocolo estándar industrial MTConnect. El trabajo se da en un momento donde la llegada de la cuarta revolución industrial trae importantes desafíos al campo de la manufactura que deben ser enfrentados con la proposición y desarrollo de nuevas plataformas de sistemas avanzados de manufactura usando tecnologías industriales estándares, abiertas y escalables.

Se propuso una arquitectura de implementación del estándar MTConnect que consiste en tener el componente adaptador embebido en el sistema de las impresoras 3D RepRap que usan tecnología de Arduino. Esta arquitectura difiere de la mayoría de arquitecturas que usan el adaptador separado del sistema de la máquina, primeramente, debido a que la mayoría de los controladores CNC comerciales son de tecnología cerrada, y segundo, porque se muestra más complaciente con lo propuesto por el estándar. Sin embargo, el adaptador de la arquitectura desarrollada en este trabajo, el cual es formado por un componente de software incorporado en el firmware de la máquina y un componente de hardware constituido por una placa de módulo Ethernet, da ventajas en términos de reducción de costo computacional y buena velocidad de comunicación. Esto se debe principalmente al hecho de que no necesita de intermediarios para sacar los datos de la máquina y publicarlos a través del agente MTConnect. No obstante, debido a las limitaciones del controlador Arduino Mega la incorporación del código del adaptador, que usa las librerías Ethernet, interfiere con algunos de los procesos internos del firmware que utilizan los timers del controlador. Esto se ha tornado algo tedioso de resolver para el firmware Sprinter, mientras para el firmware Marlin no fue posible de solucionar ya que la capacidad del Arduino Mega es llevada al máximo por el extenso y complejo de tal firmware.

Los resultados obtenidos han demostrado el gran potencial del sistema propuesto para acceso a datos y monitoreamiento remoto de impresoras 3D en ambientes de sistemas de producción ciber-físicos. El framework desarrollado tiene la capacidad de recuperar datos del proceso de impresión de una parte desde la máquina y publicarlos en internet para un cliente web que los requiera. Tres tipos de requerimiento de datos son posibles a través del framework: probe, current y sample.

El cliente web desarrollado para la demostración ha permitido realizar requerimientos de datos de la máquina a través del MTConnect. El tiempo de respuesta a un requerimiento hecho por el cliente dentro de una red local no a supera 100 milisegundos. Por otro lado, el tiempo de respuesta para un requerimiento de desde redes diferentes y distantes, como las realizadas desde el cliente en Colombia a una máquina en el LaDPRER, en Brasil, ha estado alrededor de 1 segundo. Esto supone un buen resultado si se tiene en cuenta que factores como la capacidad de computador servidor y la velocidad del internet siendo usados, influyen en la velocidad y la calidad de la conexión del framework. Otras pruebas de robustez y estabilidad del sistema fueron llevadas a cabo con la impresión de varias piezas durante muchas horas seguidas.

La contribución de este trabajo resalta en el aporte al desarrollo del estándar MTConnect con un framework para monitoreamiento remoto aplicado a impresoras 3D como las RepRap, las cuales son ampliamente usadas alrededor del mundo por pequeñas y medianas empresas, así como por investigadores y aficionados de la impresión 3D. El enfoque de este trabajo puede ser aplicado a otras máquinas CNC que también usen tecnología abierta de Arduino o alguna otra placa de desarrollo libre que pueda ser adaptada.

Trabajos futuros pretenden habilitar el Sistema con operaciones de control remoto y teleoperación vía internet agregando el protocolo OPC-UA, mientras MTConnect solo es destinado para el monitoreamiento y acceso a datos de proceso. La idea detrás de esto es construir un completo sistema de manufactura ciber-física auxiliado por DigitalTwin dentro del contexto de los trabajos de investigación siendo desarrollados en el LaDPRER-UnB-Brasil.

