

CAPÍTULO 2. DESCRIPCION DEL PROYECTO

2.1 Introducción

En la construcción del sistema a migrar, al igual que la mayoría de las aplicaciones cliente-servidor en su tiempo, fue utilizada la metodología estructurada, y por lo tanto, es necesario utilizar las herramientas de modelado del análisis estructurado de Yourdon (1993) para conocer las estructuras de estos sistemas en su versión original. Las metodologías están estrechamente relacionadas con la construcción del software y su ciclo de vida. Por lo tanto, resultaría un error emprender un desarrollo o su posterior validación sin una metodología que gobierne estos procesos. Como es sabido, esta lección fue aprendida con un elevado costo y a ello se debe la existencia de la ingeniería de software [Yourdon, 1993].

Con respecto a la aplicación Web, la mayoría de los problemas del diseño Web pueden simplificarse si se utiliza la notación UML. Desde que UML fue adoptado como el lenguaje estándar para el modelado, se utiliza como medio de expresión de los diferentes modelos que se crean durante el desarrollo de los sistemas. Como estos desarrollos están guiados por los casos de uso, los mismos se emplean para definir los requisitos funcionales del sistema, y, en general, todas las etapas del proceso se articulan en torno a los casos de uso identificados. UML ofrece la oportunidad de establecer un marco común para la representación de especificaciones de requisitos, de desarrollo y de pruebas [Koch, 2002].

Los perfiles UML incorporan mecanismos de extensión (estereotipos, valores etiquetados y restricciones) que permiten personalizarlos para distintas aplicaciones y tecnologías y han sido utilizados para representar la arquitectura del sistema a migrar y la del sistema migrado [Pinheiro, 2000].

En este capítulo se presenta en primer término una metodología para el conocimiento del sistema a migrar, luego, se presenta una metodología para el conocimiento del nuevo sistema que ya ha sido migrado, y para esto último se pone el foco en la arquitectura cliente servidor, en la tecnología Web y en las reglas de negocio que deben ser reusadas para abordar el proceso de migración.

2.2 Metodología propuesta de análisis lógico y físico del sistema a migrar

Se presenta a continuación la metodología para conocer los detalles de la arquitectura y diseño del sistema anterior, además de las limitaciones tecnológicas y de las reglas de negocio existentes en el momento de su construcción.

2.2.1 Reconstrucción de la especificación de requerimientos

Se asigna mucha importancia a la reconstrucción de la especificación de requerimientos del sistema a ser migrado. Obviamente, la profundidad con la que pueda cumplirse esta tarea dependerá de cada caso, y la antigüedad del sistema será seguramente uno de los factores determinantes. En efecto, en la medida que la antigüedad sea mayor se dispondrá de menos testimonios y se caerá en el caso de los "sistemas heredados". Sin embargo, cualquiera que sea el caso tratado, todo el esfuerzo que pueda ser realizado para la reconstrucción de la especificación de requerimientos estará ampliamente justificado y será de gran valor en las próximas etapas.

a) Estudio preliminar

Realizar un análisis exhaustivo del sistema original, utilizando para ello técnicas de entrevistas a personas que hayan participado del mismo, desde el área técnica al área operativa. Recopilar toda la documentación disponible, incluyendo manuales de procedimientos, y analizar las reglas de administración del sistema, características generales, perfiles, procesos, etc.

b) Reconstrucción

Clasificar y ordenar toda la información testimonial y documental que pueda haberse obtenido en la etapa anterior.

2.2.2. Descripción de la dimensión funcional

La dimensión funcional es el eje de las etapas de análisis y diseño, tanto de las metodologías estructuradas como orientadas a objetos, por lo que su correcta definición es esencial en todo el proceso de validación de un sistema.

a) Elaboración del diagrama de contexto

Establecer las fronteras del sistema con los otros sistemas y agentes externos que se vinculan con el mismo mediante un Diagrama de Contexto. Este diagrama de contexto es un caso especial de diagrama de flujo de datos (DFD), denominado de nivel "0", que representa globalmente al sistema como una sola burbuja y donde un proceso único define la frontera o marco del análisis con el sistema externo. Así se definen las interfaces del sistema con el resto del universo.

