

# Ingeniería de Tiempo Real en el Diseño de un Sistema de Supervisión y Control para Redes de Energía Eléctrica

Huáscar D. Espinoza Ortiz, Jorge A. Nava Amador

CPGISC - Coordinación de Postgrado: Ingeniería de Sistemas de Control

Universidad Mayor de San Andrés

La Paz Bolivia

e-mail: [cpgisc@huayna.umsa.edu.bo](mailto:cpgisc@huayna.umsa.edu.bo)

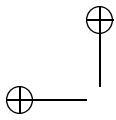
## Resumen

Este trabajo aborda la problemática de integración de los requisitos de tiempo real en el diseño de un sistema de supervisión y control (*SCADA*) aplicado al sector eléctrico. El esfuerzo subyacente se ha orientado a la búsqueda y aplicación de las técnicas más apropiadas de la ingeniería de tiempo real, que permitan predecir matemáticamente la respuesta del sistema ante diferentes situaciones de carga de procesamiento durante los estados de actividad crítica. Como resultados substanciales se han obtenido: la especialización de un proceso de desarrollo genérico, RUP (*Rational Unified Process*), enfocándolo a sistemas de tiempo real; una arquitectura SCADA abierta soportada en estándares IEEE e IEC y modelada con UML (*Unified Modeling Language*); y un conjunto de recomendaciones de implementación, obtenidas a partir del análisis del comportamiento en tiempo real con RMA (*Rate Monotonic Analysis*) y la plataforma MAST (*Modeling and Analysis Suite for real-Time applications*).

**Palabras Clave:** Sistemas de tiempo real, SCADA, supervisión y control de redes eléctricas, modelado orientado a objetos, planificación de tareas, arquitecturas abiertas y distribuidas de sistemas.

## 1. Introducción

Los sistemas de supervisión y control de redes eléctricas son en su mayoría distribuidos, exigiendo fiabilidad, seguridad y tiempos rigurosos de ejecución impuestos por el entorno de aplicación. La característica que hace a estos sistemas más complejos que cualquier otro sistema de tiempo real es que además deben manejar cantidades importantes de información, atendiendo peticiones de otras aplicaciones utilizadas por los agentes de operación del sector eléctrico.



Si bien en condiciones normales de la red eléctrica, todos los elementos SCADA (RTUs, enlaces de comunicación, computadoras frontales y procesadores principales), presentan una respuesta adecuada hacia el entorno y los operadores, generalmente en armonía con la capacidad proporcionada al sistema durante su diseño; no es sino hasta que ocurren condiciones de actividad alta no planeada (por ejemplo gran cantidad de eventos causados por perturbaciones eléctricas simultáneas) o cuando se incluyen elementos adicionales de consumo de los servicios SCADA (aplicaciones de gestión EMS y DMS entre otras), que el sistema puede degradar su respuesta hasta producir fallos tales como maniobras automáticas de elementos eléctricos no válidas en el tiempo, o retrasos considerables en la actualización de la información hacia los operadores que no permiten actuar oportunamente [4].

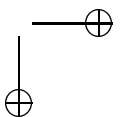
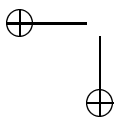
En este contexto, actualmente no es suficiente contar con criterios de rendimiento y pruebas robustas de fábrica (FAT) y de sitio (SAT) -que tradicionalmente se aplican para garantizar el comportamiento del sistema- sino que también se hace necesario utilizar técnicas formales que permitan predecir analíticamente la respuesta de todos los elementos SCADA ante las situaciones de *peor caso* posible, con la consecuente posibilidad de planificar predeterminadamente los recursos hardware y software del sistema.

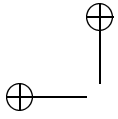
En este último campo, denominado *Ingeniería de Tiempo Real*, uno de los esfuerzos más importantes es el *Análisis de Ritmo Monotónico* (RMA), que agrupa un conjunto de técnicas aplicables a cualquier tipo de sistema de tiempo real (monoprocesador, multiprocesador, distribuidos, con exigencias estrictas y no estrictas). Su uso proporciona una base científica probada para estudiar y garantizar la ejecución de cada una de las tareas computacionales de un sistema en tiempos acotados impuestos por el entorno con el que interactúa, asignando los recursos hardware y software de una forma eficiente [8].

En los últimos años, el desarrollo de la ingeniería de tiempo real está intentando integrarse con la fuerte corriente de arquitecturas orientadas a objetos, de tal forma que se integren las cuestiones funcionales de los sistemas, con el comportamiento en tiempo real de los procesos. Un referente importante al respecto es el establecimiento de un perfil de comportamiento de tiempo real orientado a objetos, denominado *Profile for Schedulability, Performance and Time*. A partir del mismo se podrán desarrollar herramientas de modelado orientado a objetos de sistemas de tiempo real que estén enlazadas de una manera estándar con las funciones de análisis de planificabilidad existentes [7].

Queda por tanto, un reto importante en el campo de SCADAs para el sector eléctrico, que consiste en establecer guías universales para la aplicación de la ingeniería de tiempo real en el desarrollo de estos sistemas, definiendo puntualmente las exigencias de respuesta de tiempo real para cada uno de los procesos particulares involucrados en la supervisión y control de las redes eléctricas, para lo cual se requiere una comprensión integrada del problema.

El trabajo aquí descrito constituye un esfuerzo en este sentido, llevado a cabo por la *Unidad de Postgrado en Ingeniería de Sistemas de Control* (CPGISC) de la *Universidad Mayor de San Andrés* (Bolivia), como una extensión de la investigación del Grupo de Computadores y Tiempo Real (CTR) de la *Universidad de Cantabria* (España) en las





áreas de sistemas de tiempo real y supervisión y control de redes eléctricas; habiéndose obtenido entre los resultados más valiosos la instrumentación para la captación de datos y control de subestaciones (UIIMPC) [3], la plataforma de recursos de alto nivel para la gestión de la red eléctrica [5], y una plataforma para modelar y analizar aplicaciones de tiempo real denominada MAST [2].

El enfoque metodológico asumido en este ámbito específico de aplicación, pretende hacerse extensivo hacia otras áreas afines de los sistemas de control.