REFERENCIAS BIBLIOGRÁFICAS

- [1] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent Manufacturing in the Context of Industry 4.0: A Review," *Engineering*, vol. 3, no. 5, pp. 616–630, Oct. 2017.
- [2] C. Liu and X. Xu, "Cyber-physical Machine Tool – The Era of Machine Tool 4.0," *Procedia CIRP*, vol. 63, pp. 70–75, 2017.
- [3] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld, and P. Warndorf, "Improving Machine Tool Interoperability Using Standardized Interface Protocols: MTConnect," in *Proceedings of 2008 ISFA 2008 International Symposium on Flexible Automation*, 2008, pp. 1–6.
- [4] J. Moilanen and T. Vadén, "3D printing community and emerging practices of peer production," *First Monday*, vol. 18, no. 8, Aug. 2013.
- [5] A. C. Gil, *Como elaborar projetos de pesquisa*, 6th ed. São Paulo, 2017.
- [6] H. Kagermann, W. Wahister, and J. Helbig, "Recommendations for implementing the strategic initiative INDUSTRIE 4.0: Final report of the Industrie 4.0 Working Group," 2013.
- [7] M. Hankel and B. Rexroth, "The Reference Architectural Model Industrie 4.0 (RAMI 4.0)," *ZVEI*, pp. 1–2, 2015.
- [8] S. A. Coons, "An outline of the requirements for a computer-aided design system," in *Proceedings of the May 21-23, 1963, spring joint computer conference on - AFIPS '63 (Spring)*, 1963, p. 299.
- [9] A. A. Tseng, S. P. Kolluri, and P. Radhakrishnan, "A CNC machining system for education," *J. Manuf. Syst.*, vol. 8, no. 3, pp. 207–214, Jan. 1989.
- [10] X. Xu, *Integrating Advanced Computer-Aided Design, Manufacturing, and Numerical Control*. IGI Global, 2009.
- [11] D. LEGATIUK, M. THEILER, K. DRAGOS, and K. SMARSLY, "A Categorical Approach Towards Metamodeling Cyber-physical Systems," in *Structural Health Monitoring 2017*, 2017.
- [12] T. Fitz, M. Theiler, and K. Smarsly, "A metamodel for cyber-physical systems," *Adv. Eng. Informatics*, vol. 41, p. 100930, Aug. 2019.
- [13] J. Wan, "Advances in Cyber-Physical Systems Research," *KSII Trans. Internet Inf. Syst.*, vol. 5, no. 11, 2011.
- [14] J. Shi, J. Wan, H. Yan, and H. Suo, "A survey of Cyber-Physical Systems," in *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2011, pp. 1–6.
- [15] J. Lee, B. Bagheri, and H.-A. Kao, "A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems," *Manuf. Lett.*, vol. 3, pp. 18–23, Jan. 2015.
- [16] E. Rodriguez and A. J. Alvares, "A STEP-NC implementation approach for additive manufacturing," *Procedia Manuf.*, vol. 1, no. 1, pp. 1–8, 2019.
- [17] R. Bonnard, J.-Y. Hascoët, P. Mognol, and I. Stroud, "STEP-NC digital thread for additive manufacturing: data model, implementation and validation," *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 11, pp. 1141–1160, Nov. 2018.
- [18] S. LLC, "Senvol database," *Acesado: 17/02/2018*, 2018. [Online]. Available: <http://senvol.com/database/>.
- [19] S. Negi, S. Dhiman, and R. K. Sharma, "Basics, Applications and Future of Additive Manufacturing Technologies: a Review," *J. Manuf. Technol. Res.*, vol. 5, no. 1–2, pp. 75–96, 2016.
- [20] I. / ASTM52900-15, "Standard Terminology for Additive Manufacturing – General Principles – Terminology," *ASTM Int.*, 2015.
- [21] S. S. Crump and I. Stratasys, "APPARATUS AND METHOD FOR CREATING THREE-DIMENSIONAL OBJECTS," no. 19, 1989.

