

La importancia de la estructuración de patrones de *Software* y buenas prácticas para la consecución exitosa de proyectos de desarrollo de soluciones informáticas integrales

Jaime Alberto Gutiérrez Mejía*

Fecha de recepción: 13 de septiembre de 2013

Fecha de aprobación: 20 de octubre de 2013

Pp.39 a 61

RESUMEN

El presente artículo desarrolla un tema de interés fundamental en el proceso de comprensión de las particularidades del proceso de construcción de aplicaciones, bajo el paradigma de los actuales *Frameworks* de desarrollo moderno de soluciones, principalmente en el ámbito de las aplicaciones *Web* y las aplicaciones para entornos de servidor empresarial corporativo.

Igualmente, postula algunas de las principales recomendaciones que deben ser tenidas en cuenta para estos procesos, haciendo un énfasis particular en la importancia de la separación modular por capas, el comprensión de los requisitos fundamentales de arquitectura de *Software* y la gestión integral de los recursos informáticos, en el ámbito de los procesos inherentes que las organizaciones del área tecnológica deben considerar para la certera construcción de aplicaciones, que cumplan con requisitos funcionales de calidad, y ofrezcan la mayor eficiencia en el momento de utilizar la infraestructura y el basamento necesario para su despliegue y ejecución.

PALABRAS CLAVE

Patrones de *Software*, buenas prácticas, soluciones informáticas.

* Ingeniero de sistemas y computación de la Universidad del Quindío. Candidato a Magíster en Administración de Tecnologías de la Información, del Instituto Tecnológico de Estudios Superiores de Monterrey, México.

The importance of software structural patterns and effective practices for the successful execution of developmental projects of integral informatics solutions

ABSTRACT

This paper describes a topic of key interest in understanding the fundamentals of the process of building up applications under the current paradigm of the current frameworks for the modern development of solutions, particularly in the area of Web applications and those applications used in corporate business servers. Besides, it mentions some key recommendations that should be taken into account for these processes, particularly emphasizing on the importance of modular separation of layers, the understanding of fundamental requirements of Software architecture and the integral management of computing resources, used in the area of those related processes that technological organizations should take into account for the accurate construction of applications which meet functional quality requirements, and which also offer wide efficiency in time when using infrastructure and all the necessary bases necessary for deployment and execution.

KEY WORDS

Software Patterns, Effective Practices, Informatics Solutions.

De l'importance de la structuration des modèles informatiques rendant possible la réussite de projets intégrant des solutions informatiques intégrales

RÉSUMÉ

Cet article expose un thème fondamental dans le processus de compréhension des particularités de la création d'applications informatiques qui utilisent comme paradigme les tendances actuelles de mise au point de solutions modernes, principalement dans le domaine des applications Web et des applications de serveur d'entreprise.

L'article relève certaines des principales recommandations devant être prises en compte lors de ces processus, mettant l'accent sur l'importance de la séparation des couches modulaires, la compréhension des exigences fondamentales de l'architecture logicielle et la gestion des ressources informatiques dans un environnement inhérents aux entreprises du secteur technologique qui doivent en outre envisager le développement d'applications répondant aux exigences de qualité fonctionnelle et offrant un rendement optimal lors de l'utilisation de l'infrastructure informatique.

Mots-clés

Modèles de Logiciels, Pratiques Adéquates, Solutions Technologiques.

Importância da estruturação de padrões de software e boas práticas para atingir projetos bem sucedidos de desenvolvimento de soluções informáticas integrais

RESUMO

Este artigo desenvolve um tema de interesse fundamental no processo de compreensão das particularidades do processo de construção de aplicações, segundo o paradigma dos atuais Frameworks de desenvolvimento moderno de soluções, principalmente no âmbito das aplicações Web e as aplicações para entornos de servidor empresarial corporativo. Igualmente, propõem-se algumas das principais recomendações que devem ser consideradas para estes processos, fazendo ênfase particular na importância da separação modular por camadas, a compreensão dos requisitos fundamentais de arquitetura de Software e a gestão integral dos recursos informáticos, no âmbito dos processos inerentes que as organizações da área tecnológica devem considerar para a construção certa de aplicações que cumpram com requisitos funcionais de qualidade, e ofereçam maior eficiência no momento de utilizar a infraestrutura e a fundação necessária para seu estabelecimento e execução.

PALAVRAS-CHAVE

Padrões De Software, Boas Práticas, Soluções Informáticas.

1. Introducción

Con preocupante interés se observa cómo las dinámicas de desarrollo tecnológico están comenzando a crear en las empresas de esta área y aquellas dedicadas a la construcción y comercialización de productos de *Software*, espacios favorables para la generación de malas prácticas que en lugar de propender por una construcción eficiente de las aplicaciones, producen retrasos significativos en la integración de las mismas. Adicionalmente, propician la estructuración de programas informáticos difíciles de mantener, llenos de deficiencias estructurales y altamente proclives a convertirse en cajas negras inmanejables e incapaces de brindar el soporte esperado por los usuarios finales, al momento de presentarse errores durante la fase de producción y soporte, que dichas aplicaciones proveen. Lo preocupante de este panorama, es que una mala formación en el conocimiento de las tecnologías necesarias para la integración y la construcción, es la que hace que muchos ingenieros, en su afán de sacar al entorno productivo las soluciones, realicen un uso indebido de los patrones de *Software*, fundamentales para una tarea de ingeniería certera y productiva.

Es crítico que solamente algunas empresas del sector tecnológico están comprendiendo que la organización funcional de procesos de desarrollo, la sólida base de la documentación y la existencia de recursos puntuales de organización, son el fundamento para lograr que el *Software* moderno logre cumplir con los altos estándares de calidad exigidos tanto por las organizaciones clientes de los productos, como aquellas dedicadas a la certificación de los mismos.

