

# Un Sistema de Tiempo Real Distribuido Semi-Soft usando RUP

**Julio Javier Castillo**  
**Marina Elizabeth Cardenas**

LIS-Laboratorio de Investigación de Software  
Universidad Tecnológica Nacional, Facultad Regional Córdoba  
Maestro M. López esq. Cruz Roja Argentina, Córdoba - Argentina

## Abstract

*En este trabajo se aplica y se adapta la metodología de desarrollo de software RUP (Rational Unified Process [Booch, Jacobson 1999]) para la construcción de un sistema distribuido de tiempo real semi-sof, time driven (distributed semi-soft real time system). El sistema en concreto es un control distribuido en tiempo real de un horno eléctrico. Dicho control se lleva a cabo en forma remota sobre el protocolo tcp/ip a través del empleo de un webservice. Para la interfaz electrónica de control se emplea microcontroladores y se utiliza Microsoft VisualC#.Net 2005 para el control desde una computadora PC. Como resultados de este trabajo podemos observar cómo emplear RUP para el diseño de sistemas en tiempo real y como interactúan embedding software – computer software – web software en una aplicación concreta.*

**Palabras claves:** RUP, distributed semi-soft real time systems, furnace control, PID-Control.

## Introducción:

Diariamente nos presentamos con problemas de mantenimiento de temperatura, tal es el caso de mantener dentro de un determinado intervalo la temperatura del vehículo que estamos conduciendo, o bien de mantener dentro del rango permitido la temperatura del microprocesador de nuestra PC.

El cuerpo humano es otro ejemplo de un sistema de control y regulación de la temperatura, el cual debe mantenerse entre 36.4°C. y 37.1°C. dependiendo de la actividad que se realice, pero cuando esta desciende o aumenta considerablemente es señal de alarma para el organismo.

El cuerpo mantiene la temperatura constante mediante un centro conocido como termorregulador, localizada en la parte del cerebro llamada hipotálamo, que constituye en “kernel” del sistema de control del cuerpo humano.

Generalmente, la temperatura suele aumentar ante una infección o enfermedad (noisy environment) y suele descender cuando hay algún traumatismo (noisy environment).

Como podemos observar, estamos rodeados de situaciones en donde el control de la temperatura es una variable vital que debe y tiene que ser controlada. Además, el prescindir de este control puede causar graves pérdidas materiales, percances económicos y hasta pérdida de vidas humanas, u otro tipo de daños.

Por otra parte, a nivel industrial, el control de la temperatura es una variable de gran relevancia, requiriendo generalmente de sistemas en tiempo real duros (hard real time systems) que realicen el control de estos procesos.

Por medio de este trabajo se pretende realizar un control de un horno industrial eléctrico en forma remota a través del diseño de un Semi-Soft Real Time System – Time Driven, empleando para ello el RUP (Rational Unified Process) como proceso de desarrollo de software y utilizando UML (Unified Modeling Language) como lenguaje de modelado.

Decimos que este sistema es Time-Driven ya que las acciones del sistema son dirigidas principalmente por el pasaje de un

intervalo de tiempo a otro. Este sistema ejecuta en forma periódica ciertas tareas, en base a la ocurrencia de “deadlines” en el tiempo.

RUP se ha empleado exitosamente para el modelado de sistemas de los más diversos dominios, sin embargo RUP presenta algunas limitaciones cuando se modela sistemas en tiempo real. Por ejemplo, en la fase de diseño se debe decir entre el “diseño de componentes en tiempo real” (generalmente empleando Rational Rose-RT) y “el diseño de componentes que no son de tiempo real” (generalmente empleando Rational Rose como herramienta de modelado), pero RUP no provee de guías de cómo realizar esta elección, ni expresa los beneficios y las consecuencias de estos dos métodos de diseño.

La idea inicial de este trabajo era la de realizar el control de un horno industrial a gas, sin embargo, debido a la imposibilidad de realizar pruebas sobre un sistema de éstas características, se tomó como prueba un horno eléctrico que se utiliza para la esterilización de materiales.

Debido a nuestro dominio de problemas, frecuentemente preferiremos el uso del inglés para las palabras técnicas.

El presente trabajo se divide en varias secciones : Metodología, Estado Actual del Proyecto, Bloques de Construcción, Desarrollo del Software de Control, Cadena de Adquisición, Conclusiones y Referencias.