- [22] K. S. Boparai, R. Singh, and H. Singh, "Development of rapid tooling using fused deposition modeling: a review," *Rapid Prototyp. J.*, vol. 22, no. 2, pp. 281–299, 2016.
- [23] O. A. Mohamed, S. H. Masood, and J. L. Bhowmik, "Optimization of fused deposition modeling process parameters: a review of current research and future prospects," *Adv. Manuf.*, vol. 3, no. 1, pp. 42–53, 2015.
- [24] S. Kondoh, T. Tateno, Y. Kishita, H. Komoto, and S. Fukushige, "The {Potential} of {Additive} {Manufacturing} {Technology} for {Realizing} a {Sustainable} {Society}," in *Sustainability {Through} {Innovation} in {Product} {Life} {Cycle} {Design}*, M. Matsumoto, K. Masui, S. Fukushige, and S. Kondoh, Eds. Springer Singapore, 2017, pp. 475–486.
- [25] E. Rodriguez, A. J. Alvares, and C. I. Jaimes, "Conceptual design and dimensional optimization of the linear delta robot with single legs for additive manufacturing," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, p. 095965181983691, Mar. 2019.
- [26] E. de J. Cortés Torres, E. Rodriguez, C. I. Riano Jaimes, R. Bonnard, and C. A. Peña Cortés, "Enfoque sobre o Desenvolvimento de um Robô com Arquitetura Paralela 5R para Manufatura Aditiva," in *Anais do IX Congresso Brasileiro de Engenharia de Fabricação*, 2017.
- [27] J. P. Rodriguez, C. Pena, and E. Rodriguez, "Atlas_GcodeSender: a Graphical User Interface for Control of a Robotics Additive Manufacturing Platform," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2018, pp. 671–676.
- [28] S. Atluru, S. H. Huang, and J. P. Snyder, "A smart machine supervisory system framework," *Int. J. Adv. Manuf. Technol.*, vol. 58, no. 5–8, pp. 563–572, Jan. 2012.
- [29] B. Edrington, B. Zhao, A. Hansel, M. Mori, and M. Fujishima, "Machine Monitoring System Based on MTConnect Technology," *Procedia CIRP*, vol. 22, pp. 92–97, 2014.
- [30] S. Chen, C. Yin, and X. Li, "Implementation of MTConnect in Machine Monitoring System for CNCs," in *2017 5th International Conference on Enterprise Systems (ES)*, 2017, pp. 70–75.
- [31] N. Chen, Y. Yin, and Z. Yin, "Energy consumption monitoring system design of workshop processing equipment based on MTConnect," *Proc. - 2017 2nd Int. Conf. Inf. Syst. Eng. ICISE 2017*, vol. 2017–Janua, pp. 64–68, 2017.
- [32] Y. Xiao, Q. Liu, J. Wang, P. Sun, and Q. Liu, "Design and application of energy consumption acquisition system based on MtConnect," *Proc. 2018 IEEE Int. Conf. Mechatronics Autom. ICMA 2018*, pp. 1376–1381, 2018.
- [33] C. Deng, R. Guo, C. Liu, R. Y. Zhong, and X. Xu, "Data cleansing for energy-saving: a case of Cyber-Physical Machine Tools health monitoring system," *Int. J. Prod. Res.*, vol. 56, no. 1–2, pp. 1000–1015, 2018.
- [34] ISO10303-238, "Industrial automation systems and integration -- Product data representation and exchange -- Part 238: Application protocol: Application interpreted model for computerized numerical controllers," 2007.
- [35] ISO 14649-1, "Industrial automation systems and integration - Physical device control - Data model for computerized numerical controllers - Part 1: Overview and fundamental principles," 2003.
- [36] S. J. Shin, J. Woo, D. B. Kim, S. Kumaraguru, and S. Rachuri, "Developing a virtual machining model to generate MTConnect machine-monitoring data from STEP-NC," *Int. J. Prod. Res.*, vol. 54, no. 15, pp. 4487–4505, 2016.
- [37] D. Lechevalier, S. J. Shin, S. Rachuri, S. Fofou, Y. T. Lee, and A. Bouras, "Simulating a virtual machining model in an agent-based model for advanced analytics," *J. Intell. Manuf.*, vol. 30, no. 4, pp. 1937–1955, 2019.
- [38] S. Tools, "Digital Thread for Manufacturing," Available from <https://www.steptools.com/sln/thread/>, vol. Consulted:, 2018.
- [39] A. José Álvares, L. E. S. de Oliveira, and J. C. E. Ferreira, "Development of a Cyber-

- Physical framework for monitoring and teleoperation of a CNC lathe based on MTconnect and OPC protocols,” *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 11, pp. 1049–1066, 2018.
- [40] Mtc. Institute and Makerbot, “makerbot_agent,” Available on: https://github.com/mtconnect/makerbot_agent, 2015.
- [41] X. F. Liu and M. Cheng, “IMPLEMENTATION OF MTCONNECT FOR OPEN SOURCE 3D PRINTERS IN,” in *Proceedings of the ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE 2016*, 2017, pp. 1–10.
- [42] X. F. Liu, M. R. Shahriar, S. M. N. Al Sunny, M. C. Leu, and L. Hu, “Cyber-physical manufacturing cloud: Architecture, virtualization, communication, and testbed,” *J. Manuf. Syst.*, vol. 43, pp. 352–364, 2017.
- [43] L. Hu *et al.*, “Modeling of Cloud-Based Digital Twins for Smart Manufacturing with MT Connect,” *Procedia Manuf.*, vol. 26, pp. 1193–1203, 2018.
- [44] Stratasys, “Stratasys’ Introduces New F900 Production 3D Printer with MTConnect Interface, Last access 14/12/2018,” 2018. .
- [45] Y. Akao, *Quality Function Deployment: Integrating Customer Requirements Into Product Design*, 1st ed. Taylor & Francis, 2004.
- [46] C. G. Sørensen, R. N. Jørgensen, J. Maagaard, K. K. Bertelsen, L. Dalgaard, and M. Nørreemark, “Conceptual and user-centric design guidelines for a plant nursing robot,” *Biosyst. Eng.*, vol. 105, no. 1, pp. 119–129, Jan. 2010.
- [47] S. Pertuz, C. Pena, and C. Riano, “RoboTender,” in *2014 III International Congress of Engineering Mechatronics and Automation (CIIMA)*, 2014, pp. 1–5.
- [48] X. F. Liu, “Software quality function deployment,” *IEEE Potentials*, vol. 19, no. 5, pp. 14–16, 2001.
- [49] Z. Sener and E. E. Karsak, “A decision model for setting target levels in software quality function deployment to respond to rapidly changing customer needs,” *Concurr. Eng.*, vol. 20, no. 1, pp. 19–29, Mar. 2012.
- [50] L. Scalvenzi and P. A. Cauchick Miguel, “QFD aplicado ao desenvolvimento de software: priorização de requisitos do cliente em uma matriz de funções,” *Exacta*, vol. 14, no. 4, pp. 677–692, Dec. 2016.