b) Análisis de comportamiento del sistema

Confeccionar la lista de eventos o acontecimientos que recibe el sistema y a los cuales debe darse respuesta, precisando en cada caso la naturaleza del evento y el tipo de respuesta esperado. Una vez completada esta lista se podrán establecer los escenarios que componen los distintos subsistemas del sistema a migrar y se completará el estudio de su comportamiento.

c) Construcción de diagramas de flujo de datos - DFD

A partir de los tareas identificadas se deben desarrollar diagramas que permitan estudiar los flujos de datos y sus relaciones y transformaciones con los distintos procesos asociados. El diagrama de flujo de datos (DFD) es una técnica que representa el flujo de información y las transformaciones que se aplican a los datos al moverse desde la entrada hasta la salida. Cada proceso, representado por una burbuja, puede refinarse en otros DFD's de menor nivel para mostrar un mayor detalle.

d) Realización del modelo de casos de uso

El objetivo de esta tarea es especificar cada caso de uso identificado en el análisis del comportamiento del sistema, representando gráficamente los escenarios involucrados. Para cumplir con este objetivo se construye el Modelo de casos de uso de trazo grueso de la aplicación distribuida, identificándose las relaciones entre los mismos.

2.2.3. Descripción de la dimensión estática del sistema anterior

Normalmente, el modelo funcional conduce a una base sólida para completar luego la dimensión estática del sistema.

a) Reconstrucción del modelo conceptual

El modelo conceptual se utiliza para obtener una descripción del dominio del problema real, no es una descripción del diseño del software. Es usado como base para una vista unificada de los datos y se representa con un diagrama gráfico de estructura estática, con las distintas entidades que componen el diseño lógico del sistema, sus relaciones y cardinalidad.

2.2.4. Descripción de la interfaz de usuario

Como ya fue descrito con anterioridad, la interfaz de usuario es uno de los aspectos que sufre gran impacto en la migración de sistemas a entornos Web y por lo tanto esta interfaz debe ser descrita con el mayor detalle y cuidado.

a) Análisis del modelo de interfaz de usuario

Representar la interfaz de usuario, mostrando las distintas pantallas que intervienen en la aplicación a migrar y realizando un diagnóstico sobre su presentación y contenido. La interfaz de usuario es la herramienta que entiende a ambos y es capaz de traducir los mensajes que se intercambian por medio de un acceso amigable, congruente con sus necesidades y adecuado a principios ergonómicos.

b) Construcción del modelo del sistema

Se presentan, a través del uso del lenguaje unificado de modelado (UML), las interacciones entre el usuario cliente y la base de datos, mediante la utilización de modelos de formularios y de componentes, para lo cual existe una entidad central que despliega la información al usuario mediante la carga de forms.

Un form es un programa con los elementos necesarios para que el cliente pueda interactuar con la base de datos. Posee un conjunto de objetos que permiten, entre otras cosas, desplegar nuevos formularios, cajas de texto para ingresar datos, combos para seleccionar información existente, controles Data-Windows para acceso a datos y conexiones con Bases de Datos que facilitan la organización e interacción. El llamado de un form a otro es modelado por la asociación que se invoca a sí mismo [Cabero, 1995].

Los forms pueden ser estáticos o dinámicos mediante el uso de componentes de otros fabricantes, tales como librerías .dll, .ocx, entre otros. También contiene componentes generados por el propio equipo de desarrollo [Nielsen, 1990].

La organización en forms es representada por una asociación jerárquica, cuyo destino es un conjunto de entidades forms dependientes unas de otras. La subdivisión en forms tiene una asociación unaria, es decir, cada form puede recargar una y sólo una childform por vez, constituyendo esto una desventaja con respecto a las web page. Cuando un botón en un form fuerza la carga de otro form, el form de destino se convierte en el form activo [Cabero, 1995].

La interfaz existente entre los forms y la BaseDatos, es la que permite el uso del SGBD. El acceso se produce mediante una clase de componentes que provee los drivers nativos de conexión [Wegner, 1999].