## **Metodología**

Se ha decidido emplear RUP como metodología de diseño de desarrollo de software debido a su amplia aceptación en el ámbito industrial y académico, ya que brinda un marco de trabajo genérico que se puede adaptar para modelar sistemas en tiempo real distribuidos como es el caso de este proyecto.

Debido a la extensión máxima del paper, no será posible transcribir todos los artefactos que se obtienen al aplicar la

metodología RUP. Por ello, se han elegido los Criticals Uses Cases para representar las funcionalidades más importantes del ciclo de vida del desarrollo del sistema software.

Además se provee de una descripción en bloques funcionales: Bloque Físico, Bloque de Electrónica de Control y Bloque de Sistema Software de Control.

A continuación resumimos algunos de los alcances más importantes:

- Registrar la temperatura del horno durante un determinado intervalo de tiempo y mostrarla gráficamente.
- Permitir al usuario local/remoto: encender/apagar el horno cuando él lo desee, realizar una planificación del comportamiento del horno eléctrico, entre otras.
- Permitir la elección de un conjunto de usuarios (lista de usuarios) a los cuales se les informará por medio de un envío de e-mail, los cambios de temperatura que sean considerados críticos o alarmantes.
- Brindar al usuario la opción de elegir diferentes mecanismos de control de temperatura.

Como requisitos no funcionales podemos destacar:

- Informar a los usuarios de la “Lista de Alertas” acerca de las variaciones de temperatura fuera de los límites de control en un tiempo inferior a un minuto.

Este sistema ha sido desarrollado teniendo en cuenta los siguientes principios de Ingeniería del Software:

Adaptabilidad, Usabilidad, Reusabilidad, y Separación de Incumbencias.

## **Estado Actual del Proyecto**

Actualmente el sistema ha completado completamente la fase de inicio y la fase de elaboración, ubicándose en la segunda iteración de la fase de construcción. Es por ello que ya se ha

conseguido una línea base de la arquitectura y ya se han mitigado los riesgos más importantes que podrían hacer peligrar este trabajo o proyecto.

Se realizaron iteraciones a través de los 5 flujos de trabajos genéricos (workflows). En la fase de Inicio se realizaron dos iteraciones cuyo objetivo fue la investigación del state of the art de los “distributed soft/hard real time systems”, “intelligence controllers”, “distributed controllers”, entre otros. En esta etapa se identificaron y priorizaron los riesgos más importantes y se desarrolló un modelo preliminar de Uses Cases con los Critical Uses Cases, obteniendo de esta manera un esbozo de la arquitectura del sistema.

Por otra parte, la fase de elaboración constó de tres iteraciones en las que se especificaron con mayor nivel de detalle los use cases obtenidos en la fase de inicio y se diseñó la arquitectura del sistema. A su vez en la primera iteración de esta etapa se comenzó a implementar y a probar el embedded software y en la tercera iteración se comenzó a desarrollar la electrónica de control del horno.

A continuación se presenta un modelo de Acquisition Chain, que ilustra como se lleva a cabo el remote control de este semi-soft real time system.

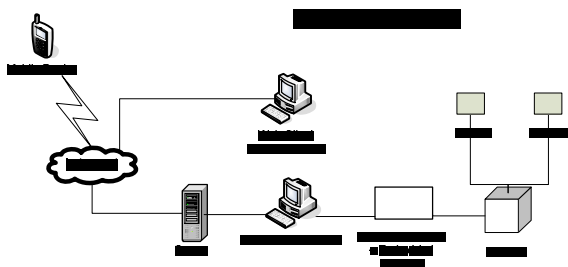


Figura 1. Cadena de Adquisición de alto nivel.

*Requeriments WorkFlow – Flujo de Trabajo de Requisitos :*

En este workflow se elaboró un context model (modelo de contexto) y un Modelo de Casos de Uso a Nivel de Análisis mediante el cual se ha identificado los actores de este Sistema en Tiempo Real

y los Use Case que proveen su funcionalidad.