La existencia de metodologías fundamentales como CMM y CMMI , que estructuran de forma certera, la primera, los procesos organizacionales intrínsecos de las organizaciones y la segunda, los procesos necesarios para garantizar una continuidad de las labores inherentes a las fases de análisis, diseño, implementación, prueba y despliegue de aplicaciones de *Software*, hacen que muchas organizaciones dedicadas a esta temática, necesiten estructurar muy bien sus dinámicas de trabajo, buscando que

los procesos convencionales, algo anacrónicos de cascada o espiral, tiendan a transformarse en verdaderos esquemas de seguimiento iterativo e incremental, que logren el mejor proceso de desarrollo de las soluciones y busquen brindar a los ingenieros de los equipos de trabajo, herramientas verdaderamente efectivas para medir sus tiempos, proyectar sus capacidades, generar inteligencia agregada de negocio con sus conocimientos tecnológicos y dar a las organizaciones, certezas claras en las labores continuas y crecientes de creación de soluciones informáticas. Esto, aunado al hecho de que muchas empresas no están logrando posicionar realmente a gerentes de proyecto, con verdadero conocimiento de causa en el área tecnológica, funcional, operativa y sistémica del área informática y de desarrollo, produce a su vez negativos procesos de exitosa salida a producción de las soluciones, ya que muchas veces la coligación de temáticas de funcionalidad sistémica específica, con procesos de creación de las soluciones, genera que la incapacidad de comprender las ventajas y capacidades que las soluciones tecnológicas ofrecen, en el contexto de las dinámicas de negocio de cada empresa específica, llevan a ciclos repetitivos y sin fin en el ámbito de la iteración incremental del proceso constructivo y producen sinsabores fuertes entre clientes finales y desarrolladores al no entender con claridad la estructura misma de la complejidad del negocio y el impacto de las plataformas en la forma de sistematizar los procesos que quieren ser susceptibles de ser convertidos en herramientas funcionales de tipo informático y operacional.

Finalmente, la existencia de tantas y tan diversas tecnologías de todo tipo, tanto en el ámbito del *Software Open Source*, como el del de soporte licenciado, ofrecen una amalgama de paradigmas que llevan a que muchas veces los ingenieros encargados del proceso de construcción, se vean sometidos a paralizarse de manera específica con estas herramientas, cerrando su mente, criterio y capacidad de pensamiento, a entender otros modelos y usar la aplicación genérica de los patrones, independiente del lenguaje de programación específico, para realizar el montaje de las soluciones, haciendo de la labor ingenieril un ejercicio realmente valioso y no circunscrito a particularidades concretas de ciertos proveedores y expertos de conocimiento de una sola tecnología de tipo cerrado y obstructivo.

2. La importancia de la innovación y de la optimización

Dentro de la experiencia en temas de desarrollo y luego de haber sido testigo y actor directo de muchos procesos, tanto en empresas públicas como privadas, de las particularidades inherentes de cada uno de los diversos procesos de desarrollo de aplicaciones informáticas y de plataformas sistémicas de implementación, se puede mencionar algunas observaciones fundamentales en torno a la temática planteada por este artículo:

- El proceso de creación de aplicaciones, adolece en muchas ocasiones de una firme documentación en el ámbito de los casos de uso, los cuales son tomados a la ligera dentro del proceso de construcción de las aplicaciones y muchas veces, se convierte en una vacua plantilla para referenciar generalidades de las funcionalidades a construir, esto se considera un error, considerando que la dirección del proceso de construcción de *Software*, debe estar fundamentalmente guiada por una certera generación y conceptualización de dichos casos de uso.
- Las arquitecturas en muchas ocasiones son algo difusas, no centrandó su particular interés en generar componentes sino en crear únicamente piezas de interoperabilidad o mera integración. En muchas ocasiones, la creación de funciones concretas dentro de la dinámica implementativa de los casos de uso, se vuelve un ejercicio abstruso de combinaciones aleatorias de recursos tecnológicos de todo tipo (sentencias de *JavaScript*, rutinas de motor de base de datos, *Shells* independientes del sistema operativo, secuencias de comandos de otros lenguajes), haciendo que en muchas ocasiones, las aplicaciones se complejicen a un nivel innecesario.
- No se realiza en muchas ocasiones, una revisión adecuada de las dinámicas de construcción de componentes, haciendo que se generen funciones, objetos y recursos que debido al ineficiente

nivel de implementación y de configuración paramétrica, terminan volviéndose lentos e ineficientes para lograr el resultado. Se convierten así, en soluciones de paso para lograr dar solución a un problema, sin considerar el impacto mismo en la infraestructura y en los demás recursos sistémicos con los que dichos componentes deben interactuar.

- No obstante, también se visualiza un elemento muy concreto en el proceso y es la necesidad de lograr que el usuario final obtenga recursos de valor agregado en la ejecución de la funcionalidad específica, esto es, a partir de la interacción con las pantallas que implementan y tangibilizan el caso de uso de manera concreta, poder generar resultados eficientes y sobre todo capaces de ayudar al usuario a la ejecución de las tareas concretas de trabajo, dependientes de dicha funcionalidad (un ejemplo, lo constituye, la capacidad de multiexportación de reportes a formatos como PDF, XLS/XLSX, CSV o XML).
- Muchas veces, los ciclos de desarrollo se hacen lentos, debido a desconocimientos concretos de la tecnología final de implementación, producto de mala documentación, ejemplos inexistentes, fallas operativas de las librerías, o desconocimiento por parte de los actores del proceso de desarrollo (analistas, ingenieros e integradores) de aspectos fundamentales de la arquitectura de los *Frameworks* de desarrollo empresarial contemporáneos como JEE, .NET, SAP, entre otros.

