

DESARROLLO DE UN CONTROLADOR DÉBILMENTE ACOPLADO PARA INCUBADORAS NEONATALES IMPLEMENTADO EN UWP

Acosta Cano de los Ríos José Eduardo, Franco Luna Fabián
Óscar Chávez López, Carlos Méndez Herrera
Tecnológico Nacional de México/I.T. Chihuahua
Laboratorio de Automatización Industrial
Ave. Tecnológico No. 2909, 31310, Chihuahua, Chih., México, (614)2026511
jacosta@itchihuahua.edu.mx, fabianfranl@hotmail.com

RESUMEN.

En el esquema de desarrollo de sistemas clásico los controladores son diseñados con un funcionamiento predefinido, en el cual desde un inicio se configuran para usar los sensores y actuadores que requerirán a lo largo de todo su periodo de vida, esto es conocido como acoplamiento fuerte. Lo que puede ser un problema en ciertos casos debido a la inherente limitación que esto presenta, ya que el sistema únicamente será capaz de utilizar los elementos para los que fue diseñado desde un inicio. El desarrollo de un controlador escalable, el cual permita incorporar funcionalidades futuras, además de reemplazar los componentes (sensores y actuadores) individuales sin modificaciones severas al resto del equipo otorgaría flexibilidad y reutilización mayor de los elementos que lo conforman. Esto permitiría el desarrollo de una incubadora neonatal la cual teniendo en mente las normas oficiales que establecen las características estructurales y funcionales para su implementación en centros de salud pretende incluir conocimiento de sistemas embebidos de otras áreas y ofrecer una forma diferente al desarrollo de este tipo de equipo donde su estado actual se puede describir como hermético, dado que solo el fabricante y solo él sabrá que elementos conforman al sistema y de esta manera ser capaz de realizar una operación de corrección o reemplazo de sus componentes.

Palabras Clave: Acoplamiento débil, Sistemas flexibles, OOP, ArquiTAM

ABSTRACT.

In the classic system development scheme the controllers are designed with a predefined operation, in which from the beginning they are configured to use the sensors and actuators that they will require throughout their life, this is known as strong coupling. Which can be a problem in certain cases due to the inherent limitation that this presents, since the system will only be able to use the elements for which it was designed from the beginning. The development of a scalable controller, which allows us to incorporate future functionalities, in addition to a replacement of the individual components (sensors and actuators) and not of the entire equipment would grant greater flexibility and reuse of the elements that comprise it. This would allow the development of a neonatal incubator which having in mind the official norms that establish the structural and functional characteristics for its implementation in health centers, intends to include knowledge of embedded systems of other areas and offer a different way to the development of this type of equipment where its current state can be described as airtight, since only the manufacturer and only he will know what elements conform to the system and thus be able to perform a correction or replacement of its components.

Keywords: Loose coupling, Flexible systems, OOP, ArquiTAM

1. INTRODUCCIÓN

La reutilización de código y hardware es una parte importante de cualquier sistema moderno, tanto en sistemas informáticos que manejen una gran cantidad de datos como en sistemas embebidos enfocados a aplicaciones específicas. Un ejemplo común de la posibilidad de conectar distintos dispositivos (diferentes fabricantes) a las computadoras personales y estos ser reconocidos automáticamente y listos para usar sin necesidad de que el usuario realice cambios en el propio código del sistema operativo. El paradigma orientado a objeto (POO) resulta un apoyo importante en el desarrollo de sistemas flexibles y modulares. El POO facilita el acoplamiento débil entre el sistema base y aplicaciones específicas. Además, otorga la posibilidad de reutilización de código: abstracción de particularidades, donde el sistema base no requiere conocer características específicas de los elementos para utilizarlos.

El presente trabajo tiene como objetivo la aplicación del concepto de acoplamiento débil dirigido al desarrollo de un sistema embebido como una alternativa al diseño clásico del control de incubadoras. Mediante un sistema genérico de control, el cual al no presentar ninguna configuración inicial dada al momento de su creación permita el agregar elementos al sistema de diferente naturaleza sin forzar al sistema a limitarse por propiedades como el tipo de conexión a manejar (I^2C , Serie, UART, SPI, CAN, Ethernet, etc.) o si hay que leer o escribir algún tipo de señal digital o analógica, si esta habrá que ser linealizada o está determinada por una ecuación polinómica, el que hacer con la información obtenida y que acciones de control tomar (implementando controles On/Off, PID, PD, Modos deslizantes, etc.).

