

CAPITULO 1. ANTECEDENTES

Muchos de los sistemas de información que están hoy funcionando en las organizaciones de nuestro país fueron desarrollados en los años ochenta. La irrupción de las tecnologías relacionadas con Internet, el paradigma orientado a objetos, los componentes distribuidos y la nueva mentalidad empresarial que intenta ofrecer mejores servicios a sus clientes, han provocado que la información que permanece en los viejos sistemas y que es totalmente aprovechable sea objeto de diversos tratamientos para fines de su recuperación. Más aún, en muchos casos se trata de conocimiento y experiencia sobre reglas del negocio que no están disponibles en ningún otro medio y esto lo convierte en un problema que puede considerarse crítico [Bisbal,1995].

La situación tecnológica en los años 80 sólo permitía una arquitectura física y lógica restringida a la oferta de los grandes fabricantes de software y hardware, quienes suministraban ambos componentes, por lo que se obtenía en consecuencia una dependencia total del cliente. Los sistemas fueron luego sometidos a un mantenimiento constante, normalmente indocumentado, lo que en muchos casos originó degradación de las aplicaciones y, por ende, un servicio deficiente para el usuario [Brodie, 1995].

Se llega así al concepto de “sistemas heredados” (Legacy Systems) que admiten diversas definiciones según el punto de vista considerado. Así Ulrich (1994) hace referencia a “sistemas independientes construidos en una era tecnológica anterior que disponen de precaria documentación”[Ulrich,1994], Brodie los define como “todo sistema de información que se resista significativamente a su modificación y evolución, para cubrir cambios en sus requerimientos”[Brodie, 1995], y Bennett (1995) habla de “grandes sistemas parcialmente desconocidos y vitales para las organizaciones”[Bennett,1995].

Ante una situación como la descrita, las organizaciones se vieron en la necesidad de tomar una determinación que puede ser resumida como un “cambio del sistema”, para adecuarlo a las nuevas necesidades, y que puede concretarse a través de una de las opciones siguientes:

- a) *Hacer reingeniería o migrar el sistema.*
- b) *Abandonar el sistema y sustituirlo por otro nuevo.*
- c) *Optar por una solución híbrida entre las dos anteriores.*

Como denominador común de todas las opciones anteriores, la organización debió plantearse la inclusión, dentro del proyecto correspondiente, de una buena administración de la evolución del nuevo software que se produzca, con el objetivo de no volver a caer en la situación de la que buscaban salir. Las pequeñas y grandes aplicaciones distribuidas, desarrolladas antes de la irrupción de la Web, que basaron su proceso de construcción en ambientes cliente-servidor estaban orientadas a la conexión en red de ordenadores personales (clientes), con servidores. En la mayoría de los casos, la herramienta de desarrollo permitía a equipos de programadores crear aplicaciones diseñadas con interfaces gráficas de usuario (GUI), y con acceso a información de base de datos locales o en servidores de red [Liem, 2006].

En este contexto, desde 1980 y hasta principios de 1990, maduraron diversos métodos de análisis y diseño de software. La aplicación del análisis y diseño estructurado para desarrollo de aplicaciones concurrentes, distribuidas y de tiempo real; el modelado de entidades del dominio del problema en términos de eventos; el uso de tareas concurrentes asociadas a entidades y los métodos de análisis y diseño orientado a objetos [Jacobson, 2000].

Tal como señala Piattini (2004), estos tipos de arquitecturas ofrecen una serie de inconvenientes; entre los cuales se destacan: a) falta de flexibilidad de los sistemas, debido al acoplamiento existente entre las aplicaciones y la base de datos, lo que origina que una modificación en la base repercute en la necesaria modificación de las aplicaciones asociadas; b) dificultades de migración, por la dependencia con el fabricante; c) problemas de escalabilidad, al incrementar el número de usuarios y d) dificultad de reutilización de componentes, debido al acoplamiento entre el nivel de presentación (interfaz del usuario) y los sistemas de administración de base de datos [Piattini, 2004].

En tanto, como evolución importante de la computación distribuida cliente-servidor, surge la tecnología Web; y, en consecuencia con ello, el interrogante sobre la conveniencia de migrar las aplicaciones existentes de tipo cliente-servidor a dicha arquitectura; para lo cual resulta necesario revisar algunas de las características diferenciadas que presentan estas tecnologías. En este sentido, se observa que en las arquitecturas tradicionales, la interfaz del sistema se instala en la computadora del usuario final; mientras que la arquitectura basada en la Web se presenta como una página Web que se muestra a través de un explorador Web, convirtiéndose en la interfaz del usuario final. Estos navegadores fueron diseñados, en principio, como herramientas de software con soporte gráfico, con el fin de desplazarse a través de nodos de información mediante el hipertexto. Con el tiempo, el navegador resultó una herramienta de trabajo para el acceso a otros sistemas [Cáceres, 2003].

Se ha hecho la migración de sistemas heredados desde plataformas origen tales como mainframes IBM y UNISYS hacia sistemas abiertos en UNIX o Linux. En este contexto, también puede mencionarse la conversión de sistemas desarrollados bajo lenguajes propietarios (RPG, COBOL, Natural, etc.) hacia Java u otros sistemas de código abierto.

En todos estos casos, los sistemas son convertidos buscando mantener la misma funcionalidad que los originales, sin modificar los procedimientos de la organización que los utiliza, pero tal como ya fue anticipado, muy poco se menciona de la consolidación y características de las pruebas y validaciones de las aplicaciones en entornos distribuidos a la Web.

1.1. Objetivos del Proyecto

Objetivo General:

Para este reporte se considera el caso de un sistema distribuido no basado en la Web, utilizado por el **Instituto Aeronáutico del Noroeste ubicado en la ciudad de Tijuana B.C. para la administración de inscripciones de los alumnos**, para el cual se dispone de especificación, diseño y código, y se determinó la conveniencia de construir una aplicación basada en la Web que preservara las principales propiedades de la aplicación original, tales como son su especificación, funcionalidad y propiedades de la interfaz gráfica con el usuario.

Objetivos específicos:

A fin de alcanzar el objetivo general propuesto, se plantean los siguientes objetivos específicos:

1. Construir modelos para abstraer propiedades en común de modelos de aplicaciones Web y de modelos de aplicaciones tradicionales, los que sirven de base para mapear casos de uso utilizados en aplicaciones distribuidas no Web a casos de uso de aplicaciones basadas en tecnología Web.
2. Formular una metodología de análisis para la migración de aplicaciones distribuidas a entornos Web, basada en un enfoque de testing con reutilización de casos de prueba y que prevea la utilización de herramientas automáticas para la ejecución y reuso de los casos de prueba generados.
3. Aplicar la metodología propuesta tomando como caso de estudio el Sistema de Administración Académica de una institución universitaria a efectos de comprobar su desempeño.
4. Definir metodologías de trabajo para la prueba de los resultados de las migraciones a la Web de aplicaciones distribuidas no realizadas bajo entorno Web, a partir de la consideración de las propiedades que se quieren preservar de estas últimas.
5. Documentar la información recopilada en función de los resultados obtenidos luego de realizar las pruebas de migración.