## 2. Planteamiento del Problema

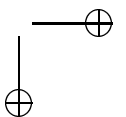
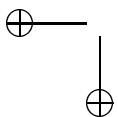
Se ha descompuesto el problema en tres aspectos específicos, que merecen soluciones puntuales (Figura 1):

1. **Fallos en plazos temporales:** Al no tomar en cuenta los aspectos de tiempo real en ninguna fase de desarrollo del sistema, se pueden presentar fallos no previstos en su funcionamiento, debido a que no existe garantía analítica del cumplimiento de las demandas temporales y resulta complicado identificar el origen de los problemas dentro del sistema. Estos fallos pueden deberse a bloqueos entre las tareas que se ejecutan dentro de uno o varios procesadores o a una sobrecarga de tareas asignadas a un mismo procesador.
2. **Altos costos de actualización:** Cuando se toma en cuenta el aspecto de tiempo real después de diseñado el sistema, el costo de adaptación, dentro de la arquitectura ya concebida, resulta elevado, debido a que los requerimientos de tiempo pueden exigir cambios en gran parte de la estructura software o requerir de recursos hardware diferentes. Por otro lado, el mantenimiento del sistema se hace más problemático, ya que no se cuenta con una infraestructura formal dentro del sistema que permita una actualización planificada.
3. **Inexistencia de plazos referenciales:** Aunque existen grupos de investigación dedicándose a este tema, las técnicas utilizadas no se han abordado cuidadosamente en sistemas SCADA de redes eléctricas, donde se requiere un análisis más profundo del dominio del problema, a fin de evaluar con precisión los plazos de respuesta impuestos por el proceso y los usuarios.

### 2.1. Alcance Propuesto

Se han planteado los siguientes objetivos específicos para la elaboración del trabajo:

1. *Evaluar y documentar los requerimientos* tanto del proceso de distribución de energía eléctrica como de los usuarios del sistema computacional de supervisión y control, especificando formalmente las exigencias de tiempo real determinadas por el entorno de aplicación.



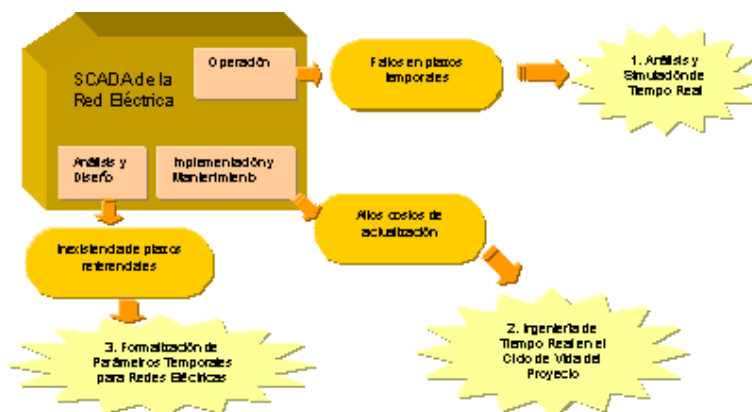


Figura 1: Formulación del problema y orientación de la solución.

2. *Abordar el análisis y diseño* de un sistema genérico de supervisión y control de una red de distribución de energía eléctrica, a través de una metodología que incluya los requerimientos de tiempo en todas sus etapas: concepción y modelado, análisis y diseño de la arquitectura hardware y software, y representación del comportamiento de los procesos del sistema.
3. *Estudiar la respuesta de tiempo real* del sistema ante un rango amplio de condiciones y ocurrencias de eventos externos e internos al sistema, con el apoyo de técnicas analíticas que sean capaces de manejar apropiadamente el carácter concurrente, las prioridades y la sincronización entre las tareas actuantes.

Dada la extensión y complejidad del alcance propuesto, se ha establecido una etapa inicial en la cual se ha planteado una primera aproximación del prototipo SCADA, definiéndose las técnicas y herramientas a emplear, y construyéndose un primer modelo de diseño y análisis de tiempo real [1].

### 3. Metodología Adoptada

La metodología asumida para el desarrollo del sistema ha sido una versión adaptada del RUP (*Rational Unified Process*) particularizado para sistemas integrales hardware y software de tiempo real, y la notación empleada es el UML (*Unified Modeling Language*), con los mecanismos de extensión que prevé el OMG (*Object Management Group*) y aplicando el meta-modelo UML-MAST (*UML-Modeling and Analysis Suite for Real-Time Applications*). Este grupo de técnicas orientadas a objetos, proporcionan un soporte sólido con las mejores prácticas de la ingeniería de software, permitiendo evolucionar hacia un producto de alta calidad y que esencialmente garantiza una planificación predecible para cumplir con los requerimientos funcionales y de tiempo real, de una manera integrada.

El RUP organiza el desarrollo en dos dimensiones: a través del tiempo, expresada

en términos de ciclos, fases, iteraciones e hitos; y a través de las disciplinas descritas en actividades, flujos de trabajo, artefactos y participantes.

En la Figura 2 se muestra el flujo principal de trabajo asignado al *Análisis y Diseño*. Nótese que existen dos actividades explícitamente asignadas a la ingeniería de tiempo real: una para la construcción de componentes software, donde se incluyen las guías para el diseño de la concurrencia y sincronización y, otra para el estudio analítico de la respuesta de tiempo real.

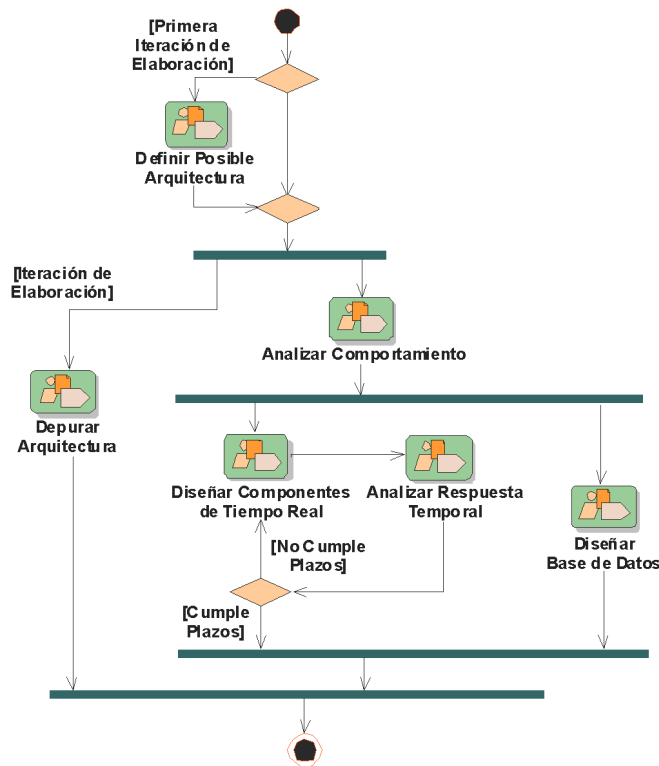


Figura 2: Flujo principal para la disciplina: *Análisis y Diseño*.