Para lograr esto se pretende hacer uso del esquema de referencia ArquiTAM, en el cual se prescribe una forma de lograr de acoplamiento débil entre el sistema base (genérico) y el resto de los elementos (algoritmo de control, sensores y actuadores). Originalmente el esquema se orientó a sistemas de manufactura, específicamente al lanzador de órdenes y su integración con las estaciones de trabajo.

El presente documento se encuentra conformado por los siguientes puntos ordenados de manera consecutiva: Antecedentes, breve descripción del estado actual del equipo médico en las instituciones del país, en especial de incubadoras y el enfoque del acoplamiento débil en proyectos anteriormente realizados. Marco teórico, se definen los elementos principales de donde parte el fundamento teórico para el posterior desarrollo de la solución. Desarrollo, aplicación del fundamento teórico sobre la solución propuesta. Análisis de resultados, describe la relación entre la solución conceptual y el sistema creado, así como las conclusiones obtenidas.

2. ANTECEDENTES

Actualmente en la industria médica existen muchas compañías internacionales y locales que ofrecen soluciones de equipos, accesorios y repuestos, así como el servicio de mantenimiento preventivo y correctivo de los mismos. La necesidad de cumplir con la demanda de servicios del sector salud en la población, ha inducido la presencia de una amplia gama de proveedores y fabricantes de esta clase de equipos y refacciones. Incluso, debido a la competencia, muchos fabricantes no proporcionan ningún tipo de información de las características o especificaciones respecto a sus componentes tal como el tipo de sensor limitándose simplemente a mostrar con que equipo en específico funciona, y por lo general, será útil solamente para el modelo especificado. Lo mismo pasa al momento de requerir utilizar el controlador en otro modelo distinto, lo cual en la mayoría de los casos presenta problemas de compatibilidad. Tal situación genera otro tipo de complicaciones, por ejemplo, se puede dar el caso en el que el hospital cuente (como en la mayoría de las ocasiones) con incubadoras básicas la cual solo estén provistas con la capacidad de controlar la temperatura del ambiente en el cual se encuentra el recién nacido, y esto no sea suficiente para ayudar a las complicaciones de salud que pueda presentar el recién nacido, lo cual es más preocupante al pensar en que es un aparato que se mantendrá en funcionamiento algunos años y supone una inversión inicial considerable para la institución como para no poder incorporar otras funciones necesarias para atender necesidades más allá de las básicas que puede ofrecer un equipo de este tipo.

Es así que en los equipos médicos todos los componentes van acoplados fuertemente desde un inicio y requieren que las piezas de reemplazo sea la especificada por el fabricante y en caso de no conseguir exactamente tal pieza requerida, el propietario del equipo se verá en la necesidad de cambiar este equipo por uno nuevo, ya que como cualquier equipo del área, este no se puede permitir fallar o presentar un mal funcionamiento. Los modelos más comunes, por lo tanto, son aquellos que cumplen con las funcionalidades básicas para su uso en centros de salud. Esto implica que su controlador se enfoca en mantener la temperatura ideal para los recién nacidos, sin incorporar otro tipo de

funcionalidades, llegando a requerir el acoplamiento externo de otro tipo de equipos como lo puede ser el monitor de signos vitales para suplir las necesidades que requiere la incubadora.

En cuanto a los trabajos realizados en acoplamiento débil se refiere, estos se encuentran enfocados a sistemas informáticos para el manejo de pisos o líneas de producción [1], donde una de las estrategias consiste en el concepto de operación dirigida por modelo, donde el sistema hace uso de la interpretación de las características abstraídas en el modelo de particularidades como es el caso de la técnica iMRP, reportada en [2], la cual está enfocada a procesos de plantas de producción.