Estas problemáticas, sumadas a la necesidad imperante de contar con una documentación pertinente y adecuada del mismo proceso de desarrollo, llevan a la necesidad concreta de redimensionar procesos fundamentales como la innovación y la optimización, y recursos de primera mano para el proceso mismo de estructuración de una nueva dinámica en el desarrollo de aplicaciones, por lo que sería ideal contar con mecanismos para subsanar estos temas, que incluyan las siguientes recomendaciones:

- Siempre entender bien el negocio, comprender los actores, el impacto, el modelo mismo de la organización, la cultura informática de los involucrados (*Stakeholders*) y a partir de un análisis profundo

de la organización, estructurar y determinar de manera coherente qué elementos son susceptibles de ser modelados como recursos propios del proceso de creación de aplicaciones, que impacten la dinámica de trabajo de dichos actores. Se hace importante comprender hasta los detalles más nimios, que en un momento u otro sean visibles a través de una pantalla del sistema o de un recurso de interacción del mismo con la información, los procesos u otros sistemas legados de la organización o de sus proveedores y *Partners* estratégicos.

- Desarrollar una orientación del proceso de construcción de las aplicaciones desde un punto de vista modular y estructurado, que facilite la comprensión en el que las capas del sistema deben estar orientadas a proporcionar servicios de datos concretos, independientemente de los recursos finales de consumo (sean estos tipo escritorio, *Web*, móvil, etc.); a partir de esa conceptualización, garantizar que las interfaces de los componentes son libres, acoplables con otros recursos y fácilmente administrables en su proceso de refactorización, sea que la causal esté relacionada con una corrección a un problema de operación del componente, o a la necesidad de mejorar lo actualmente implementado.

Figura 1. Ejemplo de una interfaz remota EJB necesaria para la creación de un componente lógico de negocios

```
import javax.ejb.*;
import java.rmi.*;

public interface Account extends EJBObject {

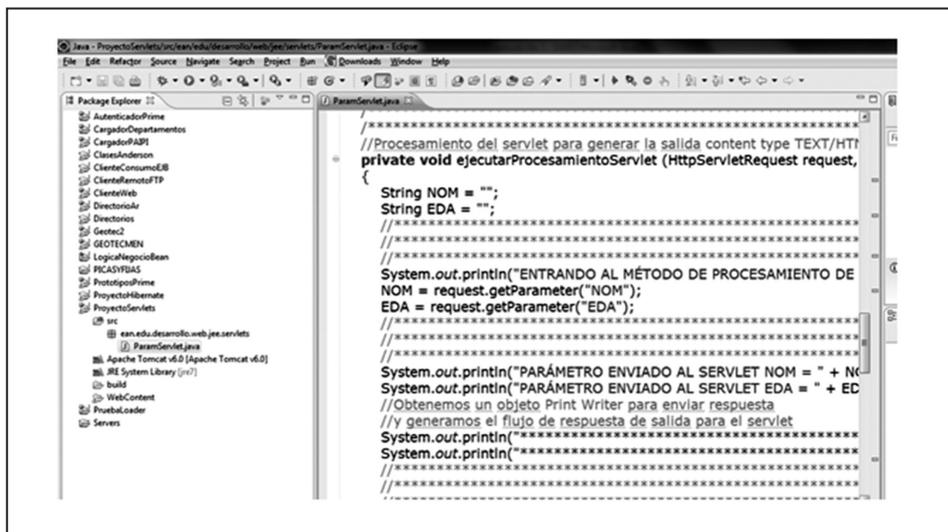
    void credit (double amount) throws RemoteException;
    void debit (double amount) throws RemoteException;
    double getBalance () throws RemoteException;
}
```

Fuente. Gsraj.tripod.com

- Tener claramente estructurado qué recursos de control se encargarán de interactuar con el modelo de negocio, esto es, establecer los controles lógicos y filtros necesarios para garantizar que los datos capturados desde las interfaces de entrada, vienen correctamente validados, no

representan amenazas de vulneración al procesamiento lógico ni a la integridad de los componentes de operación en la infraestructura, y que además dichos recursos se optimizan desde la perspectiva de su implementación en aspectos tales como el rendimiento, la multi-instanciación y la concurrencia simultánea de miles de usuarios en los escenarios de máximo estrés para el aplicativo; un ejemplo, lo constituye la forma como se construyen las variables de instancia y las variables locales a nivel de método en los servlets o en los managed beans con los que interactúan los componentes visuales en las aplicaciones *Web JEE*.

Figura 2. Ejemplo de implementación de Servlet Java en la creación de una aplicación Web JEE



Fuente. Elaboración propia del autor.

- Considerar la reducción significativa del uso de rutinas de procesamiento lógico en los componentes de presentación, circunscribiendo a lo meramente operativo en el ámbito de las validaciones y de las lógicas de remozamiento y despliegue de data procesada, los recursos de dichos recursos de presentación, y así garantizar que es posible utilizar posteriormente otros recursos gráficos para el despliegue de la data corporativa procesada por la solución.

Figura 3. Ejemplo de una implementación lógica de una *Managed Bean* para construir la lógica estructural de poblamiento de archivos para tabla dinámica página XHTML.

```

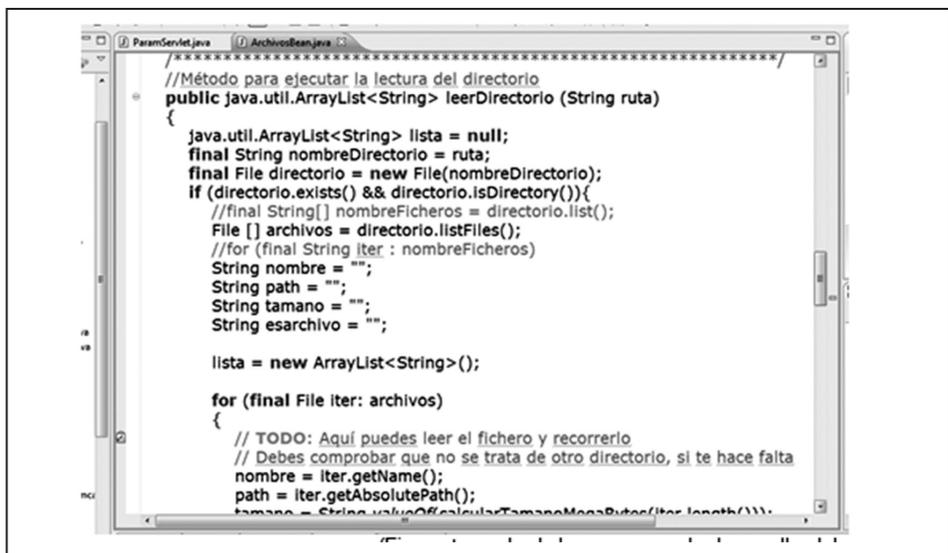
    poblarListaArchivos(archivosDirectorio);
}
/*****
/*****
/*****
//Método para actualizar la lista de la página dinámica
private void poblarListaArchivos(List<Archivo> list)
{
    java.util.ArrayList<String> listaArchivos = null;
    listaArchivos = leerDirectorio("E:\\Desarrollo\\Directorio");
    Line: 8

    for (int i = 0; i < listaArchivos.size(); i+=4)
    {
        Archivo archiv = new Archivo();
        archiv.setNombre((String)listaArchivos.get(i));
        archiv.setRutaArchivo((String)listaArchivos.get(i+1));
        archiv.setTamanoArchivo((String)listaArchivos.get(i+2));
        archiv.setEsArchivo((String)listaArchivos.get(i+3));
        archiv.setTipoArchivo(obtenerTipoArchivo(archiv.getNombre()));
        list.add(archiv);
    }
}
/*****
/*****
/*****

