

# Sistemas de Software Distribuido

P. Pesado<sup>(1,2)</sup>, H. Ramón<sup>(1)</sup>, P. Thomas<sup>(1)</sup>, M. Boracchia<sup>(1)</sup>,  
I. Rodríguez<sup>(1)</sup>, E. Ibañez<sup>(1)</sup>, L. Marrero<sup>(1)</sup>, L. Delia<sup>(1)</sup>,  
C. De Vito<sup>(1)</sup>, N. Restelli<sup>(1)</sup>, G. Caseres<sup>(1)</sup>, J. Pettoruti<sup>(1)</sup>

<sup>(1)</sup>Instituto de Investigación en Informática LIDI (III-LIDI)  
Facultad de Informática – UNLP

<sup>(2)</sup> Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

{ppesado, hramon, pthomas, marcosb, ismael, eibanez, lmarrero, ldelia, cdevito, nrestelli,  
gcaseres,josep}@lidi.info.unlp.edu.ar

## CONTEXTO

Esta línea de Investigación forma parte del Proyecto “Sistemas de Software Distribuidos. Aplicaciones en procesos industriales, E-government y E-learning” del Instituto de Investigación en Informática LIDI acreditado por la UNLP y de proyectos específicos apoyados por CIC, Agencia y Telefónica, ALTEC e INASE.

## RESUMEN

El objetivo de este subproyecto es realizar investigación y desarrollo en temas relacionados con los aspectos de Ingeniería de Software que se orientan al desarrollo e implementación de proyectos concretos de Sistemas Distribuidos, manejo de datos físicamente distribuidos y soluciones de hardware y software para Sistemas Distribuidos de Tiempo Real.

En este contexto se trabaja en aspectos de la Ingeniería de Software de diferentes sistemas con inteligencia distribuida (computadoras, robots, teléfonos móviles).

**Palabras claves:** *Sistemas Distribuidos – Ingeniería de Requerimientos – Planificación – Metodologías de Desarrollo – Sistemas Web – Bases de Datos Distribuidas - Sistemas Distribuidos de Tiempo Real.- Robótica*

## 1 INTRODUCCION

Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan los recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada aunque ésta pueda estar implementada por varias máquinas en distintas ubicaciones [1].

El desarrollo de sistemas distribuidos es una necesidad a partir de la utilización de redes de computadoras y de computadores personales de alta performance.

Algunas ventajas del procesamiento distribuido son:

- Mejora de la disponibilidad: la operación es factible en una configuración reducida cuando algunos nodos están temporalmente no disponibles.
- Configuración más flexible: una aplicación puede configurarse de distintas maneras, seleccionando el número apropiado de nodos para una instancia dada.
- Control y administración más localizada: un subsistema distribuido, ejecutando en su propio nodo, puede diseñarse para ser autónomo, de modo que puede ejecutar en

relativa independencia de otros subsistemas en otros nodos.

- Expansión incremental del sistema: si existe sobrecarga, el sistema puede expandirse agregando más nodos.
- Costo reducido: con frecuencia una solución distribuida es más barata que una centralizada.
- Balance de carga: en algunas aplicaciones la carga total del sistema puede ser compartida entre varios nodos.
- Manejo eficiente de datos distribuidos físicamente.
- Mejora en el tiempo de respuesta: los usuarios locales en nodos locales pueden obtener respuestas más rápidas a sus requerimientos.

Las características de los Sistemas Distribuidos conducen a la utilidad de desarrollar prácticas de Ingeniería de Software que apunten a los distintos aspectos del desarrollo de sistemas desde la captura de requerimientos y la planificación, pasando por las metodologías de desarrollo, hasta la verificación y simulación de procesos distribuidos, incluyendo el aseguramiento de calidad.

En particular un sistema distribuido de tiempo real debe interactuar con el mundo real, en puntos físicamente distantes y no necesariamente fijos, en períodos de tiempo que vienen determinados por el contexto o las restricciones de la especificación (en muchos casos a partir de una activación asincrónica).

Algunas de las dificultades principales del desarrollo de software para sistemas distribuidos de tiempo real son:

- Modelizar condiciones de concurrencia y paralelismo.
- Manejar las comunicaciones inter-procesos e inter-procesadores.
- Tratamiento de señales en tiempo real.
- Tratamiento de interrupciones y mensajes asincrónicos con diferente prioridad.
- Detectar y controlar condiciones de falla, a nivel de software, de procesadores y de comunicaciones. Prever diferentes grados de recuperación del sistema.

- Asegurar la confiabilidad de los datos y analizar su migración en condiciones de funcionamiento normal o de falla.
- Organizar y despachar la atención de procesos, manejando las restricciones de tiempo especificadas.
- Testear y poner a punto un sistema físicamente distribuido.