A continuación se adjunta el diagrama de Casos de Uso de Análisis :

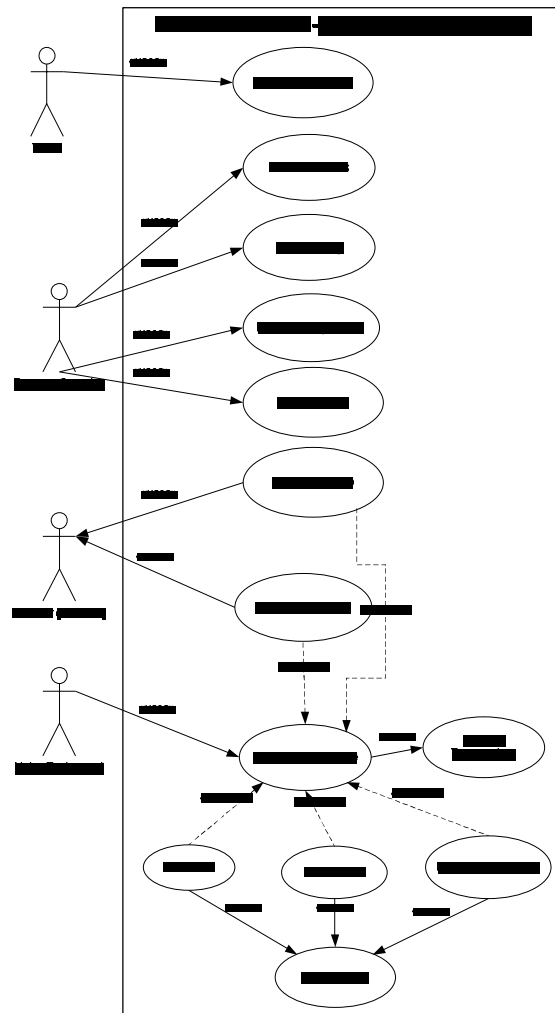


Figura 3. Modelo de Casos de Uso del Sistema – simplificado - mostrando los uses cases críticos.

Los nombres de estos uses cases son muy descriptivos, razón por la cual no se provee de un resumen.

**Bloques de Construcción:**

A continuación se enumeran y describen los bloques funcionales de este sistema en tiempo real distribuido.

**Ø Bloque Físico:**

Este bloque está constituido por un Horno Eléctrico empleado para la esterilización de materiales quirúrgicos, que

puede soportar temperaturas que oscilan entre los 0°C y 140°C, aunque en este sistema solo trabajaremos dentro del intervalo de 15°C a 94°C.

#### Ø **Bloque de Electrónica de Control :**

Este bloque esta conformado por una:

- Tarjeta de adquisición de datos, que permite la comunicación con la PC y es el medio a través del cual es posible obtener y controlar la temperatura del horno(no se enfria por medios electro-mecánicos).

El componente principal que permite la comunicación PC – Mundo Exterior es un Microcontrolador PIC 16F877A, el cual permite sensar la temperatura y activar un actuador para que modifique la temperatura del horno eléctrico.

#### Ø **Bloque de Software de Control del Sistema :**

Este bloque de control presenta 3 elementos constitutivos, cada uno con un nivel de abstracción diferente y que corresponden a 2 diferentes paradigmas de programación:

- Programa Pic 16F877A-(Paradigma Estructurado Lineal).
- Programa de Aplicación en C# (Paradigma Orientado a Objetos ).
- Programa WebAplicattion sobre un Web-Server –( Paradigma Orientado a Objetos con patrón de arquitectura Cliente-Servidor de 3 capas).

#### • **Cadena de Adquisición :**

A continuación se describen sus principales elementos (Véase figura 1):

- 1) Horno eléctrico de temperatura.
- 2) Conversor ADC de 10 bits
- 3) Interfaz USART – RS232.
- 4) PC
- 5) Web-Server, con Webservices
- 6) Web-Client / Remote Control

La temperatura es sensada a través de un sensor de temperatura LM35, el cual entrega un valor de tensión que es

proporcional a la temperatura medida en el horno eléctrico.

Este valor es tomando por el PIC y pasado a la PC a través del puerto serie COM1 empleando el protocolo RS232.

Si la temperatura se encuentra fuera de los límites mínimos/máximos establecidos, entonces se envía un e-mail de alerta a una lista de usuarios.

Además, es posible visualizar gráficamente el historial de variación de temperatura de las últimas k-mediciones efectuadas.