2.2.5. Descripción de la arquitectura física y del software de base

a) Arquitectura física

Al describirse la arquitectura física del sistema a migrar deben considerarse dos aspectos principales: 1) tipo y capacidad de unidades centrales, dispositivos de almacenamiento, características de monitores y otros periféricos de entrada y salida (lectoras de códigos de barra, scanners, impresoras, ticeadoras, etc) y 2) Interconectividad entre los equipos (red LAN, WAN, etc). En la figura 1 puede verse el esquema descrito:

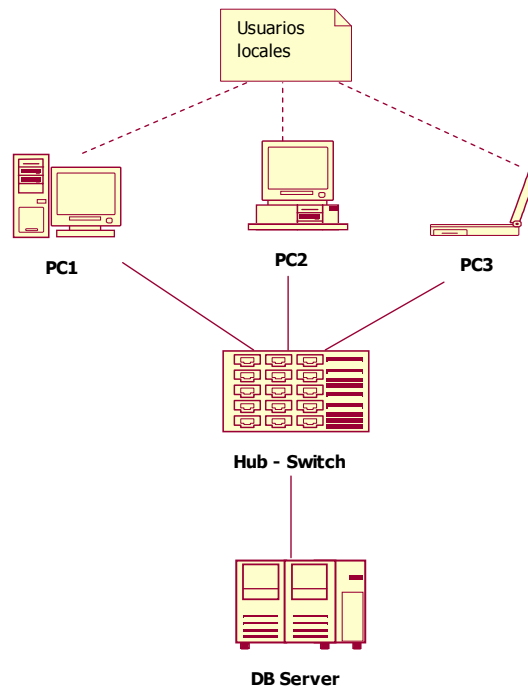


Figura 1 – Gráfico de componentes distribuidos del sistema

b) *Arquitectura del Software de Base y diagrama de componentes*

Describir la plataforma, incluyendo sistema operativo, interfaz gráfica, motor de base de datos (local o remota), uso de API y librerías propias del Sistema Operativo, otras librerías, componentes OCX, componentes DLL propias y de otros fabricantes.

Como se expresara con anterioridad, la aplicación a migrar constituye un sistema cliente-servidor, en un modelo bicapa, en el que sus clientes son FAT CLIENT, porque tanto los componentes de presentación como los de administración de datos se encuentran del lado del cliente, generando dificultades en el mantenimiento; en tanto que el manejo de las transacciones las realiza el DTS (Data Transaction Server) del lado del servidor [Brodie,1995].

2.2.6. *Análisis de las limitaciones del modelo anterior*

Describir las limitaciones o restricciones que se presentan en el sistema a migrar, tales como el empleo de determinadas metodologías de desarrollo, lenguajes de programación, normas particulares, restricciones de hardware, de sistema operativo etc. Debe además considerarse que, con mucha frecuencia, la migración de un sistema arrastra expectativas referidas a mejoras operativas y/o funcionales que resultan ajenas a la propia migración pero que también deben ser consideradas [Bisbal, 1999].

La arquitectura de dos capas presenta dificultades en la migración, por su dependencia con el fabricante del entorno operativo, el alto grado de acoplamiento entre las aplicaciones del cliente y la base de datos, y los problemas de escalabilidad, ya que este tipo de arquitectura no presenta buen rendimiento en su implementación en los sistemas donde crece el número de usuarios interactivos [Liem,2006].

En consecuencia, debido a la fuerte dependencia entre la aplicación residente en el cliente y la base de datos, al cliente se lo denomina como ya se ha mencionado, "cliente pesado" (fat client).

2.3 Metodología propuesta de análisis lógico y físico del sistema migrado

Otro de los pilares básicos en el proceso de migración es el conocimiento de las nuevas reglas de negocio que se presentan así como las características tecnológicas necesarias para el sistema en la Web. Con tal fin se presenta a continuación una metodología para conocer los detalles de la arquitectura y diseño del sistema migrado a la Web.

2.3.1. Adecuación de la especificación de requerimientos

a) Estudio preliminar

Descripción de los servicios generales que debe brindar el nuevo sistema, incluyendo aquellos que ofrecía el sistema anterior, con el detalle de las nuevas reglas de gestión y la tecnología a utilizar para la migración.

b) Reutilización de requisitos en el proceso de migración a la Web

Se propone el reuso de los requisitos que se trasladan al nuevo sistema y la identificación de los requisitos nuevos y aquellos sujetos a cambios en el nuevo sistema. Con fines de abordar el proceso posterior de reutilización de requisitos, resultó interesante el estudio comparativo de la metodología Web realizado por María José Escalona de la Universidad de Sevilla y Nora Koch de la Universidad de Munich (2002), que se sintetiza a continuación. El mismo plantea que el desarrollo de sistemas de aplicaciones Web agrupa una serie de características que lo hacen diferente del desarrollo de otros sistemas. Por un lado, hay que tener en cuenta que en el proceso participan roles muy diferentes de trabajadores: analistas, clientes, usuarios, desarrolladores, diseñadores gráficos, expertos en multimedia y seguridad, etc [Koch, 2002].