```

Fuente. Elaboración propia del autor.

Figura 4. Página XHTML



```

/*****
/*****
/*****
//Método para ejecutar la lectura del directorio
public java.util.ArrayList<String> leerDirectorio (String ruta)
{
    java.util.ArrayList<String> lista = null;
    final String nombreDirectorio = ruta;
    final File directorio = new File(nombreDirectorio);
    if (directorio.exists() && directorio.isDirectory()){
        //final String[] nombreFicheros = directorio.listFiles();
        File [] archivos = directorio.listFiles();
        //for (final String iter : nombreFicheros)
        String nombre = "";
        String path = "";
        String tamano = "";
        String esarchivo = "";

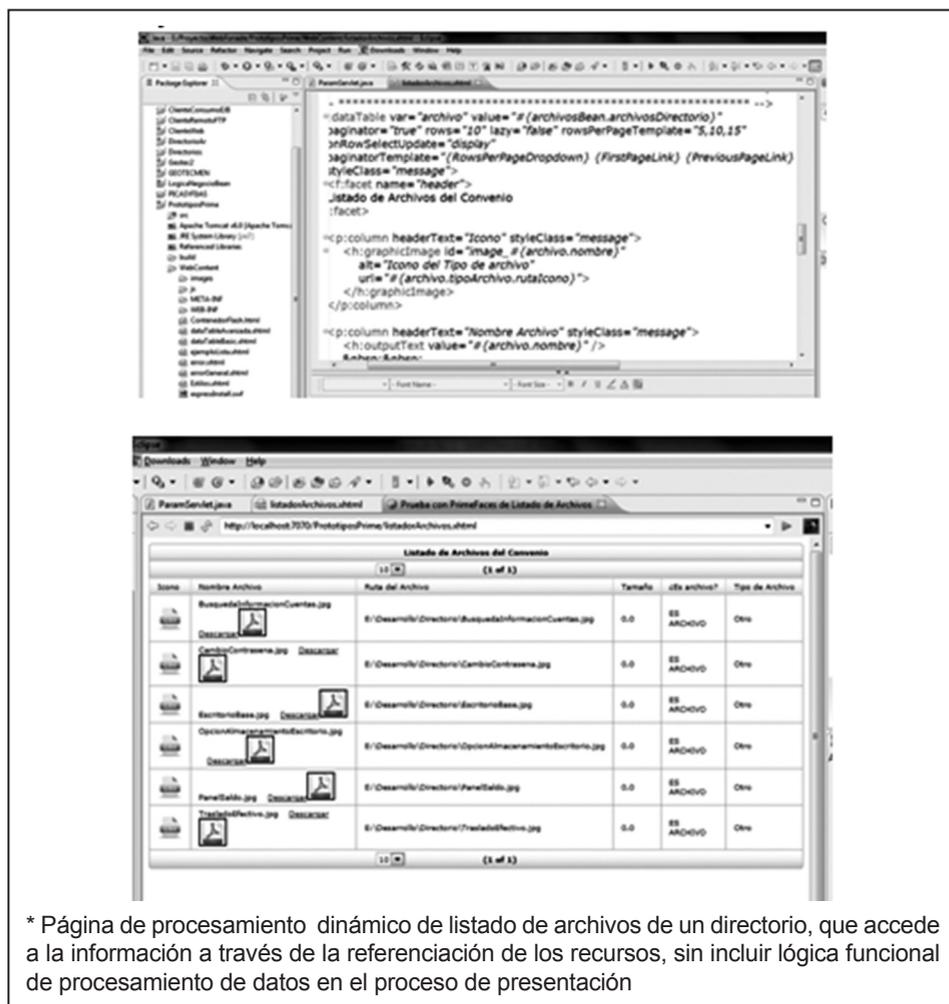
        lista = new ArrayList<String>();

        for (final File iter: archivos)
        {
            // TODO: Aquí puedes leer el fichero y recorrerlo
            // Debes comprobar que no se trata de otro directorio, si te hace falta
            nombre = iter.getName();
            path = iter.getAbsolutePath();
            tamano = String.valueOf(iter.length());

```

Fuente. Elaboración propia del autor.

Figura 5. Presentación en XHTML*.



Fuente. Elaboración propia del autor.

- Tener en cuenta las condiciones de seguridad, despliegue, acceso a recursos, permisos de disco, capacidades de ejecución, entorno operativo y características de la infraestructura computacional concreta donde se va a ejecutar el desarrollo, con el fin de que dichos procesos afecten de manera significativa la construcción de la aplicación, sino por el contrario, propendan por brindar los

recursos adecuados y pertinentes para el proceso de desarrollo en una dinámica que facilite la parametrización y el fácil despliegue, tanto en entornos integrales de pruebas, como de producción. Es importante comprender qué se puede y qué no se puede hacer a nivel tecnológico e implementativo, antes de postular la integración de recursos computacionales que probablemente entren en conflicto con el proceso de despliegue de la solución.