Todas estas dificultades conducen a la utilidad de desarrollar herramientas de Ingeniería de Software orientadas a STR y SDTR, de modo de trabajar en la modelización, especificación y verificación del software considerando las restricciones temporales.

La Ingeniería de Software comprende la aplicación de principios científicos para realizar la transformación ordenada de un problema en una solución elaborada de software, y el mantenimiento subsecuente de ese software hasta el final de su vida útil [2]. La utilización de estas prácticas para resolver sistemas distribuidos y de tiempo real hacen necesaria su adaptación en función de las características de dichos sistemas.

La adopción de un enfoque ingenieril para el desarrollo de software, genera una serie de fases o estados conformando un ciclo de vida. Este ciclo de vida está guiado por una planificación que incluye el conjunto de acciones a realizar, y los productos generados por la aplicación del plan (inclusive el mismo plan) están administrados por diferentes Metodologías de Gestión y Desarrollo [3].

En el recorrido del ciclo de vida del desarrollo del software, la fase inicial comprende a la Ingeniería de Requerimientos que permite comprender, documentar y acordar sobre el alcance del problema, teniendo esto impacto directo sobre la Planificación y la Gestión del Proyecto de acuerdo a la Metodología de desarrollo seleccionada para el mismo [4]. Este no es el único impacto que justifica el énfasis en la Ingeniería de Requerimientos, ya que esta disciplina permite establecer claramente *que* se debe hacer posponiendo el *cómo* a etapas posteriores en el ciclo de vida,

constituyendo el fundamento de la construcción de un Sistema de Software [5].

El modelo de procesos elegido para el desarrollo de software define las actividades a realizar para la generación de productos de acuerdo a los objetivos planteados [6].

Asociado con la Ingeniería de Software de Sistemas Distribuidos, está el problema de utilizar un entorno WEB para los servicios que ofrece el Sistema. La tendencia creciente al desarrollo de arquitecturas centradas en un servidor (o un conjunto de servidores distribuidos) que ofrecen una interfaz WEB a los usuarios, ha generado un importante desarrollo de la Investigación en metodologías y herramientas orientadas a Sistemas WEB, así como ha obligado a establecer nuevas métricas y parámetros de aseguramiento de la Calidad para tales Sistemas. [7] [8] [9]

Toda aplicación esta dividida, al menos, en dos subsistemas, front-end y back-end. El front-end es una aplicación que interactúa en forma transaccional con los usuarios (clientes, empleados, proveedores, entre otros) de una Organización y requiere del apoyo del subsistema back-end que posee características analíticas y de workflow. Un ECA-rules (Event-Condition-Action) representan la tecnología para BD activas y el método natural para soportar dicha funcionalidad. ECA rules puede ser utilizado para actividades de chequeo automático de restricciones, o mensajes. Una cuestión importante, asociada a un ECA es cómo predecir el comportamiento en tiempo de ejecución. ECA rules posee una sintaxis declarativa de alto nivel, que le permite llevar a cabo acciones automáticas en respuesta a condiciones previamente definidas. Un ECA-rules sobre un subsistema Back-End permite controlar de manera transparente para el usuario un conjunto de reglas que hacen a su negocio. Estos controles se realizan directamente en el servidor de la aplicación y, ante una acción, disparan los mensajes correspondientes sobre el usuario en su aplicación front-end. [10] [11]

Por último, el modelo distribuido de datos hace posible la integración de BD heterogéneas proveyendo una independencia global del administrador de bases de datos respecto del esquema conceptual. [12].

## **2 LINEAS DE INVESTIGACION Y DESARROLLO**

- Conceptos de procesamiento distribuido. Arquitectura, comunicaciones y software.
- Metodologías de especificación, validación y desarrollo de SSD y SSDTR.
- Metodologías ágiles de desarrollo utilizando frameworks propios y disponibles de uso libre con diferentes tecnologías.
- Ingeniería de Requerimientos, en particular de sistemas distribuidos.
- Planificación de tareas para desarrollo de sistemas distribuidos.
- Bases de Datos Distribuidas [13]
- Herramientas de integración y mantenimiento de proyectos distribuidos.
- Lenguajes y ambientes para procesamiento distribuido.
- Reingeniería de sistemas complejos que migran por down-sizing a esquemas cliente-servidor distribuidos.
- Sistemas de Tiempo Real. Sistemas Distribuidos de Tiempo Real.
- Tratamiento de señales en tiempo real.
- Sistemas de control de robots en tiempo real. Algoritmos colaborativos entre máquinas móviles.
- Sistemas con identificación segura en tiempo real.
- ECA-rules sobre un subsistema Back-End permite controlar de manera transparente para el usuario un conjunto de reglas que definen su comportamiento [14]

## **3 RESULTADOS ESPERADOS/OBTENIDOS**

- Avanzar en la capacitación continua de los miembros de la línea de investigación.
- Desarrollar soluciones a problemas concretos de software de sistemas distribuidos, poniendo énfasis en el desarrollo de metodologías y herramientas específicas para clases de aplicaciones.