Por otro lado, la presencia en estos sistemas de una estructura de navegación a través de sus páginas, obliga a un desarrollo preciso en este aspecto que garantice que el usuario no se "pierde en el espacio de la navegación del sistema".

2.3.2. Adecuación de la descripción funcional

a) Revisión del modelo funcional

En función del rediseño del modelo de casos de uso de la nueva aplicación y de las reglas de administración acordadas, deben introducirse modificaciones al esquema funcional, que se reflejará en el nuevo modelo de casos de uso y en las especificaciones que surjan del mismo.

b) Realización del modelo de casos de uso de la aplicación web

Las nuevas reglas de administración que resultan de los requerimientos formales del nuevo sistema, plantean un nuevo modelo de casos de uso que atienda las modificaciones y agregados a la funcionalidad existente en la aplicación GUI.

En este nuevo diagrama de casos de casos de uso puede observarse el reuso de la funcionalidad de la aplicación anterior y, la incorporación de nueva funcionalidad a la aplicación migrada, acompañada de un reordenamiento de la misma por la inserción de la nueva tecnología.

2.3.3. Adecuación de la descripción de la dimensión estática

a) Revisión del modelo conceptual

Se revisa el modelo conceptual de la aplicación anterior a los efectos de establecer las nuevas entidades que lo componen, las que responden a la redefinición de las funcionalidades existentes, a las nuevas reglas de negocio relevadas y a las exigencias de la nueva tecnología a implementar.

Para superar las limitaciones existentes en el sistema a migrar, debe comprobarse la existencia de las nuevas reglas de gestión con la interacción entre todos los actores que juegan un rol importante en el sistema, estas reglas deben estar escritas y aprobadas por los responsables de las áreas involucradas.

Por otro lado, debe constituirse un equipo multidisciplinario como se plantea en el punto anterior, que disponga de los conocimientos necesarios de la nueva tecnología a aplicar.

2.3.4. Descripción de la interfaz del usuario en la aplicación Web

a) Análisis del modelo de interfaz del usuario

La interfaz del usuario en la tecnología Web es el “browser”, por lo que la misma debe contemplar, de acuerdo al usuario al que va dirigido, un diseño basado en la funcionalidad y en la navegabilidad.

Una ventaja significativa en la construcción de aplicaciones web, es que soporten las características de los browsers estándar, independientemente de la versión del sistema operativo instalado en el cliente. Esto significa que, en lugar de crear clientes para Windows, Mac OS X, GNU/Linux, entre otros, la aplicación es escrita una vez y es mostrada de la misma forma en todos los entornos desde donde se accede [Murugesan, 2001].

En este análisis se muestran las distintas pantallas que intervienen en la aplicación migrada, y se realiza un diagnóstico sobre su presentación y contenido.

b) Construcción del modelo

Se describe, utilizando el UML, la aplicación web centrada en el navegador (browser), que muestra la información de sitios a través de la administración de protocolos de comunicación, la interacción de páginas subdivididas en frames para facilitar los enlaces y, la administración de acceso a la base de datos a través de su interfaz [Pinheiro, 2000].

2.3.5. Revisión de la arquitectura física y del software de base

a) Arquitectura física

La arquitectura física del sistema migrado, basado en la tecnología web, se implementa generalmente en un modelo multicapa donde los usuarios remotos, a través de su terminal y del navegador se conectan a Internet a través de distintos tipos de conexión: RDSI, ADSL, Cable, Satélite, Redes inalámbricas, como se observa en la figura 2.