Cada una de las actividades se desglosa a su vez en otro conjunto de flujos que detallan los roles (responsabilidades), las actividades atómicas y los artefactos (modelos, documentos, etc.) que se generan en el proceso de desarrollo. En la Figura 3 se muestra el flujo detallado de la actividad *Analizar Respuesta Temporal*, en el cual se puede advertir la utilización de los modelos lógicos de análisis y diseño para la generación del modelo de tiempo real que es esencial para la aplicación de las técnicas analíticas de planificación de tareas.

Todos los elementos constitutivos de estos flujos de actividades se han documentado en detalle, tanto a un nivel de descripción como de lineamiento. El objeto principal de esta adecuación del RUP es que se constituya en un marco formal para hacerse extensible como guía en el desarrollo de proyectos similares.

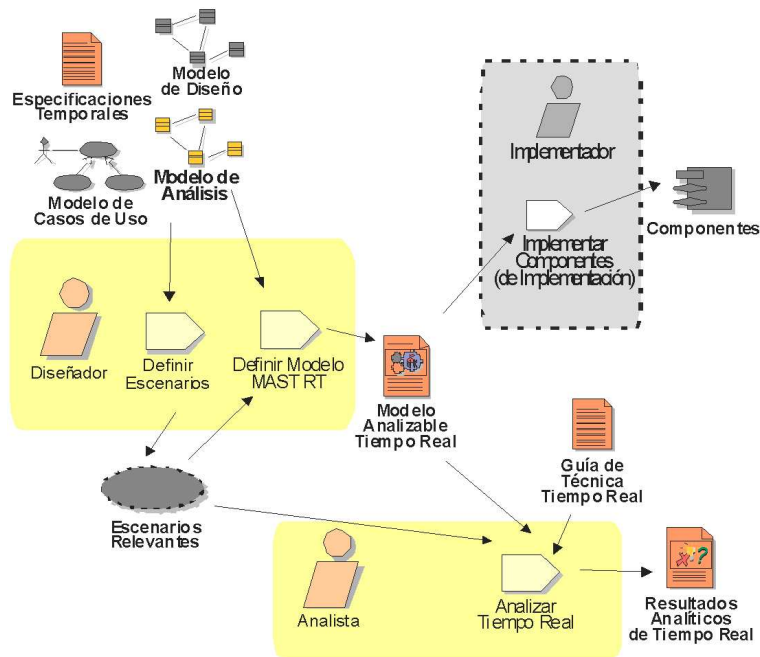


Figura 3: Flujo para la actividad: Análisis de Respuesta Temporal.

#### 4. Especificación de Requisitos

Para el planteamiento de requerimientos se aprovechó como base el *Sistema de Gestión de Red* (SIGGERE) desarrollado por la *Universidad de Cantabria*, para *Electra de Viesgo* (Norte de España) y adicionalmente se estudió un grupo de SCADAs del mercado, entre ellos la del *Grupo Iberdrola* (SCADA para el uso la compañía *Electropaz* de La Paz, Bolivia), lo cual ha permitido plantear un prototipo con funciones básicas y genéricas.

La estructura planteada para la especificación de los requisitos, inicialmente ofrece una *Visión* global de las características del sistema, para posteriormente detallar las propiedades de cada requerimiento en *Matrices de Atributos Funcionales y Temporales*. Los requerimientos correspondientes a transacciones visibles por parte del usuario se especifican en el *Modelo de Casos de Uso* y los requerimientos no funcionales, como las métricas de rendimiento y de calidad, se capturan en enunciados de texto.

Las matrices de atributos son un mecanismo de seguimiento de los requisitos en las etapas posteriores de desarrollo y adicionalmente proporcionan una primera aproximación de la organización de los requerimientos, para lo cual se han establecido 4 grupos funcionales (*adquisición de datos, control, almacenamiento histórico y gestión del sistema*). Cada uno de éstos se ha desglosado en requerimientos atómicos, 32 en total (por ejemplo: *procesar señales analógicas, activar mando sobre elemento eléctrico, almacenar tendencias de medidas, monitorear el estado de los recursos del sistemas, etc.*). A su vez, estos requerimientos atómicos se los ha cualificado en tablas que definen

sus atributos del tipo funcional por un lado y del tipo temporal por otro. En el Cuadro 1 se muestran algunos de estos atributos y su significado.

Tipo	Atributo	Descripción
Funcional	Complejidad	Describe detalles de procesamiento de la funcionalidad en cuestión.
	Fiabilidad	Establece los mecanismos para garantizar la integridad de la funcionalidad esperada.
	Seguridad	Define los mecanismos de seguridad de acceso a la funcionalidad.
	Parámetros	Establece los parámetros configurables para cada funcionalidad.
	Dimensión	Cuantifica el número de elementos que manejará la funcionalidad, por ejemplo el número de alarmas que soportará el sistema.
Temporal	Patrón de Llegada	Se discriminan: patrones periódicos, aperiódicos y esporádicos.
	Patrón de Sincronización	Comportamiento: asíncrono, síncrono en espera, síncrono de llamada y síncrono temporizado.
	Tiempo de Respuesta	Se acotan los tiempos de respuestas, divididos en tres grupos: nivel de actividad normal, nivel de actividad alta y nivel de actividad de punta.
	Criticidad	Clasifica la criticidad de los tiempos de respuesta en: no tiempo real, tiempo real no estricto y tiempo real estricto.

**Cuadro 1:** Atributos asignados a cada requisito.

Estas matrices facilitan la construcción del Modelo de Casos de Uso, por cuanto los *grupos funcionales* coinciden con los paquetes UML (existe un paquete adicional de actores), y los *requisitos atómicos* coinciden en un 95 % con los casos de uso asignados. Adicionalmente, cada caso de uso ha sido documentado, estableciéndose su flujo de operación en *Diagramas de Secuencia UML*.

## 5. Arquitectura del Sistema

El enfoque asumido para el diseño de la arquitectura, parte de una concepción distribuida y abierta de las partes lógicas del sistema SCADA, con la adaptación de interfaces estándares genéricas y la aplicación de modelos y mecanismos específicamente orientados al dominio eléctrico [6]. La estructura interna se modela haciendo énfasis en el comportamiento dinámico que es relevante para el estudio de tiempo real.