3. Consideraciones del proceso de *Software*

Se requiere por tanto, que al crear *Software*, no solo se haga uso de patrones estructurales acertados, sino que se apliquen buenas políticas de programación y definición en las aplicaciones, que garanticen que desarrollos subsiguientes, muy similares a aquellos que el modelo general ha planteado y que ha permitido construir, permitan a otras realizaciones llegar a resultados de calidad, en tiempos adecuados, buscando que los ingenieros puedan centrarse en la lógica funcional, brindando mayor nivel de concentración a otros niveles críticos de la solución. La innovación, por tanto, de la mano con el proceso de optimización, debe permitir lograr los siguientes resultados concretos:

- Ayudar a definir un modelo estructural de aplicación, basado en capas arquitectónicas claras, que orientadas al modelo de servicios, permitan definir procesos concretos, centrados en las interfaces y en los objetos de negocio, necesarios para la materialización de los resultados de procesamiento planteados en los casos de uso del sistema.

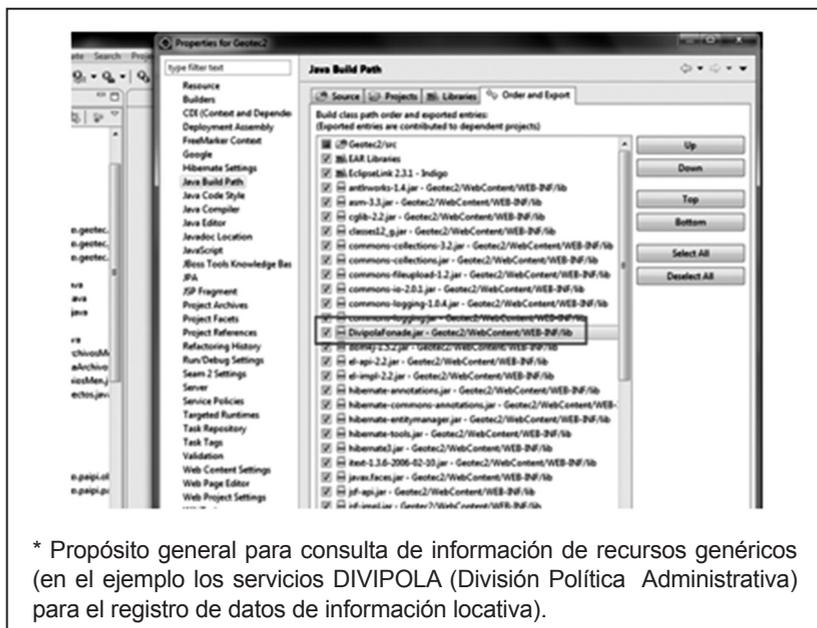
- Comprender que se requiere de un patrón concreto para cada caso de uso y que la buena estructura de los mecanismos del patrón, deben permitir que al implementarlo en múltiples funcionalidades de carácter similar, solo se hace necesario cambiar el *Core* de datos y la lógica de invocación a los recursos de procesamiento en la interfaz de negocio. Un ejemplo lo constituyen, las operaciones (CRUD) *Create*, *Retrive*, *Update* y *Delete* del sistema, donde si se construye, un EJB, un *Servlet* o un *Managed Bean* y una página XHTML que materialicen cada caso de uso, lo único que cambia entre funcionalidades es la estructura de cada formulario concreto a nivel de la presentación, la recopilación de los datos en el lado de la clase administrada en el Bean o el Servlet, y en la interfaz de negocio, la invocación del recurso concreto de operación sobre los datos del sistema (es decir, los métodos crearXXX, consultarXXX, modificarXXX o eliminarXXX, donde XXX representaría la entidad de negocio concreta sobre la que se esté trabajando en dicho momento).
- Valorar la importancia de las potencialidades de cada Framework de desarrollo, verificando en el modelo arquitectónico si es decir se hace más eficiente centrar la lógica en elementos estructurales del negocio mismo o en las potencialidades de otros recursos, como por ejemplo procedimientos almacenados, tareas transaccionales o sistemas de programabilidad extendida en los motores de persistencia del modelo mismo.

Igualmente, verificar dentro de cada *Framework*, cuáles elementos optimizan la presentación, con el fin de reducir el código fuente utilizado, determinar las mejores alternativas para el despliegue gráfico y garantizar un bajo acoplamiento que haga que los componentes puedan ser utilizados de forma indistinta en proyectos diferentes y garantizando el modelo orientado a servicios explicado anteriormente.

- Tratar de construir funciones comunes y genéricas de la aplicación en librerías especializadas que puedan ser importadas de forma tal, que cuando se requiera acceder a un recurso de uso común para muchas aplicaciones, no se incurra en tener que volver a reimplementar nuevamente las funcionalidades (lo que se señala como reinventar la rueda) para hacer que dichos elementos estén disponibles desde el