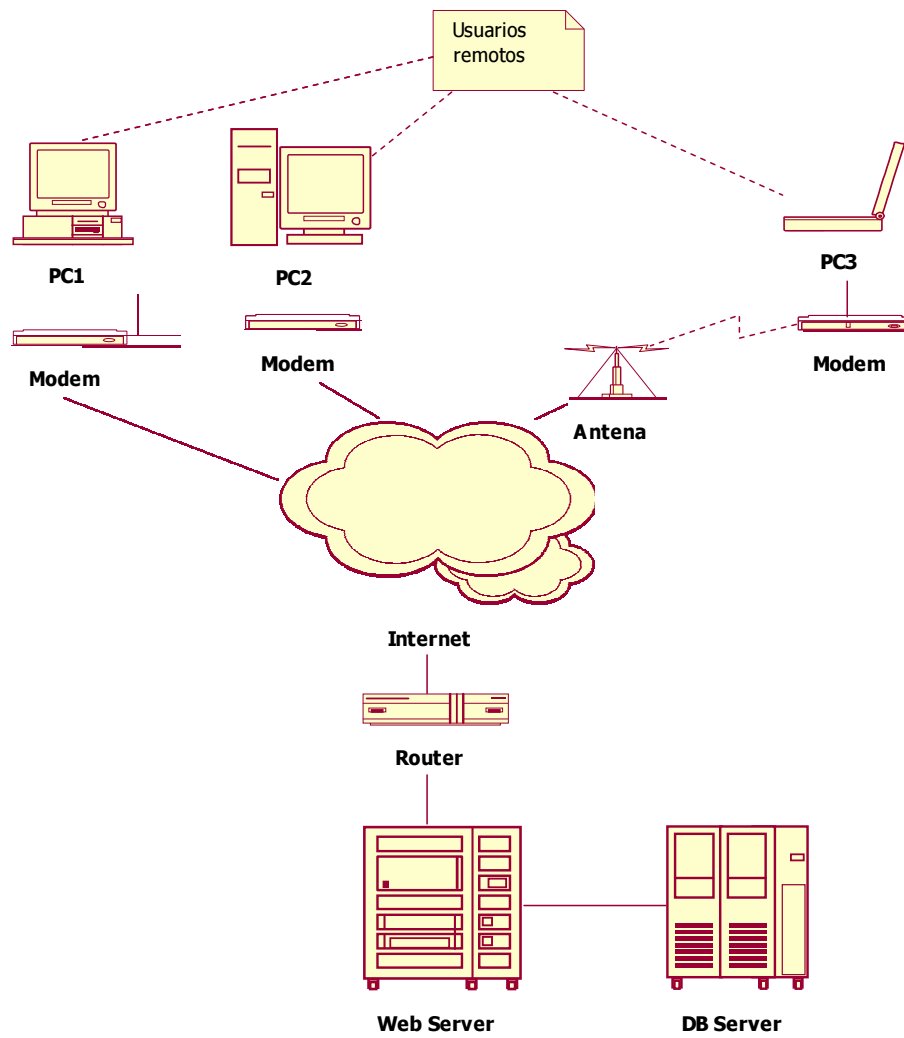


Figura 2 – Gráfico de componentes distribuidos de la aplicación Web

b) Arquitectura del Software de Base y diagrama de componentes

A continuación se presentan los distintos componentes que permiten realizar la visualización y transferencia de la información en un sistema distribuido en la Web, de acuerdo a la arquitectura física detallada en el punto anterior

Componentes:

- Sistema Operativo con interfaz gráfica.
- Conexión remota a servidores (local o de Internet).
- Browser de navegación.
- Componentes de la aplicación:
 - Páginas HTML.
 - Páginas para procesamiento de información (ASP, PHP, CLASS).
 - Scripts y archivos de código del lado del cliente (VBScript, JavaScript, JSP, etc.).
 - Plantillas de diseño (CSS).
- Componentes de comportamientos (OCX, DLL).
 - Manejo de imágenes (JPG, GIF, BMP, PNG).
 - Animaciones (AVI, SWF).
 - Sonido (WAV, MP3, etc.).
- Transferencia de información (XML, XSL, XLT, etc.).
 - Archivos de configuración y ocultos (CONFIG, WEBINFO).
 - Archivos inmodificables (PDF).
 - Facilidades de conexión a webcam.
 - Ensamblados directos a email.

El diagrama de componentes consiste en la descripción de la vista física de las tres capas de la aplicación Web: cliente, servidor web y servidor de base de datos, donde es posible observar la realización de los componentes de la aplicación GUI en la aplicación Web.