- Framework NET4DB desarrollada en el III-LIDI que permite generación automática de código .NET, interactuando con diferentes motores de BDD [14] [15]. Actualmente se estudian herramientas que permiten tomar un modelo UML de un sistema y derivar código a partir de él o realizar una verificación de consistencia lógica de dicho modelo.
- El motor simpleECA para resolver el back-end de PHP4DB [16] y NET4DB.

Se ha realizado transferencia a los siguientes Organismos.

### **3.1 Instituto Nacional de Semillas (INASE):**

El Instituto Nacional de Semillas dependiente de la Secretaría de Agricultura de la Nación fiscaliza la comercialización de semillas para siembra, con el propósito de certificar la calidad de estas semillas.

Para realizar este control, el organismo utiliza un Sistema Web donde se tramitan las presentaciones realizadas por las empresas vendedoras de semillas.

El III-LIDI tiene a cargo el proceso de reingeniería de este Sistema, con el objetivo de que las empresas mencionadas previamente puedan realizar trámites ante el INASE utilizando este software.

Con este proyecto se pretende descentralizar la distribución de numeración de rótulos utilizados para identificar a las semillas con “control de calidad realizado”, así como facilitar la gestión de trámites, permitiendo que varias etapas del proceso generado con estos trámites, sean cumplidas directamente por las empresas.

El resultado esperado es el fortalecimiento del proceso de fiscalización nacional de semillas, dado que las empresas podrán gestionar sus trámites a través de Internet, pudiendo controlar la identidad de la semilla en el mercado a través de la información del rótulo que la identifica, así como mantener el seguimiento de todo trámite iniciado en tiempo real.

### **3.2 Proyecto de Portal de la Innovación en TICs – Fase Primera**

El portal de innovación es una herramienta de comunicación comunitaria y social cuyo fin es soportar las actividades de innovación de las Instituciones de I+D y las empresas TIC de la Provincia de Bs. As. Al mismo tiempo se instaura como espacio por excelencia para información, capacitación, networking y colaboración en temas de innovación TIC.

En la primera fase, el proyecto propone desarrollar la tecnología base que permita publicar contenidos ricos, poner en marcha las áreas funcionales de desarrollo de contenidos y consultoría/capacitación, e investigar funcionalidad de redes sociales y colaboración para el contexto de la innovación en TICs.

La demanda local de necesidades de los proyectos de innovación es amplia pero dispersa y no hay organismo o programas públicos no privados que se ocupen sistemáticamente, de organizarla, y buscar soluciones en la oferta en el sector científico y privado, las necesidades de un portal para:

- Promover el intercambio de información sobre oferta y demanda de proyectos de innovación y experiencias de I+D entre los actores de la comunidad TIC.
- Sacar del aislamiento a los focos de innovación e interconectarlos entre sí.
- Crear herramientas que sistematicen la comunicación y vinculación a una escala local-regional-nacional.
- Proveer servicios de transferencia de capacidades para el nivel local (capacitación en gerenciamiento tecnológico, financiamiento a empresas de base tecnológicas e internacionalización).

El portal está alineado con las tendencias de Web 2.0 dando libertad para la generación de contenido para la comunicación entre actores.

### **3.3 Sistema de Billing para Locutorios VoIP**

Este proyecto forma parte del acuerdo entre el III-LIDI y el Grupo Angras. Se realizó la

implementación de un sistema de billing para el manejo de locutorios implementados sobre sistemas de Voz sobre IP (VoIP).

Se implementó un emulador del sistema POSNet para realizar testing y un prototipo del sistema evitando detalles de implementación en el hardware.

En el marco de la arquitectura VoIP se decidió utilizar Asterisk como central telefónica (PBX VoIP) para el desarrollo del proyecto, haciendo uso de la distribución de software libre AsteriskNow.

La implementación del proyecto se realizaron en C utilizando, sockets, librerías Standard ANSI y pthreads; esta última para el manejo de concurrencia en la aplicación.

La comunicación entre el servidor de billing y los clientes POSNet, requirieron la definición de un protocolo de comunicación adecuado al marco del problema, que trabaje sobre el protocolo de Internet TCP/IP. Se utilizaron los beneficios que brinda TCP pues proporciona una cantidad considerablemente mayor de servicios a las aplicaciones que UDP, como ser: la recuperación de errores, control de flujo y fiabilidad.