En la Figura 4 se muestra un esquema de la visión propuesta para que el sistema

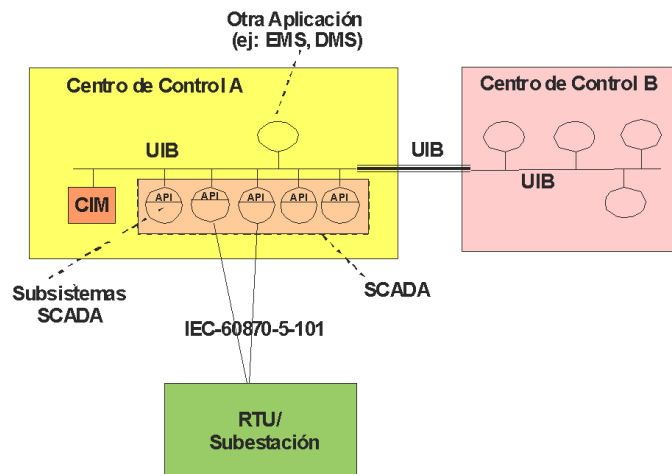


Figura 4: Arquitectura general del sistema SCADA.

cumpla con las características de sistema distribuido abierto. Son tres los elementos que dan soporte a la interconectividad: el bus software UIB (*Utility Integration Bus*), el cual constituye el medio por el cual se interconectarán los subsistemas SCADA entre sí y con otras aplicaciones; las API (DAIS, HDAIS, SCIS y SMIS) que representan la forma estándar de acceder a los servicios proporcionados por los subsistemas SCADA; y el CIM (*Common Information Model*) que permite contar con un modelo estándar de datos SCADA para que puedan ser entendidos entre los subsistemas y por aplicaciones externas al SCADA.

En nuestro prototipo mantenemos la interconexión entre el centro de control y las RTUs de manera separada al bus UIB. En su lugar utilizamos una arquitectura de comunicaciones basada en el EPA (*Enhanced Performance Architecture*), que contempla 3 niveles del estándar ISO de OSI: aplicación, enlace y físico. El protocolo es el IEC 60870-5-101, que está orientado a las comunicaciones en tiempo real precisamente entre centros de control y RTUs del sector eléctrico. En la Figura 5 se muestra el modelo en capas de la arquitectura general del sistema.

Dentro de los subsistemas SCADA considerados, el subsistema HMI (*Human-Machine Interface*) está distribuido en un entorno separado, que dentro del centro de control puede implementarse en varias máquinas o estaciones de trabajo separadas. Los demás subsistemas SCADA, para nuestro caso particular, estarán disponibles en un entorno central, constituido por una única máquina de procesamiento con altas prestaciones. Para el alcance de este trabajo se han definido cuatro subsistemas SCADA implantados en el entorno central: *Telemida, Control, Almacenamiento Histórico y Gestión del Sistema*.

Los subsistemas del sistema SCADA se han representado con cinco paquetes en las dos capas superiores del modelo, su función es la siguiente:

1. *Subsistema HUMAN\_MACHINE\_INTERFACE*: implementa la aplicación



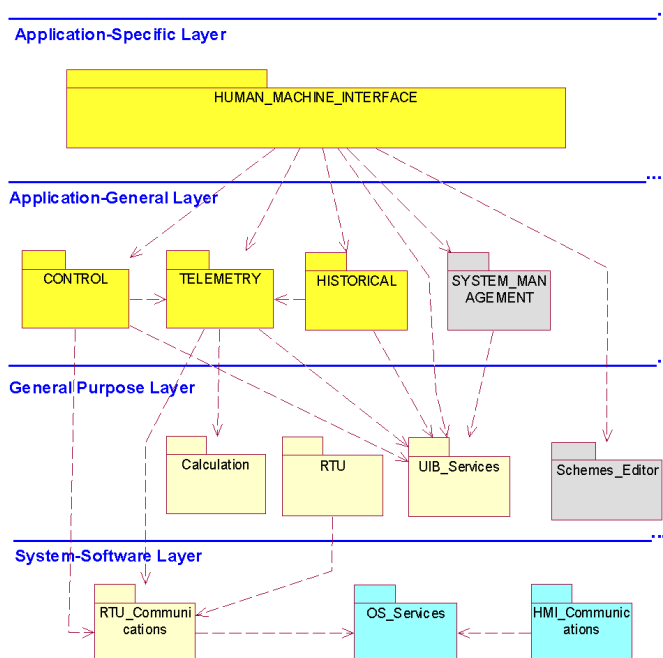


Figura 5: Estructura en capas de la arquitectura del sistema.

SCADA de cara a los usuarios, permitiendo ejecutar todas las acciones de operación, obtener el estado de la red eléctrica y administrar el sistema computacional. Este subsistema solicita los servicios de los subsistemas de la capa inmediatamente inferior a través del bus UIB.

2. *Subsistema TELEMETRY*: captura los datos de las RTUs, a través del sistema de comunicaciones RTU.Communications y, procesa la información realizando los cálculos programados sobre las medidas, procesando las alarmas y eventos. La información resultante la almacena en la base de datos representada por el CIM.
3. *Subsistema CONTROL*: ejecuta mandos de los elementos eléctricos maniobrables y permite ejecutar secuencias de control programadas. Toda vez que las secuencias de control requieren para su funcionamiento, de un conocimiento de los datos procesados por TELEMETRY, entonces se convierte en cliente de este último.
4. *Subsistema HISTORICAL*: Efectúa el almacenamiento en la base de datos de tiempo real y eventualmente en archivo por petición del usuario. La información almacenada es solicitada al subsistema TELEMETRY.
5. *Subsistema SYSTEM\_MANAGEMENT*: realiza las funciones de monitoreo de los programas y hardware del sistema, gestiona la seguridad en el acceso y permite crear salvaguardas de la información.

Las APIs que permiten el acceso a los subsistemas de la segunda capa (DAIS para TELEMETRY, HDAIS para HISTORICAL, SCIS para CONTROL y, SMIS para SYSTEM\_MANAGEMENT) se modelan dentro de los paquetes correspondientes.

En la capa de propósito general se muestran los modelos de la RTU, el bus UIB: **UIB\_Services** y dos utilitarios de uso genérico: **Calculation**, que es un programa que tiene un conjunto de funciones estándar y que ejecuta, en base a las llamadas del subsistema TELEMETRY, operaciones de cálculo sobre valores determinados; y **Schemes\_Editor** que es un programa orientado a manejar una librería de objetos gráficos que pueden ser mapeados a objetos específicos del CIM, para construir esquemas gráficos de las partes de la red eléctrica.