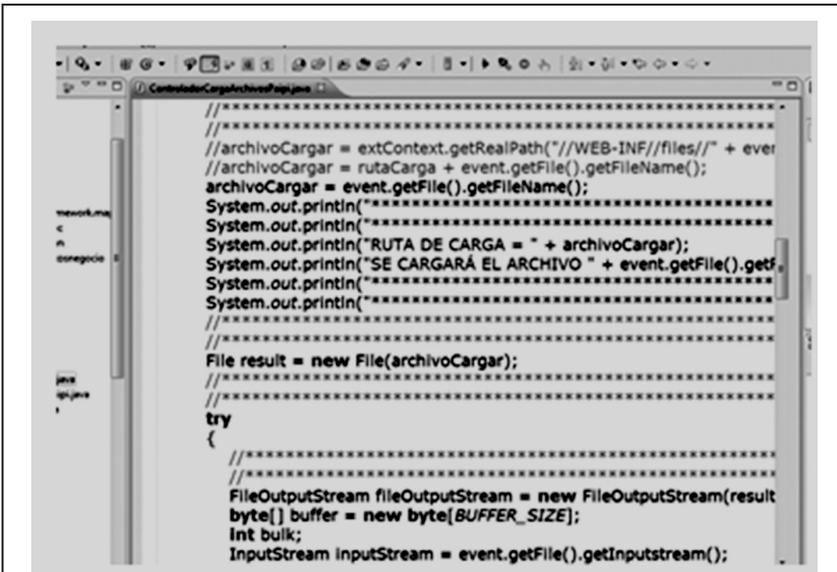
- modelo integral de acceso a datos de toda la organización. Un ejemplo de esto, lo constituyen los famosos mecanismos de consulta, anidada de datos para pantallas como lo son la correlación municipio, departamento (entidad territorial) y país, que muchas pantallas exigen para el registro de información locativa de una entidad concreta de información.
- Considerar que las API de acceso a otros recursos, deben usarse de forma eficiente, siendo enmascaradas por nuevas librerías de propósito general, que permitan acceder a funcionalidades complejas de datos tales como consulta de imágenes, inserción de datos de tipo BLOB en bases de datos, acceso a la descarga de información de ubicaciones en disco, consulta de directorios de *Path* físico, acceso a cargadores de datos, exportación de documentos, transformación de data para recursos de ofimática, entre otros y que en la mayoría de casos de uso, son requeridas para tareas complejas de procesamiento de información.

Figura 6. Importancia de un producto de desarrollo de una librería*.



Fuente. Elaboración propia del autor.

Figura 7. Implementación de librerías para el acceso a recursos de complejidad.*



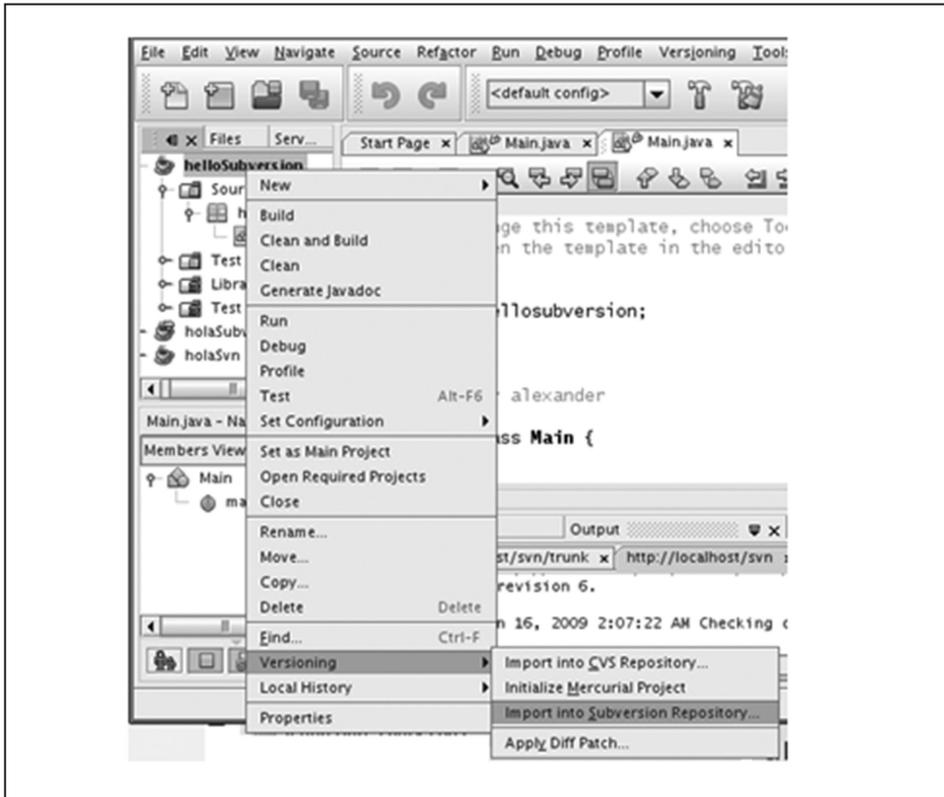
```
//*****  
//archivoCargar = extContext.getRealPath("//WEB-INF//files//") + event  
//archivoCargar = rutaCarga + event.getFile().getFileName();  
archivoCargar = event.getFile().getFileName();  
System.out.println("*****");  
System.out.println("*****");  
System.out.println("RUTA DE CARGA = " + archivoCargar);  
System.out.println("SE CARGARÁ EL ARCHIVO " + event.getFile().getF  
System.out.println("*****");  
System.out.println("*****");  
//*****  
//*****  
File result = new File(archivoCargar);  
//*****  
//*****  
try  
{  
//*****  
//*****  
FileOutputStream fileOutputStream = new FileOutputStream(result  
byte[] buffer = new byte[BUFFER_SIZE];  
int bulk;  
InputStream inputStream = event.getFile().getInputStream();
```

* Ambito del manejo de imágenes y control de carga de recursos para el sistema

Fuente. Elaboración propia del autor.

- Estructurar de forma adecuada la documentación de la API de las librerías, con el fin de garantizar que al ser utilizados dichos recursos por otros desarrolladores del equipo de integración, sean claras las interfaces y se tenga un acceso pertinente al proceso de consumo de servicios, haciendo que la transparencia de uso de dichos recursos sea alta y garantice una alta cohesión de los componentes a la hora de realizar el proceso de integración de las soluciones. Por otro lado, esto no solo debe ser replicable al ambiente de desarrollo, sino que también debe ser visible en el momento de la ejecución del sistema, cuando las interfaces de consumo sean correctamente invocadas y en caso de excepcionamiento por ausencia de las mismas o desconexión de los recursos de red, la aplicación pueda garantizar el despliegue del error y el manejo administrado e inteligente del mismo.