3. MARCO TEÓRICO

3.1 Incubadoras Neonatales

Los bebés prematuros son aquellos nacidos antes de las treinta y siete semanas de gestación [3], lo cual deriva en que ciertos órganos no se encuentren desarrollados completamente, así mismo su sistema inmune suele ser muy débil, lo cual puede llevar a ciertas complicaciones e incluso a la muerte del recién nacido si es que este se expone al mundo exterior de una manera no controlada. Por lo tanto, una incubadora se puede definir como un equipo de soporte de vida destinado a recién nacidos prematuros, con el objetivo de mantener las condiciones ambientales para completar su correcto desarrollo [4]. Generalmente entre las variables de control de una incubadora más comunes se encuentran: temperatura, humedad y oxígeno; y cuentan con interfaces de salida mediante las cuales es posible observar las condiciones de las variables de manera fácil de observar por parte del personal encargado del cuidado del infante.

3.2 Normas

Para el desarrollo del controlador es importante conocer las normas y requerimientos propios del equipo para garantizar así su correcto funcionamiento. La norma NOM-066-SSA1-1993 establece las especificaciones sanitarias de las incubadoras para los recién nacidos [5], dentro de las cuales algunas de las más importantes son:

- Tener una estructura que permita la fácil visibilidad del recién nacido.
- Tener un correcto aislamiento del ambiente, de modo que las condiciones exteriores no afecten al interior de la incubadora.
- Mantener una temperatura ideal de 37°C con una variación de no más de ± 0.3 °C.
- Proporcionar hasta 85% de humedad relativa con incrementos de hasta 5%.
- Manejo de concentración de oxígeno del 21 a 60% con incrementos de 1%.

3.3 Acoplamiento débil

El acoplamiento débil es aquella característica del sistema que facilita su reconfiguración ofreciendo la posibilidad de agregar/eliminar/modificar características no contempladas explícitamente en tiempo del diseño del sistema. Mediante la aplicación del concepto de acoplamiento débil es posible afirmar que si dos elementos del sistema están débilmente acoplados, los cambios realizados en uno tendrán poco o nulo efecto sobre otro [6].

Para tal efecto en el presente desarrollo se emplear el esquema de referencia ArquiTAM como estrategia base. En ArquiTAM se plantea el desarrollo de un sistema genérico a partir del cual se instancia el sistema de aplicación a un caso particular mediante la interpretación del modelo de particularidades creado a partir del sistema específico a controlar. Aunque el esquema de referencia fue originalmente planteado para desarrollo de sistemas informáticos de control de producción, en el presente trabajo se plantea su uso en el desarrollo de sistemas dinámicos de control.

ArquiTAM [7] propone tres capas en el acoplamiento, las cuales están conformadas por:

- *Objeto representante.* Clases de objetos que se mantienen (sensores y actuadores).
- *Objeto envolvente.* Aquel que aporta la funcionalidad requerida por cada elemento.
- *Objeto propulsor.* Encargado de la comunicación del elemento con el sistema informático.

Para lograr un bajo acoplamiento entre el sistema base y los elementos físicos, el Framework no considera en cuenta las características específicas de cada elemento, simplemente utilizando comandos genéricos es que el lanzador de ordenes puede comunicarse con dichas capas utilizando funciones como lo puede ser “Operar()” en donde el argumento contiene la dirección del elemento al que se dirige.

3.4 ArquiTAM

En ArquiTAM (acrónimo de arquitectura de referencia de taller automático de manufactura), se plantea el desarrollo de un marco de trabajo (Framework) en donde se tome en cuenta los conceptos y detalles de la solución desde el modelo base hasta la implementación de un modelo particular dividido en tres niveles de abstracción (Figura 3.4):

- *Arquitectura de referencia.* Requerimientos para el dominio y desarrollo de la solución.
- *Modelo de referencia.* Funciones y métodos para facilitar la instanciación del sistema para una aplicación particular.

- *Modelo particular.* Instancia creada a partir de las clases implementadas.

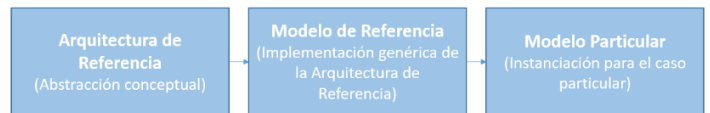


Figura 3.4 Etapas de abstracción ArquiTAM, [8].