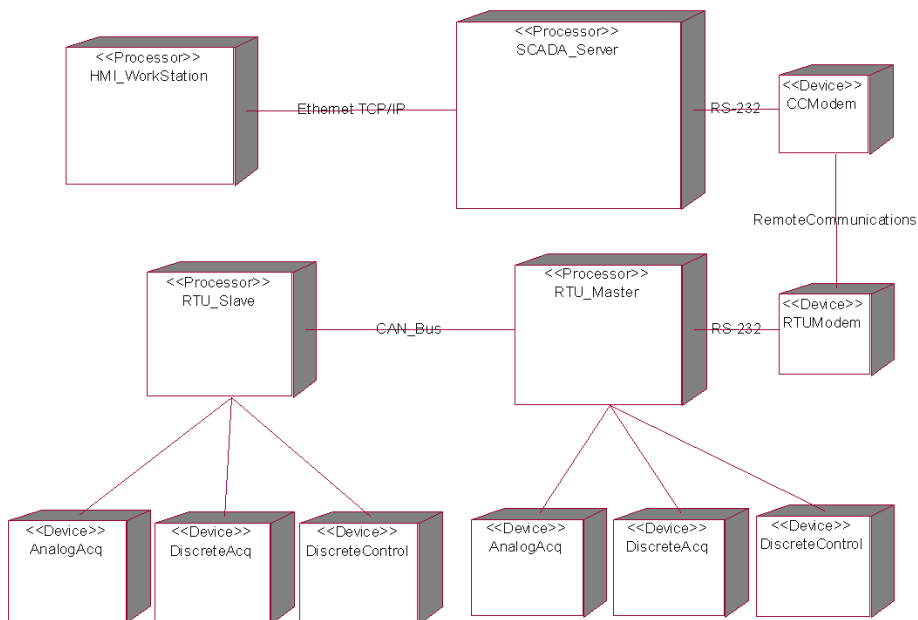
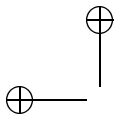


Figura 6: Vista de Distribución del Sistema SCADA.

La capa inferior modela las comunicaciones IEC 60870-5-101 entre el centro de control y la RTU (**RTU\_Communications**), siendo los subsistemas CONTROL y TELEMETRY los que interactúan con éste paquete por parte del centro de control y el paquete RTU por parte de la terminal remota. Adicionalmente, existe un paquete (**HMI\_Communications**) para modelar las comunicaciones entre el entorno central y las estaciones de trabajo que implementan el HUMAN\_MACHINE\_INTERFACE. Éstas comunicaciones están basadas en el protocolo TCP/IP. Ambos paquetes que modelan las comunicaciones utilizan los servicios del Sistema Operativo (**OS\_Services**).

Cada uno de estos paquetes ha sido organizado en otros sub-paquetes que contienen en detalle los modelos estáticos (de clases) y dinámicos (de secuencia y de colaboración), los cuales constituyen la vista lógica de la arquitectura.

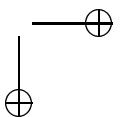
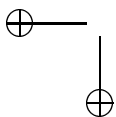


Por otro lado, en la Figura 6 se muestra la vista UML de distribución o despliegue de nuestro sistema, la cual representa la arquitectura física que da soporte a la aplicación SCADA. El centro de control se comunica con las RTUs a través de un medio de comunicación tal como fibra óptica o una red telefónica privada, utilizando dispositivos DCE para transmitir los datos. En el lado de las subestaciones, una RTU se ha dividido en dos tipos de nodos de procesamiento: un RTU\_Master y uno o varios RTU\_Slave. La descomposición en estos dos tipos de entidades, esta orientada a hacer escalable la arquitectura de las RTUs y a distribuir la carga de trabajo en procesadores diferentes. Tanto RTU\_Master como RTU\_Slave gestionan un conjunto de tarjetas de adquisición de datos y control, y se comunican entre sí mediante un bus CAN orientado a la transmisión en tiempo real.

## 6. Modelado de Tiempo Real

En base a los modelos lógicos de la arquitectura del sistema, se aplicó el meta-modelo UML-MAST para representar los parámetros de tiempo real que hacen a la dinámica de los subsistemas SCADA. El UML-MAST representa una vista adicional dentro del *Modelo de las 4 + 1 Vistas* para la descripción del comportamiento de tiempo real. A través de ella el diseñador puede construir gradualmente el modelo de tiempo real de forma paralela al desarrollo de su modelo lógico. Este modelo puede ser analizado por un conjunto de herramientas automáticas (MAST) relativas al análisis de planificabilidad, asignación óptima de prioridades, detección de bloqueos, etc. El UML-MAST se compone de tres sub-vistas complementarias, cada una de las cuales describe un aspecto específico del modelo de tiempo real:

1. **Modelo de la plataforma:** Modela la capacidad de procesamiento y las restricciones operativas de los recursos de procesamiento hardware y software. Estos recursos son: procesadores, coprocesadores, equipos hardware específicos, redes de comunicación, etc. En la Figura 7 se muestra un ejemplo para el nodo RTU Maestro, en el que se especifican los parámetros de tiempo real asignados al procesador, tales como la velocidad de procesamiento, los rangos de prioridades, los tiempos de cambios de contexto, etc.
2. **Modelo de los componentes lógicos:** Modela la cantidad de procesado que requiere la ejecución de las operaciones funcionales definidas en los componentes que se representan en el diseño lógico del sistema. Estos son los métodos, procedimientos y funciones definidos en las clases, las primitivas de sincronización de hilos, procesos de comunicación por las redes, operaciones que realizan los dispositivos hardware, etc. En la Figura 8 se muestra un ejemplo, para la captura de datos por parte del RTU Maestro, donde se establecen los tiempos de ejecución de las operaciones y el acceso a los recursos compartidos.
3. **Escenarios de tiempo real:** Modelan las diferentes configuraciones hardware y software que puede alcanzar el sistema y en las que se establecen requerimientos de tiempo real. Cada escenario se modela como un conjunto de transacciones