Figura 8. Sistema de control del versionaje para el entorno de desarrollo.



Fuente. Diagonalasterisco.blogspot.com.(2009)

- Finalmente, llevar un proceso de versionamiento pertinente y controlado, que haciendo uso de herramientas de automatización y centrado de información en un repositorio común y colectivo de control de versiones, facilite la generación de las versiones del aplicativo, su correspondiente documentación y que facilite el reversamiento de las mismas, en caso de fallas operacionales presentadas en el proceso de pruebas y despliegue. Esto, con el fin no solo de crear el recurso de control de dichas versiones, sino de garantizar que la automatización se hace eficientemente, que se entregan liberaciones de los prototipos del producto en los tiempos coherentes y que las funcionalidades de los casos de uso, corresponden a lo que los binarios ejecutables del proceso de ensamblaje, efectivamente poseen a partir del desarrollo producto del proceso iterativo.

- Estos elementos son fundamentales, pero no solo por la importancia que reviste el proceso mismo, sino porque debe generar una cultura organizacional adecuada, que garantice que todos los actores del proceso de desarrollo, trabajan de manera coordinada en la construcción de los recursos, que entienden la dinámica del negocio y orientan sus esfuerzos efectivos a crear *Software* de calidad, usando patrones coherentes, prácticas estructurales de mercado probadas y efectivas y generar a partir de su trabajo, un mecanismo eficiente de resolución de las problemáticas concretas de procesamiento de información de los usuarios finales.

No basta solo con documentar y conocer bien como construir un EJB, un *Servlet*, una DLL o un *Stored Procedure*. Se requiere tener una conciencia clara de hacia donde se quiere llegar con el desarrollo, del alcance (*Scope*) claro de los proyectos y de su contextualización en el ámbito del usuario final y de las soluciones que se quieren lograr, y que a partir de una visión clara del negocio y de sus procesos susceptibles a ser convertidos en funcionalidades sistémicas concretas, de como dar respuesta oportuna a la solución integral informática a construir.

- Se necesita adicionalmente, el compromiso firme de las organizaciones, tanto desarrolladoras como consumidoras de *Software*, de comprender de forma clara cómo orientar el proceso de desarrollo no solo hacia las buenas prácticas informáticas, sino a lograr que la coherencia de conceptos, la definición clara del glosario de negocio y la estructuración de los procesos funcionales de operación de la organización, sean insumos pertinentes para la creación de las aplicaciones. Se necesita inteligencia, disposición, orden, documentación, criterio y conocimiento de causa, para encauzar dichos proyectos a la satisfacción de funcionalidades de usuario final, pero no como el resultado del capricho de unos pocos, sino como una forma clara de obtener resultados, a todas luces, necesarios para que la empresa entienda que el *Software* construido para ella, debe ser un aliado estratégico en su encaminamiento objetivo en el ámbito de las TIC y como una forma de potencializar los conocimientos, el caudal humano y las intencionalidades concretas de hacer de la empresa un recurso fundamental para el devenir organizacional y el enfoque objetivo de las tareas misionales de todos sus actores.

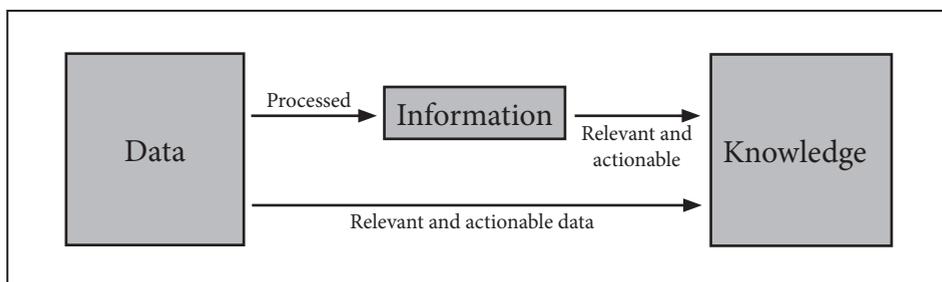
4. La importancia de la administración del conocimiento como herramienta en la evolución de los sistemas de información

Un elemento clave que hay que diferenciar, es que el conocimiento no es lo mismo que los datos y la información. El conocimiento es información que está contextualizada y es relevante, pero lo más importante, es que se convierte en herramienta estratégica para la toma de decisiones que debe ser compartido de forma eficiente al interior de la organización para hacer a cada empleado y cada miembro de la empresa, una parte activa en el proceso de consolidación y formación de dicho capital de conocimiento. Por ello, una adecuada implementación de un sistema de manejo de información de administración de conocimiento (KM), ha de contar con herramientas para comunicar, generar esquemas de colaboración y almacenar persistentemente el conocimiento para su posterior utilización.

Las organizaciones muchas veces, consideran que solo basta con tener un adecuado sistema de información para lograr sus propósitos. No obstante, es fundamental diferenciar que el activo de capital más valioso de toda empresa, reside, no en sus datos corporativos, sino en la forma como su talento humano y el potencial de desarrollo de sus procesos son capaces de lograr las metas de productividad esperadas y necesarias para hacer de la empresa un referente social y cultural, con impacto de profunda relevancia en su entorno de construcción y evolución. El conocimiento es una acumulación de capitales intelectuales, procedimentales, culturales, económicos y sociales, que durante las fases de evolución de la empresa se convierten más que en datos almacenados en forma de documentos, en verdaderos elementos de constitución de la construcción de los procesos de la organización y en los pilares axiológicos de definición de su identidad corporativa y real cultura organizacional.

La relevancia del conocimiento se da cuando los datos al pasar por el tamiz de la experiencia, la percepción racional de los actores involucrados en su transformación y la generación de patrones comportamentales derivados de su estudio, permite que los datos corporativos se transformen en herramientas de peso para la toma de decisiones estratégicas en la organización y en fuentes de análisis que permitan tomar buenas prácticas para evitar los errores que hacen fracasar el desarrollo y la industria modernas (figura 9).