3.5 PROGRAMACIÓN ORIENTADA A OBJETOS

La programación orientada a objetos es el paradigma de programación más utilizado en la actualidad [9]. Se puede referir a ella como un modelo que representa un subconjunto (abstracción) del mundo real, el cual cuenta con dos elementos: clases y objetos. Las clases son un prototipo de código el cual puede utilizarse para instanciar (reproducir) objetos iguales, se puede decir que es la estructura de un objeto y estas tienen atributos y métodos que definen a cada objeto en particular.

Si bien se podría seguir este paradigma en cualquier tipo de lenguaje de programación, existen algunos que facilitan su uso, entre ellos se encuentran:

- Objective-C
- C++
- C#
- Java
- Python
- Ruby
- Basic.NET
- ActionScript
- PHP

Los cuales presentan características como el manejo de memoria automática, la abstracción de datos a través del lenguaje, estructura modular en objetos tomando como base sus estructuras de datos, concepto de clases, herencia y polimorfismo que pueden ser manipuladas a través del lenguaje.

3.6 UWP y Windows 10 IoT

UWP (acrónimo de Universal Windows Platform) es la plataforma de Windows, la cual permite desarrollar aplicaciones para distintos dispositivos que utilicen el núcleo de dicho sistema operativo, de una manera flexible, y adaptativa. Es programable en C#, C++, Visual Basic y Javascript. Permite desarrollar la UI (interfaz de usuario) en XAML, HTML o DirectX.

Windows 10 IoT es una versión de Windows 10 orientada al “Internet of Things” enfocada a la creación de sistemas embebidos con estas capacidades. El ambiente cuenta con algunos requisitos como lo es el uso de una PC con Windows 10

para el desarrollo de sus aplicaciones, sin embargo, al seguir haciendo uso del núcleo de Windows permite desarrollar aplicaciones a través de herramientas como por ejemplo Visual Studio, tanto en C# como Visual Basic .Net, siendo el primer lenguaje el más utilizado para su programación. Si bien existe una versión “Enterprise” del sistema operativo, la versión “Core” está diseñada especialmente para el uso en CPUs ARM.

3.7 Sistema débilmente acoplado

Anteriormente se ha definido el acoplamiento débil y la utilidad de ArquiTAM para el diseño de este tipo de sistemas, siguiendo la línea de pensamiento, el POO nos permitirá crear una analogía entre el esquema de referencia y un sistema informático funcional. Tomando como punto de partida el modelo presentado por ArquiTAM se tendrían tres niveles de abstracción, los cuales se representan a través de clases, por lo que si se hace uso de sensores existirán tres clases: la clase representante la cual se mantiene sin cambio entre una operación y otra contando con las propiedades del sensor, la clase envoltura aporta las funciones y métodos; y la clase propulsor encargada de comunicarse con el elemento a través del medio físico. Siendo la misma estructura base para el caso del actuador.

Ahora bien, para el manejo de los recursos de la tarjeta, si bien existen varias opciones, se propone el uso de Windows IoT Core como sistema operativo lo cual permitirá utilizar la plataforma universal de Windows (UWP) y programar una aplicación que pueda ser transferible a otros dispositivos, así como programarse en diferentes lenguajes como C# y Basic. Teniendo la ventaja de ser estos lenguajes enfocados a la programación orientada a objetos, el cual ha probado anteriormente su eficacia al momento de solucionar problemas de este tipo, en comparación a las herramientas que nos facilita Raspbian para este tipo específico de programación, y la relativa escalabilidad que se puede presentar en este tipo de sistemas para dispositivos como lo es la Raspberry Pi.

4 DESARROLLO

El sistema informático se implementa con base en lo establecido en ArquiTAM como sistema genérico a partir de cual es posible instanciar el sistema particular deseado.

