



Universidad Politécnica de la Zona Metropolitana de Guadalajara

## **INGENIERÍA MECATRÓNICA**

Programación de sistemas embebidos

**EV\_1\_1\_seleccio\_de\_sistemas\_embebidos**

**NOMBRE DEL ALUMNO.-** Alejandro Almaraz Quintero

**Grado, Grupo y Turno.-** 8ºA T/M

**Matricula:** 17311336

**Docente.-** Carlos Enrique Moran Garabito.

Tlajomulco de Zuñiga, jal. A 31 de enero del 2020.

## **INTRODUCCIÓN**

A menudo los diseñadores de sistemas cometen el error de comenzar a diseñar e implementar soluciones que no han sido completamente especificadas y que corresponden a problemas a los que les falta delimitación, lo cual conduce a la construcción de sistemas que no satisfacen las necesidades de los clientes y que incurren en el aumento de los costos y en el incumplimiento de los plazos establecidos. Todo lo anterior refleja las carencias que existen en cuanto a la definición de requisitos como se describe en las estadísticas del Standish Group (Standish Group, 1994) (Standish Group, 2002) y en la literatura de Ingeniería de requisitos desde los trabajos pioneros de Ross (Ross y Schoman, 1977).

Esta problemática sigue existiendo tal como se ha mostrado en diferentes trabajos desde hace tres décadas (Zave y Jackson, 1997). En el campo de los sistemas embebidos tratados en Capel y Holgado (2004) y en (Ingham et al., 2006) dicha problemática general de los sistemas de información no solo se conserva sino que se agrava hasta el punto que el 60% de las componentes que integran hardware (HW) y software (SW) deben ser rediseñadas luego de haber sido programadas (Ganssle, 1999). La problemática expuesta para los sistemas en general tiene mayor impacto en el caso de los sistemas embebidos.

Dificultades adicionales en el caso de los sistemas embebidos surgen, entre otros, por los hechos siguientes: a) Estos sistemas se realizan generalmente por expertos en electrónica, pero alejados del uso de las metodologías de análisis y diseño de sistemas. b) La ausencia de metodologías específicas, que son reemplazadas por aquellas de propósito general que no contienen ninguna adaptación a los sistemas embebidos. c) Las metodologías existentes en el campo de los sistemas embebidos no cubren todas las fases de la IR. Son orientadas a la fase de diseño con poco apoyo de las fases previas del ciclo de vida de los sistemas.

## **DESARROLLO.**

Algunas metodologías orientadas a sistemas embebidos sólo permiten expresar los requisitos que se encuentran en el nivel de sistema, es decir, los que tienen que ver con funcionalidades, pero olvidan los requisitos que están en niveles externos, en donde se encuentran las interrelaciones con el súper-sistema y otros sistemas embebidos. Son metodologías muy orientadas al diseño, ignorando casi por completo la fase de captura y análisis de requisitos, como la metodología basada en el análisis de estados tratada en Ingham et al. (2006), la cual permite la separación temprana de HW y SW. El hardware se modela con

un diagrama de efectos de estado, y el software mediante la definición de metas y restricciones. La metodología CARA expuesta en Alur et al. (2004) también se ocupa del nivel del sistema y es concebida exclusivamente para sistemas médicos embebidos. En el proceso se transforman requisitos de diseño informales a lenguajes formales, como las máquinas de estado finito.

Otras metodologías para sistemas embebidos consideran, además del nivel del sistema, el contexto determinado por el súper-sistema y otros sistemas embebidos. En esta categoría están los trabajos de la metodología ECSAM (Lavi et al., 2005) centrados en el análisis de caja negra (detectar sólo interacciones entre el sistema y sistemas externos), lo cual permite que usuarios finales y diseñadores entiendan los requisitos capturados. Pertenece también a esta categoría la transformación de requisitos expresados en diversas notaciones fuente a un grafo conceptual realizado por Cyre (1997).

En los sistemas embebidos, los procesos de ingeniería de requisitos se complican a tal punto que en las áreas de aplicación típica más del 50% de los problemas se producen porque el sistema no cumple con las expectativas del usuario, debido a la captura errónea de los requisitos (Cheng et al., 2006). De esta manera surgen nuevas tareas de rediseño que conllevan aumento en los costos por la compra adicional de componentes hardware e incremento en los tiempos de entrega debido a la necesidad de desarrollar nuevo software de control (Kovitz, 2001) (Lavi y Kudish, 2005).

En consecuencia, el objetivo fundamental de la investigación asociada con este artículo es el de proponer una metodología de ingeniería de requisitos para sistemas embebidos que permita el análisis de requisitos y la generación de un modelo conceptual que facilite la entrada a un lenguaje de especificación de sistemas embebidos. En este trabajo se presenta el modelo de requisitos para el dominio de sistemas embebidos, el cual da cuenta del logro del primer objetivo específico de la mencionada investigación.