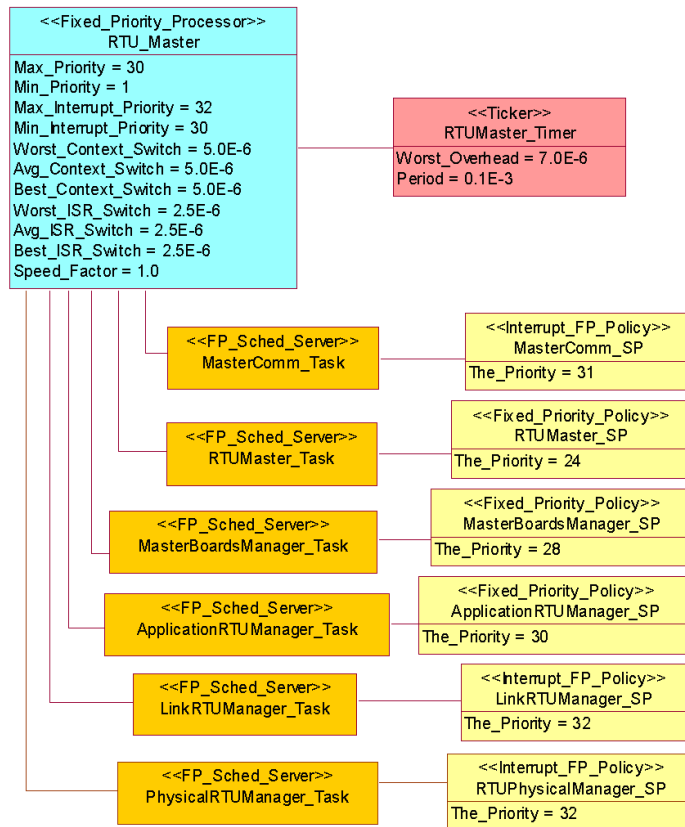


Figura 7: Modelo de la Plataforma: *RTU Maestro*.

que describen las secuencias de eventos y actividades que deben ser analizadas para que se satisfagan los requerimientos de tiempo real establecidos en ellas. El conjunto de transacciones de un escenario constituye la carga del sistema en esa configuración y afecta al análisis de cada una de ellas. En la Figura 9, se muestra el escenario para: *la lectura de los datos por parte de las RTU maestro y esclavo, escrutinio de los datos desde el maestro RTU y, sondeo y procesamiento de datos en el servidor SCADA*. En este modelo se establecen: los tiempos de la ocurrencia periódica de eventos y los plazos establecidos para cada transacción. Dentro de cada transacción se modela la secuencia de operaciones involucradas y la plataforma sobre la que se ejecutan (diagramas de actividad).

Puesto que la etapa en que se encuentra el desarrollo del sistema es de diseño, los parámetros asignados al modelo de tiempo real han sido extraídos de elementos similares implementados por el CTR (tiempos de procesamiento, cambio de contexto, etc.), y de las características propias de normalización de cada elemento (rango de prioridades de los sistemas operativos, velocidad del bus CAN, etc.). Si bien es en la etapa de

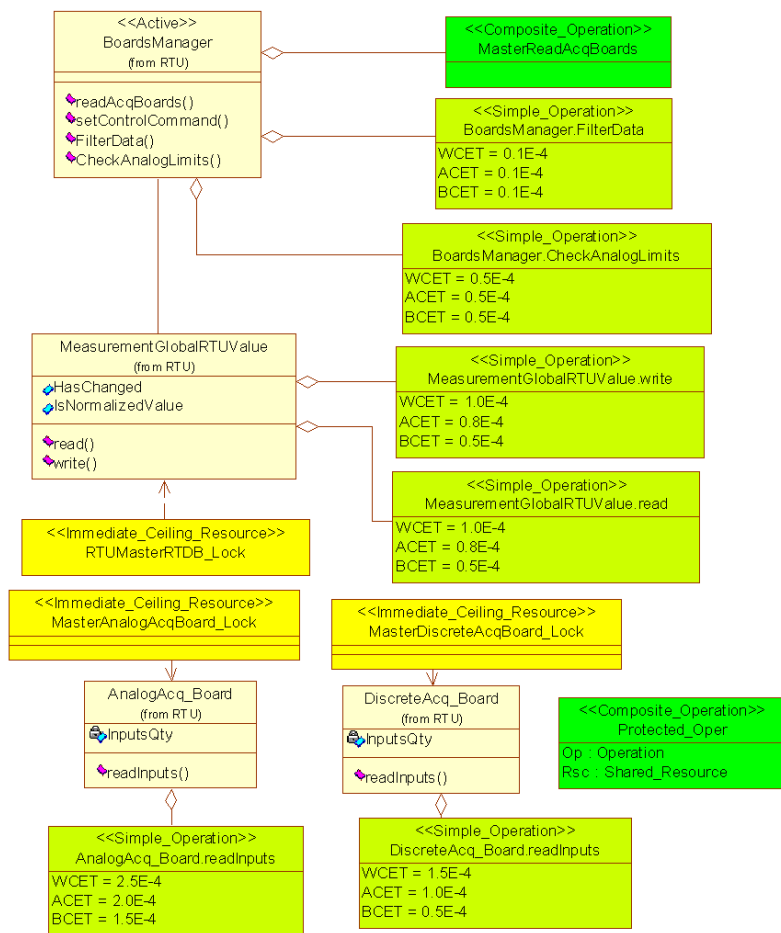


Figura 8: Modelo de los Componentes Lógicos: *Captura de Datos-RTU Maestro*.

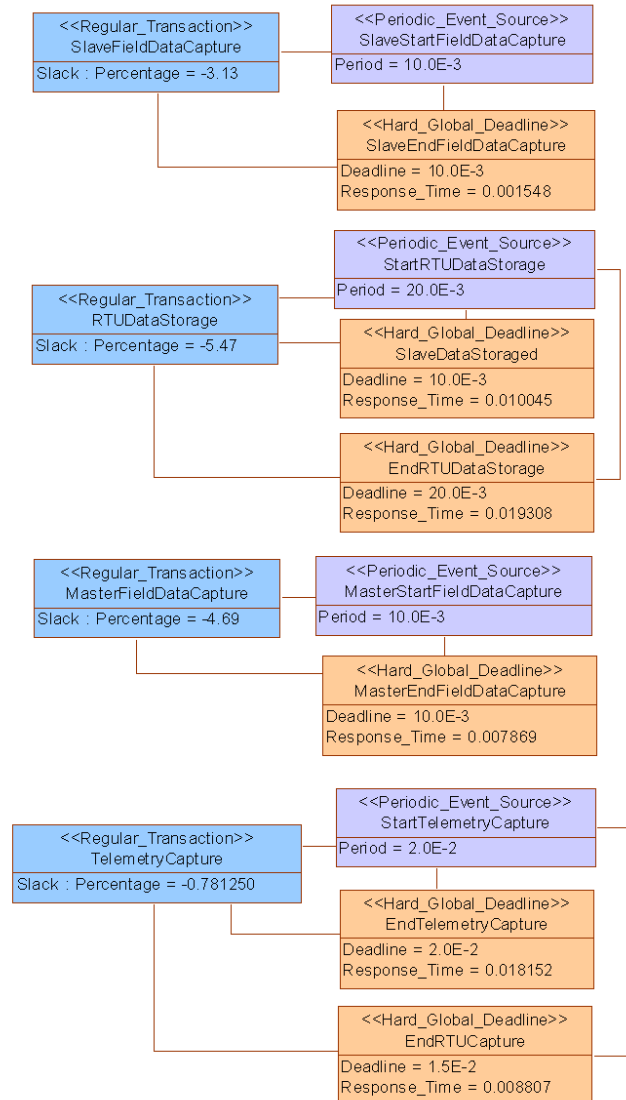
implementación que podrán verificarse y afinarse los parámetros asignados, para un estudio inicial del comportamiento del sistema, los valores establecidos son apropiados.