4.5 Implementación del modelo

El primer paso es analizar los requisitos del sistema para desarrollar una estructura base del controlador genérico, sus elementos y la conexión entre ellos. A manera de ilustración se presenta la prueba de un sistema de control de temperatura, donde el primer paso es listar los elementos necesarios, en este caso serán: el sensor para la medición temperatura, una resistencia calefactora como actuador que aumente la temperatura y el controlador (tarjeta) a utilizar para interpretar los valores leídos y aplicar algoritmo de control y generar la

señal de control a aplicar al sistema. En la figura 4.1 se muestra el diagrama de conexión de la estructura antes mencionada.

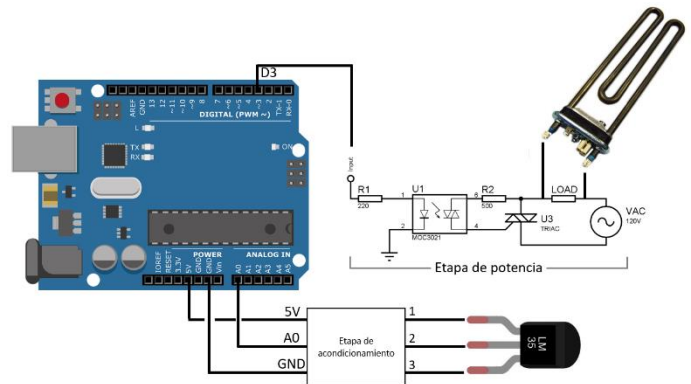


Figura 4.1 Diagrama de conexión

En tal sistema base, se puede observar como el sensor (LM35) envía la señal de medición (donde un grado centígrado equivale a 10mV) desde su pin dos a una etapa de acondicionamiento la cual se encarga de tomar el rango de temperatura más significativo para nuestro sistema y adaptarlo al voltaje de la tarjeta, leyendo dicho valor a través del pin analógico (en este caso A0), donde los niveles de lectura dependerán de la resolución del ADC de la tarjeta a utilizar en una función no lineal, por lo cual posteriormente se podrá realizar una linealización en el rango de la temperatura de interés. Una vez medido el valor, el trabajo de la tarjeta será la de convertir esta medición en unidades estándar como lo pueden ser los grados centígrados (°C), con lo cual se podrá controlar la temperatura según el valor de referencia seleccionado (en el caso de la incubadora el valor deseado siempre será de 37°C, lo cual será definido en el código interno de la tarjeta). La lógica de control implementada en el código de la tarjeta se podría representar con el pseudocódigo de la Tabla 4.1.

Tabla 4.1 Pseudocódigo controlador fuertemente acoplado

```

'Declaración de variables
Dim Temp As Float
Dim Setpoint As Integer = 37

Public Sub Setup()
    Dim AO As Input
    Dim DO As Output
End Sub

Public Sub Main()
    ReadTemp()
    UpdateVariables()
    ControlTemp()
End Sub

Public Sub ReadSensors()
    Temp = (5.0*analogRead[A0]*100)/1024
End Sub
    
```

```
Public Sub UpdateVariables()
    lblTemp.txt = Temp
End Sub

Public Sub ControlTemp()
    If (Temp>Setpoint)
        D0 = LOW
    Else If (Temp<Setpoint)
        D0 = HIGH
    End If
End Sub
```

```
Public Sub ControlTemp()
    If (Temp>Setpoint)
        S = LOW
    Else If (Temp<Setpoint)
        S = HIGH
    End If
End Sub
```

La clase sensor:

Tabla 4.3 Pseudocódigo Clase sensor (envoltura)

```
'Declaraciones
Private _ID As String
Private _TipoSeñal As String
Private _Polinomio As Integer
Private _Unidades As String

'Constructor
Public Sub New()

End Sub

Public Property ID as String
    Get;Set
End Property

Public Unidades ID as String
    Get;Set
End Property
```

Como se puede observar, todos los parámetros se encuentran definidos explícitamente desde el diseño, como lo son: elementos a utilizar, los pines de entrada y salida, los rangos de voltaje de las señales a medir/enviar, la ecuación que establece la relación entre los valores medidos con la temperatura en grados centígrados, la acción de control, etc. A pesar de ser un ejemplo de control bastante sencillo, como lo puede ser una acción de control On/Off, sirve para ejemplificar la manera común de realizar este tipo de controladores, donde sus elementos se encuentran fuertemente acoplados.