ANEXOS.

ANEXO A. Archivo PrusaMendel.xml

Este anexo contiene la estructura para el archivo PrusaMendel.xml necesario para el funcionamiento correctamente del agente. Este archivo representa la estructura de datos que estarán disponibles para el cliente web a través del agente.

```
<?xml version="1.0" encoding="UTF-8"?>
<MTConnectDevices xmlns="urn:mtconnect.org:MTConnectDevices:1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:mtconnect.org:MTConnectDevices:1.1
http://www.mtconnect.org/schemas/MTConnectDevices_1.1.xsd">
<Header bufferSize="130000" instanceId="1" creationTime="2014-
11-06T19:21:45" sender="local" version="1.1"/>
<Devices>
  <Device id="PrusaMendel" iso841Class="6"
name="PrusaMendel" sampleRate="10" uuid="PrusaMendel001">
    <Description manufacturer="RepRap">Prusa
Mendel</Description>
    <DataItems>
      <DataItem category="EVENT"
id="PrusaMendelavail" type="AVAILABILITY"/>
    </DataItems>
    <Components>
      <Axes id="PrusaMendel-axes" name="Axes">
        <Components>
          <Linear id="PrusaMendel-X"
name="X">
            <DataItems>
              <DataItem
category="SAMPLE" id="PrusaMendel-x" name="Xact"
nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION"
units="MILLIMETER"/>
            </DataItems>
          </Linear>
          <Linear id="PrusaMendel-Y"
name="Y">
            <DataItems>
              <DataItem
category="SAMPLE" id="PrusaMendel-y" name="Yact"
nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION"
units="MILLIMETER"/>
            </DataItems>
          </Linear>
          <Linear id="PrusaMendel-Z"
name="Z">
```

```

                                <DataItems>
                                    <DataItem
category="SAMPLE" id="PrusaMendel-z" name="Zact"
nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION"
units="MILLIMETER"/>
                                </DataItems>
                                </Linear>
                                <Linear id="PrusaMendel-E"
name="E">
                                    <DataItems>
                                        <DataItem
category="SAMPLE" id="PrusaMendel-e" name="Extruder"
nativeUnits="MILLIMETER" subType="ACTUAL" type="POSITION"
units="MILLIMETER"/>
                                    </DataItems>
                                </Linear>
                                </Components>
                                </Axes>
                                <Sensor id="extruderSensor"
name="ExtruderSensor">
                                    <DataItems>
                                        <DataItem category="SAMPLE"
id="extruderTemp" name="extruder temp sensor"
type="TEMPERATURE"/>
                                    </DataItems>
                                </Sensor>
                                <Sensor id="bedSensor" name="BedSensor">
                                    <DataItems>
                                        <DataItem category="SAMPLE"
id="bedTemp" name="bed temp sensor" type="TEMPERATURE"/>
                                    </DataItems>
                                </Sensor>
                                <Controller id="ramps1.4" name="Ramps1.4">
                                    <DataItems>
                                        <DataItem category="EVENT"
id="###-progress" name="build progress" type="EXECUTION"/>
                                    </DataItems>
                                </Controller>
                                </Components>
                                </Device>
                                </Devices>
                                </MTConnectDevices>

```

ANEXO B. Archivo Agent.cfg

Este archivo contiene la configuración para la ejecución del agente en el con los parámetros de nivel superior.

```
ServiceName = MTConnectAgent
Devices = PrusaMendel.xml
Port = 5000
CheckpointFrequency=5000
AllowPut=true
Adapters
{
    PrusaMendel
    {
        Host=192.168.10.73
        Port=7878
        DeviceType=Faunc - CNC4axis
        FilterDuplicates=yes
        ReconnectInterval=5
        RealTime=true
    }
}
# Logger Configuration
logger_config
{
    logging_level = debug
    output = cout
}
```

ANEXO C. Artículo para publicación

Este anexo contiene una imagen con la primera página del artículo que será publicado en la revista IFAC PapersOnline desarrollado a partir de este trabajo.