Por otra parte, la PC permite realizar dos tipos de controles :

-control del tipo SI/NO y control PID(Proporcional Integral Derivativo).

#### • **Desarrollo del Software de Control:**

El sistema software de control se desarrollo en assembler en el caso del PIC, en C# en el caso de la Windows Application, y en ASP.Net y C# en el caso de la Web Application y Webservice.

Se han integrado diferentes lenguajes de programación para poder alcanzar el objetivo inicialmente planteado.

El software de la Windows Application será la encargada de mostrar la temperatura actual del horno, y su historial de variación. Se muestra en forma gráfica para lograr una comprensión más fácil por parte del usuario.

Finalmente, el software de la Web Application será el encargado de enviar el mensaje de alerta hacia una lista de distribución previamente definida, y de brindar una interfaz a través de un webservice, a un operador del horno que puede estar localizado a cualquier distancia del dispositivo controlado.

#### **Conclusiones**

Se ha mostrado una aplicación exitosa y funcional de RUP para el modelado de un sistema distribuido semi-soft de tiempo real, en el cual se han

empleado conocimientos de distintas áreas, tales como ingeniería del software, electrónica, y ciencias de la computación.

En cuanto a su fiabilidad, este sistema a pequeña escala ha sido probado múltiples veces, entre ellas, se lo probó durante 48 horas seguidas, introduciendo variaciones controladas en el ambiente como ventilación o añadiendo más temperatura, y durante este tiempo no se ha evidenciado ningún error ni desfasaje mayor a 1,5°C grados centígrados respecto de la temperatura deseada, esta prueba se realizó a 67°C. Se realizaron otras pruebas a 50°C, 60°C, 70°C, 80°C y 90°C, y en todos los casos se obtuvieron resultados similares.

Este sistema esta sujeto a muchas mejoras y ampliaciones como por ejemplo la aplicación de técnicas de tuning pid, fuzzy logic, entre otras. Estas técnicas se estudiarán y se aplicarán en el presente trabajo en un futuro cercano.

El Sistema de Control Automático de Temperatura del Horno Eléctrico diseñado presenta características novedosas que lo convierten en un sistema de tiempo de real blando, cuyas variación de temperatura fuera de los límites parametrizados y previamente establecidos, permite a un usuario ubicado en cualquier parte del mundo enterarse de tal evento con una demora en el caso promedio de unos 50 segundos.

Por otra parte este sistema es susceptible de grandes mejoras de funcionalidad y performance, entre tales mejoras citaremos la generalización a un control de k-dispositivos simultáneos, o la adición de un mecanismo de enfriamiento de la temperatura para realizar de una manera más versátil el control de la temperatura.

Además sería posible trabajar con hornos de mayor temperatura y aplicarles este mismo sistema de control con mínimos cambios.

Este trabajo tuvo como principal motivación la necesidad de integrar diferentes tecnologías en pos de construir un sistema de tiempo real de control de

temperatura empleando RUP y que sea capaz de informar a un conjunto de personas responsables los momentos en que el sistema esté fuera de control o bien cuando se haya llevado el sistema fuera de los límites preestablecidos.

## Referencias

- q *Jacobson, I ; Booch, G. y Rumbaugh J. El Proceso de Desarrollo de Software.*
- q *Jacobson, I ; Booch, G. y Rumbaugh J .UML , El Lenguaje Unificado de Modelado.*
- q *BOOCH, G. (1996) Object Oriented Design with Applications. 2ª ed. Wilmington, Addison-Wesley Iberoamericana.*
- q *JACOBSON, I. y otros. (1994) Object-Oriented Software Engineering. A Use Case Driven Approach. Estados Unidos de América, Addison-Wesley/ACM press.*
- q *RUMBAUGH, J. y otros. (1991) Object-Oriented Modeling and Design. Englewood Cliffs, Prentice-Hall.*
- q *MARTIN FOWLER, KENDALL SCOTT. (1999) UML Gota a Gota, Addison Wesley.*
- q *Jean-Pierre Courriou, Process Control: Theory and Applications , Springer*
- q *Rich, E. y Knight, K. (1995) Inteligencia Artificial. McGraw Hill.*
- q *OMG – [www.omg.org](http://www.omg.org) – artículos varios.*
- q *Papers varios.*