Como una evolución natural del desarrollo, se extendió la implementación del servidor de billing para posibilitar la atención a múltiples POSNets, verificando cada método de protección de la ejecución concurrente.

### **3.4 Convenio de Cooperación con Alta Tecnología Sociedad del Estado (ALTEC SE)**

El proyecto Intranet Pública Provincial se basa en una Única Red de Datos, este moderno sistema permitirá que los diferentes organismos públicos, empresas públicas, y beneficiarios de servicios públicos reales y potenciales distribuidos territorialmente a grandes distancias se conecten a través de una Intranet basada en estándares predefinidos y administrada en forma centralizada. El objetivo principal es el de brindar comunicaciones de datos a los Organismos rionegrinos optimizando los servicios que presta el Estado en las áreas de educación,

salud y seguridad. Las tareas a desarrollar son:

- Análisis de la Performance de los sistemas de software utilizados en la Pcia. de Río Negro (Dirección General de Rentas, Policía de Río Negro, Dirección de Catastro, Registro de la Propiedad del Inmueble, Ministerio de Educación, Ministerio de Salud), sobre los cuales ALTEC tiene responsabilidad.
- Diseño del manual de procedimientos de prueba de la intranet Provincia. Esto debe incluir comunicaciones puras y sobrecarga con los sistemas anteriores
- Análisis de los resultados obtenidos con las sugerencias que surjan de dicho análisis para la infraestructura y arquitecturas de sistemas.

## **4 FORMACION DE RECURSOS HUMANOS**

Los integrantes de esta línea de investigación dirigen Tesinas de Grado en la Facultad de Informática, 1 Tesis Doctoral, 2 Tesinas de Grado y Becarios III-LIDI en temas relacionados con el proyecto y participan en el dictado de asignaturas/cursos de postgrado de la Facultad de Informática de la UNLP.

## **5 BIBLIOGRAFIA**

- [1] G. Coulouris. Distributed Systems – Concepts and Design. Addison-Wesley. 1994.
- [2] R. Pressman. Ingeniería de Software: Un Enfoque Práctico. McGraw-Hill. 2002
- [3] R. Wysocki. Effective Project Management: Traditional, Adaptive, Extreme, .Wiley .2003
- [4] Loucopoulos, P; Karakosas, V.. Systems Requirements Engineering. McGraw Hill. Book Company. 1995
- [5] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, Wiley. 1998
- [6] Pleegeer. Ingeniería de Software: Teoría y Práctica. Prentice-Hall. 2002

[7] Stephen Kan. .Metrics and Models in Software Quality Engineering (2nd Edition).Addison Wesley. 2003

[8] Offutt J., “Quality Attributes of Web Software Applications”. IEEE Software: Special, Issue on Software Engineering of Internet Software 19 (2):25-32, Marzo/Abril 2002.

[9] Wu, Y. y Offutt, J. “Modeling and testing web-based Applications”.  
<https://citeseer.ist.psu.edu/551504.html>: 1-12, Julio 2004

[10] Automatic Control of Workflow Processes Using ECA Rules, Joonsoo Bae, Hyerim Bae, Suk-Ho Kang, Yeongho Kim, IEEE Transactions on Knowledge and Data Engineering , Volume 16 , Issue 8 (August 2004), 2004, ISSN:1041-4347

[11] An Event-Condition-Action Language for XML, J. Bailey, A. Poulouvasilis, P. Word, WWW2002, May 7-11, 2001, Honolulu, Hawaii, USA. ACM 1-58113-449-5/02/0005.

[12] Silberschatz A et all: “Fundamentos de Bases de Datos”, Tercera Edicion Mc Graw Hill 1998

[12] Ozsu M. Valduriez, P. : “Principles of Distributed Database Systems”, Segunda Edicion. Prentice Hall 1999

[13] Mello S. J., “Executable UML”, Addison-Wesley, 2002.

[14] S. Ceri, P.Fraternalli,A. Bongio, “Web Modeling Language (WebML): a modeling language for designing web sites”. Computer Networks, vol 33, 2000.

[15] Tesis Stefan Böttger, Distributed Composite Event Monitoring. June, 2006 Ruprecht-Karls-Universität Heidelberg (Kirchhoff-Institut für Physik) – Universität Leipzig (Fakultät für Mathematik und Informatik – Institut für Informatik).

[16] Delia, Caseres, Ramón, Thomas, Bertone; Framework para el Desarrollo Ágil

de Aplicaciones Web, CACIC 2006, San Luis, 17 al 21 de octubre de 2006.