Si se quisiera ejemplificar un controlador de temperatura haciendo uso de este paradigma a través de pseudocódigo se tendría una clase para el sistema base, otra de envoltura para el tipo de objeto a instanciar el cual servirá de interfaz y las clases del tipo de objeto instanciado.

Empezamos con la primera clase, la clase genérica, la cual funcionara como lanzador de órdenes.

Tabla 4.2 Pseudocódigo clase representante

```
'Declaraciones
Dim oSensor As cSensor
Dim oActuador As cActuador
Dim ValorLeido, Temp as Integer = 0
Dim Setpoint As Integer = 37

Private Sub NuevoSensor_click()
    oSensor = New cSensor
    oSensor.ID = txtID.txt
    oSensor.TipoSeñal = txtTipoSeñal.txt
    oSensor.Polinomio = txtPolinomio.txt
    oSensor.Unidades = txtID.txt
End Sub

Public Sub Main()
    ReadSensors()
    UpdateVariables()
    ControlTemp()
End Sub

Public Sub ReadSensors()
    ValorLeido = sMaq
End Sub

Public Sub UpdateVariables()
    lblTemp.txt = ValorLeido
    Temp = ValorLeido
End Sub
```

Siendo la clase actuador, muy parecida con la diferencia del cambio de los atributos del objeto necesarios para el control de los actuadores a diferencia de los requeridos para la medición de los sensores.

En el caso de la clase propulsor, generalmente cuenta con su interfaz propia para el manejo de dicha iteración, más sin embargo, en el caso de la incubadora, en donde se contara con una interfaz unificada para observar y manipular todas las variables en la misma ventana, esta solo contara con las funciones necesarias para el funcionamiento del objeto.

Tabla 4.4 Pseudocódigo Clase propulsor

```
'Declaraciones
Private sID As String
Private sTipoSeñal As String
Private sPolinomio As Integer
Private sUnidades As String

'Constructor
Public Sub New()

End Sub

Public Property ID as String
    Get;Set
End Property

Public Unidades ID as String
    Get;Set
End Property
```

Por otra parte el acoplamiento débil a diferencia de lo anteriormente mostrado, básicamente no cuenta con ninguna de las características del sistema final, no sabe qué elementos o de que tipo contará, la forma en que se funciona la interfaz de conexión o la ecuación característica que lo define. Obteniendo así un sistema base, el cual se puede definir como una arquitectura de referencia, donde con respecto a la POO, existirán otras clases que funcionen como interfaz entre el sistema genérico y el de particularidades, este último siendo instancias de la clase de referencia que sirve para su envoltura según el objeto que se desee agregar. Es de esta manera, en la cual podrá crear los objetos sin tener que modificar el código base del controlador.

5 ANÁLISIS DE RESULTADOS

En la Figura 5.1 se muestra la abstracción de los elementos que componen el sistema. Donde se pueden apreciar tres secciones del sistema informático, siendo la del controlador, sensores y actuadores numeradas uno, dos y tres respectivamente y a pesar de mostrarse el Sensor y Actuador en estas áreas solo es indicativo de las secciones en las que tienen relación directa, mas no son parte del sistema informático *per se*.

El controlador (sección uno), es el sistema genérico (Framework) encargado de lanzar órdenes dirigidas tanto a los sensores y actuadores. Sin embargo hay que recordar que este no cuenta con ninguna de las características de estos, por lo cual solo se dirigirá a ellos con comandos sencillos como lo pueden ser leer, apagar, encender o escribir (ya sea algún valor numérico o cadena de caracteres), es así que al querer realizar una acción de control, se requerirá de una capa que especifique la acción a realizar (representadas por los modelos particulares cOn/Off, cPID y cPD en el diagrama).