## 7. Análisis de Tiempo Real

Una vez definido el modelo de tiempo real, el análisis de planificabilidad se efectúa desde la propia herramienta ROSE -empleada para la asistencia del modelado-, a través de un menú que se incorpora al mismo cuando se instala la plataforma MAST.

La secuencia de operación consiste en:

1. Revisar la consistencia del modelo.



**Figura 9:** Escenarios de Tiempo Real: *Adquisición de Datos-SCADA*

2. Compilar el modelo en archivos MAST.
3. Analizar los resultados llamando a la herramienta MAST.
4. Cargar los resultados de planificabilidad al ROSE.

En una primera fase del trabajo, se han asumido algunas consideraciones para orientar el análisis de tiempo real a la generación de un conjunto de recomendaciones para

el diseño del hardware SCADA (cantidad de tarjetas RTU esclavas a utilizar, velocidad de transmisión a través del bus CAN y entre centro de control y subestaciones, entre otras).

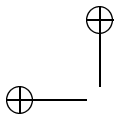
Asimismo, inicialmente no se efectúa un estudio comparativo de algoritmos de planificación, ni una evaluación de la eficiencia de los mismos. En este sentido, se aplica el algoritmo *Offset Optimizado Dinámico*, que es parte de las extensiones de la teoría RMA para optimizar la planificación de tareas distribuidas y que pueden bloquearse entre sí.

Para un estudio de la carga sobre cada uno de los nodos, se han establecido como variables de análisis: el número de *tags* a procesar por las RTU (entradas, salidas y parámetros), el número de RTUs que maneja el SCADA, y la cantidad de eventos procesados por el servidor SCADA. Bajo este criterio se han tabulado los resultados en esquemas como el de el Cuadro 2.

Transacción	Nro. de Tags	Respuesta (ms)	Stack (%)	Utilización (%)
<b>Maestro RTU Captura Datos Campo</b> Periodo del Evento Fuente = 10 ms. Plazo de la Transacción = 10 ms.	8	3.328	340.63	22.69
	16	4.080	193.75	34.04
	24	5.515	119.98	45.46
	32	5.987	72.98	57.81
	40	6.356	39.06	71.91
	48	7.117	15.63	86.48
	56	7.869	-1.56	101.58
	64	8.633	-14.84	117.43
	72	9.554	-25.78	134.73
	80	10.315	-34.28	152.16
<b>Esclavo RTU Captura Datos Campo</b> Periodo del Evento Fuente = 10 ms. Plazo de la Transacción = 10 ms.	8	1.548	350.78	22.18
	16	2.349	201.76	33.14
	24	3.021	132.65	42.98
	32	3.754	78.98	55.87
	40	4.432	51.12	66.17
	48	5.103	23.87	80.73
	56	5.958	1.56	98.46
	64	6.689	-14.87	117.47
<b>Maestro RTU Captura Datos de los Esclavos</b> Periodo del Evento Fuente = 20 ms. Plazo de la Transacción = 20 ms.	16	10.216	400.00	20.00
	32	11.019	268.75	27.12
	48	12.114	181.25	35.56
	64	11.234	121.45	45.16
	80	14.304	80.65	55.36
	96	15.395	52.87	65.42
	112	16.435	34.65	74.27
	128	17.521	19.21	83.89
	144	18.610	9.56	91.27
	160	19.720	1.56	98.46
176	20.799	-3.13	103.23	

**Cuadro 2:** Ejemplo de los resultados de planificabilidad.

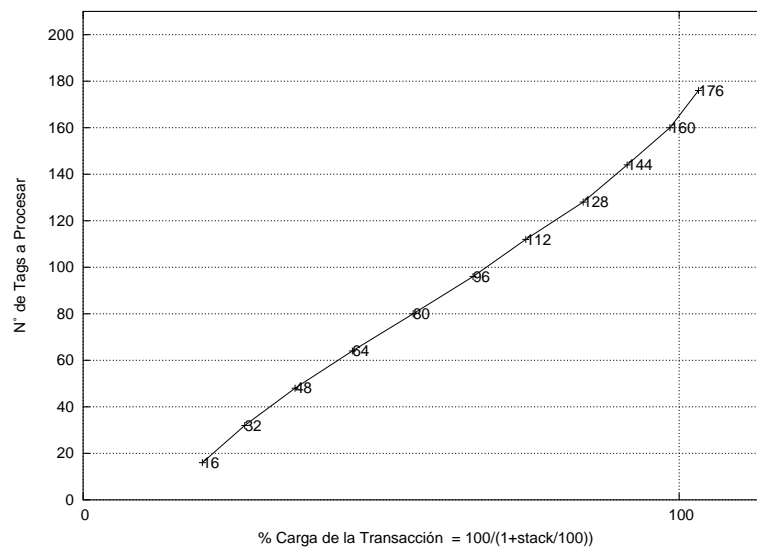
En este ejemplo se muestran algunas de las transacciones extraídas del modelo de *Escenarios de Tiempo Real*, con los respectivos periodos de sus eventos de disparo y los plazos de ejecución traducidos de los requisitos temporales. La variable de estudio para este caso es el número de *tags* procesados por el sistema, obteniéndose por la herramienta MAST: *los tiempos de respuesta y la holgura* (slack) – que se define como el porcentaje a incrementar uniformemente en los tiempos de las operaciones que intervienen en la transacción, para que el sistema deje de ser planificable–. Adicionalmente, se ha generado un parámetro de carga de procesamiento relacionado con cada transacción, calculado según:



$$\%CargaTransaccion = [1/(1 + Slack/100)] * 100 \quad (1)$$

Por tanto, el límite para que el sistema sea planificable es con [ $Slack = 0\%$ ] o [ $CargaTransaccion = 100\%$ ].