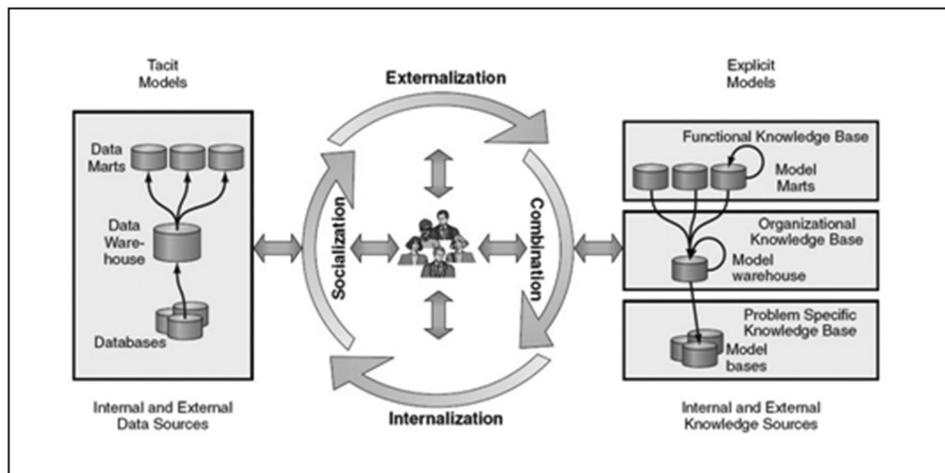
Figura 9. Proceso de transformación de la información en conocimiento relevante y accionable



Fuente. Turban (2007).

Una gran ventaja que posee la moderna administración del conocimiento, es la existencia de *Datamarts* (es decir repositorios de datos específicos a la funcionalidad lógica de negocio, de cada dependencia de la organización), que combinadas por medio del uso de algoritmos de minería y análisis de información, así como por la existencia de programas capaces de generar patrones de inferencia, permiten combinar la inteligencia estratégica de la empresa de forma eficiente ayudando a tomar decisiones y visualizando la prospección futura de situaciones que basadas en los datos actuales, ayuden a evitar futuras dificultades y problemas generados por los elementos afines de patrones repetitivos en la ocurrencia de los datos críticos del negocio (figura 10).

Figura 10. Marco de trabajo para la integración de sistemas de administración del conocimiento y sistemas de soporte decisional



Fuente. Turban (2007).

Producir conocimiento debe ser inherente de las organizaciones, no una necesidad, puesto que en las empresas puede aplicarse la máxima que dice, “quien no conoce la historia, está condenado a repetirla”. Con las modernas herramientas de análisis de datos y minería de información que muchos proveedores como *Oracle*, *Microsoft* e *IBM* actualmente ofrecen para el desarrollo de procesos de inteligencia de negocio, sería impensable no tener mecanismos que hagan más eficiente el desarrollo de la gestión del conocimiento y lo más importante, que permitan explotar el capital intelectual de las personas que con su diaria labor, contextualizan los esquemas de calidad que actualmente son fundamentales para el buen devenir de las organizaciones.

Es importante tener en cuenta la esencia del trabajo de los empleados corporativos en la dinámica de administración de conocimiento, la que se reduce a “establecer prioridades estratégicas para el KM, establecer un repositorio de conocimiento de las mejores prácticas, obtener un compromiso serio los ejecutivos senior en el soporte a los ambientes de aprendizaje, enseñar a los buscadores de información a hacer las más intuitivas y mejores preguntas, establecer los procesos para manejar los capitales intelectuales, obtener información de satisfacción del cliente en

tiempo real y globalizar la administración del conocimiento”(Turban,2008)¹, todo ello para lograr metas estratégicas, incrementar el *Balanced Score Card* de la organización y definir nuevos procesos de calidad que hagan mucho más eficiente el diario día a día de la empresa, en aras de obtener resultados más positivos, ganancias superiores al promedio y generar un posicionamiento de la organización que la haga atractiva, dinámica y siempre acorde con las necesidades del momento, en conjunción dinámica con la evolución del mundo y las tecnologías de información.

¹ Para ampliar esta información consultar el capítulo del libro *information technology for management: transforming organizations in the Digital Economy*.

5. Referencias bibliográficas

Revista Transferencia. Instituto Tecnológico de Estudios Superiores de Monterrey, México. Grupos de Sistemas Inteligentes como soporte a la decisión. Recuperado de: [http://biblioteca.itesm.mx/cgi-bin/doctec/opendoc?cual=869&archivo=13968&pagina=2&paginas=2&query=\(sistemas AND, informacion,AND,estrategica\),AND,tipo%3Ds](http://biblioteca.itesm.mx/cgi-bin/doctec/opendoc?cual=869&archivo=13968&pagina=2&paginas=2&query=(sistemas AND, informacion,AND,estrategica),AND,tipo%3Ds)

Revista Integratec. Instituto Tecnológico de Estudios Superiores de Monterrey. México. Base de Datos Proquest. Universidad virtual: el reto de incorporar la realidad al aprendizaje. Disponible en: [http://biblioteca.itesm.mx/cgi-bin/doctec/opendoc?cual=1249&archivo=22290&pagina=2&paginas=2&query=\(sistemas,AND,informacion,AND,estrategica\),AND,tipo%3Dsopendoc?cual=869&archivo=13968&pagina=2&paginas=2&query=\(sistemas,AND,informacion,AND,estrategica\),AND,tipo%3Ds](http://biblioteca.itesm.mx/cgi-bin/doctec/opendoc?cual=1249&archivo=22290&pagina=2&paginas=2&query=(sistemas,AND,informacion,AND,estrategica),AND,tipo%3Dsopendoc?cual=869&archivo=13968&pagina=2&paginas=2&query=(sistemas,AND,informacion,AND,estrategica),AND,tipo%3Ds)

Turban, E., Leidner, D., McLean, E. Wetherbe J. (2008). *Information Technology for Management: Transforming Organizations in the Digital Economy*. Editorial John Wiley & Sons, Inc.