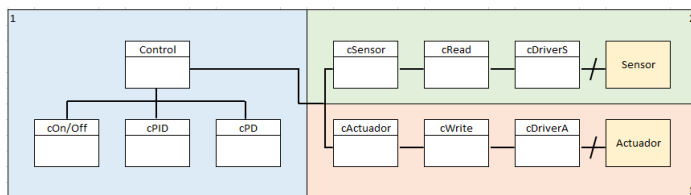


Figura 5.1 Secciones del sistema informático

Localizados en las sesiones dos y tres de la figura anteriormente mencionada y formando parte del marco de trabajo del acoplamiento débil se encuentran los objetos representante, envolvente y propulsor. Se puede observar que las áreas dos y tres comparten una estructura similar, en la cual la clase Control se comunica ya sea con las clases Sensor o Actuador las cuales son consideradas como el objeto representante, las cuales contarán con propiedades de clase según las características

inherentes a la naturaleza del elemento a manipular. A su vez este se comunica con el objeto envolvente en donde se implementan las funciones del elemento, representado por las clases Read y Write. Y a su vez estos se comunican con el objeto propulsor el cual se encarga de la conexión física con el sensor. De esta manera el lanzador de ordenes simplemente se encargara de enviar comandos genéricos hacia el sensor y actuador escogiendo como parámetro el objeto (instancia de elemento) creado según las necesidades del sistema, dicho objeto es creado al momento que el usuario decide crear una nueva instancia ya sea del tipo sensor o actuador y es almacenada en una colección de objetos. Es así que si se ve desde el enfoque jerárquico [10], el cada objeto desde el propulsor hasta el lanzador de órdenes va teniendo un nivel de abstracción que permite un acoplamiento débil en cada una de las etapas dada por el esquema de referencia ArquiTAM.

Si bien en la figura antes mostrada el sensor y actuador se encuentran dentro de las áreas dos y tres, hay que recordar que aparte de estos, las demás etapas son parte de un mismo sistema, el cual en cierto punto puede hacer uso de "Reflexión", permitiéndonos incorporar código en tiempo real sin que el usuario tenga que pre configurar las características de los nuevos elementos.

6 REFERENCIAS

- [1] Nakagawa, Elisa & Becker, Martin & Maldonado, José. (2013). Towards a process to design product line architectures based on reference architectures. 10.1145/2491627.2491651.
- [2] Gonzales H. Miguel A. (2017). APLICACIÓN DE LA TÉCNICA IMRP EN LA OPERACIÓN DIRIGIDA POR MODELO EN UN ÁREA DE PRODUCCIÓN POR LOTES. Tesis.F.I. (UACH)
- [3] Alistair G. S. Philip. (October, 2005). The Evolution of Neonatology. Pediatric Research, 58(4):799-815.
- [4] Baker, J. P. (2000). The Incubator and the Medical Discovery of the Premature Infant. Journal Of Perinatology, 20(5), 321.
- [5] S. d. s. d. g. d. l. E. U. Mexicanos, Norma oficial mexicana NOM-066-SSA1-1993, que establece las especificaciones sanitarias de las incubadoras para recién nacidos, México, D.F.: Normas oficiales mexicanas, 1994.
- [6] Ortegón S. Daniel A., Acosta. C. de los Ríos José E. (2018). SISTEMA GENERICO PARA MONITOREO Y CONTROL DE VARIABLES DE PROCESO BASADO EN ARQUITAM. ELECTRO, 40, 44-49.
- [7] Acosta, J.A., & Sastrón, F. (2006). Schematic Architecture / Frameworks / Particular Models for the Shop Floor Environment. IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics, 4563-4568.
- [8] Acosta. C. de los Ríos José E. (2015). Esquema de Referencia para Acoplamiento Débil entre Sistema Informático y Equipo de Producción. Tesis (Doctoral), E.T.S.I. Industriales (UPM)
- [9] Bruno Lopez T. (2016). CURSO DE PROGRAMACION ORIENTADA A OBJETOS EN C#.NET. EJEMPLOS CON APLICACIONES VISUALES Y DE CONSOLA. ALFAOMEGA.
- [10] Acosta-Cano, J., & Sastrón-Báguena, F. (2013). Loose Coupling Based Reference Scheme for Shop Floor-Control System/Production-Equipment Integration. Journal of applied research and technology, 11(3), 447-468.