Developing a MTConnect Framework for RepRap Additive Manufacturing Machines

Efrain Rodriguez* Juan Pablo Rodriguez**
Alberto J. Alvares* Cristhian Riaño*
Luiz Eduardo de Oliveira*

* Department of Mechanical and Mechatronics Engineering, University of Brasilia, Brasilia, Brazil, (e-mail: efrainrj2009@gmail.com, alvares@AlvaresTech.com, cristhianivanrj@gmail.com, luiz.oliveira@unidesc.edu.br)

** Department of Mechanical, Mechatronics and Industrial Engineering, University of Pamplona, Pamplona, Colombia, (e-mail: pablo.rodriguez@unipamplona.edu.co)

Abstract:

The MTConnect standard provides a bridge to bring networked physical manufacturing resources to cloud manufacturing environments. However, making legacy manufacturing equipment compliant with this new standard is still a difficult task. Regarding additive manufacturing machines, this issue has been poorly studied. To address the subject, this work introduces an MTConnect-compliant framework for shopfloor data access and monitoring of RepRap additive manufacturing machines based on Arduino open technology controllers. A communication channel based on TCP/IP protocol using an Ethernet module was incorporated into the machine's system to extend the functionalities of RepRap 3D printers to connect to the Internet. Two MTConnect implementation architectures named Type 1 and Type 2 were proposed. A web-client application was also developed to validate the system by performing real-time machine monitoring over the Internet. The proposed MTConnect solution allows retrieving data of axes position, heat-bed temperature, hot-end temperature, material extrusion, current layer number and elapsed time from the web application, demonstrating feasibility to operate in cloud manufacturing environments. Copyright © 2019 IFAC.

Keywords: Cloud Manufacturing, MTConnect, Cyber-Physical Systems, Additive Manufacturing

1. INTRODUCTION

The advent of Industry 4.0 is shifting the traditional business models of the manufacturer companies, from production-oriented to service-oriented manufacturing and from mass production to mass customization of products, in order to make them more agile and cost-effective to stay competitive in the global market. Part of this new industrial movement is becoming a reality thanks to the disproportionate growth of technological trends such as cloud manufacturing and additive manufacturing.

The core of cloud manufacturing is, essentially, in the integration of advanced manufacturing systems and cloud computing technologies to enable a pool of networked manufacturing resources and capabilities to be virtualized, encapsulated and provisioned to consumers in a loose-coupled network of on-demand manufacturing services over the Internet (Lu and Xu (2019)). By using cloud manufacturing, substantial enhancements are expected in terms of reducing the supply chain (Charro and Schaefer (2018)), a high performance of centralized management

* The authors would like to thank the financial support of the Brazilian government agency MEC/CAPE/CNPq/FAPDF.

and more sustainable production systems (Park and Jeong (2014)).

On the other hand, additive manufacturing is now trusted as the enabling technology with which companies can challenge the heterogeneous and ever-changing user's demands with greater efficiency and high-level of customization (Yao et al. (2017)). Because of its advantages for the creation of three-dimensional objects with complex geometries, multi-material and multicolored, minimizing the waste of material and without needing expensive tools, additive manufacturing becomes convenient for the production of customized products in small batches. Moreover, complex assemblies can now be achieved in a single 3D printed part, reducing manufacturing time and complexity.

Combining additive manufacturing with cloud-based services enables new possibilities to achieve more flexible, intelligent and agile manufacturing systems for mass product customization and on-demand manufacturing services. This idea is gaining more and more enthusiasm from both industry and academy. Nonetheless, essential activities in a cloud manufacturing environment such as ubiquitous connectivity, real-time monitoring and operations control are still crucial issues to be solved (Lu and Xu (2019)). The

ANEXO D. Archivos del proyecto

El siguiente link contiene los anexos referentes a los archivos utilizados y modificados que se utilizaron en este trabajo. Los archivos se alojados en repositorio de GitHub:

<https://github.com/juangasca95/Adaptador-MTcoonect---Prusa-Mendel-i2>