## **MATERIALES Y MÉTODOS**

En esta sección se presentan los conceptos básicos que permiten entender las secciones siguientes. El proyecto se enmarca dentro de dos grandes áreas: la ingeniería de requisitos (IR) y los sistemas embebidos (SE).

Un sistema embebido es un sistema de procesamiento de información de uso específico integrado en otro sistema de mayor tamaño y conformado por componentes hardware y software (Marwedel, 2003).

Los sistemas embebidos tienen algunas particularidades como la integración de componentes hardware y software (Marwedel, 2003), y su relación de jerarquía con un súper-sistema que se encarga de controlar la comunicación entre sistemas del mismo nivel.

En la fase de definición del sistema se obtiene el conocimiento de los contextos externos que expresan la problemática que se debe afrontar y fundamentan una solución. En la segunda fase se deberá obtener un documento con todos los requisitos. En la fase de operacionalización de requisitos se logra producir un documento detallado de funcionalidades y restricciones de bajo nivel de abstracción del sistema a construir; y en la cuarta fase se construye un modelo conceptual que contenga la solución acorde con los requisitos y restricciones.

Una metodología de IR pensada para este dominio debe permitir la representación de requisitos en el nivel de sistema, es decir, los que expresan funcionalidades, pero, además, debe plasmar requisitos que están en niveles externos, en donde se encuentran las interrelaciones con el súper-sistema y otros sistemas embebidos.

## **RESULTADOS.**

Esta sección se subdivide para mostrar inicialmente la forma en que se obtuvo el modelo de requisitos para sistemas embebidos, y luego pasar a un caso de aplicación.

### **Obtención del modelo de requisitos para sistemas embebidos**

Teniendo en cuenta la carencia que en general presentan las metodologías descritas en la revisión de la literatura, se propone intervenir la metodología ABC-Besoins, que fue diseñada para el dominio de sistemas web, adaptándola y transformándola para tener en cuenta aspectos del dominio de sistemas embebidos, y construir un modelo conceptual que facilite la entrada a un lenguaje de especificación como SystemC. La decisión de retomar esta y no otra metodología se fundamenta principalmente en que ABC-Besoins ofrece soporte para todas las fases del análisis de requisitos y posee un modelo de requisitos

bien estructurado que permite descubrir requisitos de los niveles de sistema y los niveles externos. En las sub-secciones siguientes se presenta el proceso y posterior resultado de adaptar el modelo de requisitos de la metodología seleccionada al dominio de sistemas embebidos.

Para lograr dicha adaptación fue necesario estudiar, primero, las características propias de los sistemas embebidos, las cuales se enuncian en la siguiente sub-sección.

### **A) Identificación de las características propias de los sistemas embebidos**

La característica más importante de los sistemas embebidos es su interacción con el mundo exterior en función del tiempo o en función de la presencia de estímulos. Para garantizar una interacción exitosa con el ambiente, el sistema debe incorporar algunas características, tales como la disponibilidad, fiabilidad y seguridad. Otras características propias de un sistema embebido son (Marwedel, 2003) (Lavi y Kudish 2005):

- Compuesto por hardware y software, con la característica de que el software tiene una interacción directa con los elementos hardware, pues se encarga de controlarlos y comunicarlos. Para esta composición debe ser posible representar: comportamiento (estados, eventos, y señales) y estructura o composición física.
- Relaciones jerárquicas, en las cuales se incluyen las interrelaciones entre el sistema embebido y su súper-sistema, el sistema embebido y sistemas del mismo nivel que se encargan de otras funciones específicas.
- Comportamiento basado en el estado de las componentes, por ejemplo, si X puerto no está disponible, entonces no se podrá enviar la señal que activa el proceso Y.
- Manejo de eventos, que son los que permiten constatar el cambio de estado de las componentes. Un evento puede ser externo (causado por el ambiente) o interno (causado por componentes del sistema).
- Recursos limitados en cuanto al tamaño, el consumo de energía, la memoria, y demás recursos que permitan garantizar la portabilidad del sistema embebido.
- Mínima interacción con el usuario, por lo tanto, son sistemas que deben funcionar durante años sin errores y ser capaces de recuperarse por sí mismos

en caso de que estos ocurran. Deben ser sistemas con un alto grado de autonomía.

- Presencia de sincronización y comunicación, para permitir el flujo de información entre los diferentes sistemas embebidos que hacen funciones específicas y contribuyen a la realización de la función del súper-sistema.
- Propiedades no funcionales, tales como la tolerancia a fallas, el tamaño, el consumo de potencia, el peso, la disponibilidad, la seguridad, la fiabilidad, deben definirse desde etapas tempranas de la construcción del sistema.
- La controlabilidad es otra característica de los sistemas embebidos, pues son sistemas pensados, en su mayoría, para el control, y, además, por sus restringidas y muy específicas funciones, también son fácilmente controlables.