Cada una de las tablas obtenidas ha sido graficada en curvas del tipo mostrado en la Figura 10, donde se ha podido identificar explícitamente el límite de planificabilidad, para cada conjunto de parámetros.



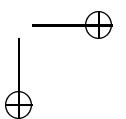
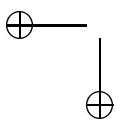
**Figura 10:** Carga de procesamiento para la transacción: *Maestro RTU Captura Datos de los Esclavos*.

En el Cuadro 3 se muestran algunas de las recomendaciones generadas, en base a los escenarios estudiados. Por ejemplo, se ha establecido que para esta arquitectura el número de tags que puede gestionar un maestro RTU sin necesidad de esclavos es de 52. Asimismo, se observa que el número total de Tags que puede gestionar el bus CAN para cumplir los plazos de ejecución impuestos sobre la RTU, es de 162.

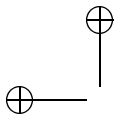
## 8. Conclusiones

Quizá el aporte más significativo de este trabajo es el planteamiento de un lineamiento formal de desarrollo de proyectos basada en marcos sólidos, analíticos y normalizados, cuya utilización se recomienda a objeto de optimizar costos, tiempo, calidad y universalidad de los mismos. Las conclusiones más remarcables del trabajo podemos resumirlas en las siguientes:

1. Como producto importante de este proyecto se ha propuesto una *especialización de la metodología de desarrollo de sistemas RUP, orientada a soportar el análisis y diseño de sistemas de tiempo real*. El modelo de extensión propuesto y la







Criterio	Cantidad	Recomendación
Cantidad de Tags a procesar por la RTU, con microprocesadores de 10 MHz en los nodos Maestro-RTU y/o Esclavo-RTU	52	Límite para utilizar solo un nodo maestro en la RTU
	56 (esclavo) + 52 (maestro)	Límite para utilizar un solo esclavo RTU
	162 (esclavos) + 52 (maestro)	Límite para el bus CAN: velocidad 500 KBytes/seg
Cantidad de RTUs	11	Límite para aumentar frontales de comunicaciones en el centro de control

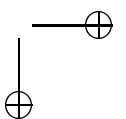
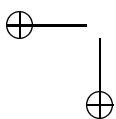
**Cuadro 3:** Recomendaciones como resultado del análisis.

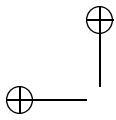
gran versatilidad del RUP, permiten ampliar y formalizar su especificación para otros proyectos, entre los cuales podemos sugerir: sistemas de control en general, sistemas de telecomunicaciones, sistemas puramente electrónicos y otros.

2. Se ha propuesto un formato y un *modelo de especificación de requerimientos* no sólo para sistemas de supervisión y control, sino también para cualquier otro sistema de tiempo real, cuya complejidad exige mantener una organización comprensible de la documentación. Dentro del dominio concreto del sector eléctrico, se ha aportado a contar con parámetros de tiempo real, que no estaban claramente establecidos a nivel de los procesos del centro de control.
3. La *arquitectura propuesta*, puede verse como un esquema básico, sobre el cual es factible realizar una gran cantidad de mejoras y extensiones, ya sea en la misma área de conocimiento, o en áreas afines de la ingeniería de los sistemas de control. Las características de adherencia a los estándares y el perfil de adopción de las mejoras prácticas en el diseño de arquitecturas de sistemas, hacen al sistema apropiado para continuar el estudio de este tipo de sistemas.
4. Se ha establecido un *modelo de comportamiento en tiempo real de sistemas SCADA* trabajando en aplicaciones del sector eléctrico, cuya problemática es tradicionalmente manejada por los fabricantes de este tipo de sistemas, sin que sus resultados puedan apoyar al diseño de nuevos modelos SCADA. A partir de este modelo, podrá evaluarse la respuesta del sistema, ante una amplia gama de escenarios posibles.

Este trabajo se encuentra en desarrollo y los temas inmediatos a abordar tienen que ver con las siguientes actividades:

- Optimización de la guía RUP propuesta para sistemas de tiempo real, en base a la experiencia acumulada.
- Depuración de los requerimientos de tiempo real, para hacerlos más universales.
- Diseño detallado de los módulos que no han sido completamente construidos en este trabajo.





- Extensión del estudio de tiempo real para incluir recomendaciones en los recursos software y comparación de algoritmos de planificación.
- Incremento de las exigencias de tiempo real a partir de la inclusión de funciones más críticas.

## Referencias

- [1] H. Espinoza. Sistema de supervisión y control de una red de distribución de energía eléctrica: Ingeniería de tiempo real. Tesis de Licenciatura, Universidad Mayor de San Andrés, La Paz, Bolivia, julio, 2002.
- [2] M. González Harbour, J. J. Gutiérrez, J. C. Palencia, y J. M. Drake. MAST: Modeling and Analysis Suite for Real-Time Applications. En *Euromicro Conference on Real-Time Systems*, Delft, The Netherlands, June, 2001.
- [3] Grupo de Computadores y Tiempo Real, Universidad de Cantabria-Electra de Viesgo. *Proyecto UIIMPC, Unidad informática integral de metrología, perturbometría y control*, 1992.
- [4] V. Hanson. Some aspects of computer loading problems in modern control centres. *Electra*, (114):112–127.
- [5] J. Nava. *Modelo Abstracto Distribuido y Entorno de Comunicaciones de Alto Nivel Orientado al Control de Redes Eléctrica*. Tesis Doctoral, Universidad de Cantabria, mayo, 1996.
- [6] J. Nava, J. M. Drake, y R. Menéndez. Architecture and distribution of functions in a distributed multiprocessor system for the supervision and control of an electric power system. En *Industrial Automation*, p 1026, Canada, julio, 1993.
- [7] OMG. *Profile for Schedulability, Performance and Time*, May, 2002. OMG document ad/2002-05-05.
- [8] L. Sha, M. Klein, y J. Goodenough. Rate monotonic analysis for real-time systems. Reporte Técnico CMU/SEI-91-TR-6, Carnegie-Mellon University. Software Engineering Institute, March, 1991.

