

Una Propuesta de Metodología para la Migración de Sistemas Heredados

A proposal for a legacy systems migration methodology

Daniel Torres Silva, Juan Diego Ortiz Galván, Héctor Andrade Gómez, Rafael Rivera López* 

Departamento de Sistemas y Computación

Instituto Tecnológico de Veracruz

Calzada Miguel Ángel de Quevedo 2779 Col. Formando Hogar, Veracruz, Ver. México C.P. 91860

*rrivera@itver.edu.mx

Área de participación: Ingeniería en Sistemas Computacionales

PALABRAS CLAVE:

Análisis Modular, Procesos de Negocio, Métodos Ágiles, Interoperabilidad de Sistemas.

RESUMEN

En este artículo se describe una Metodología Ágil de Desarrollo de Software Incremental e Iterativa para la migración de Sistemas Heredados (MADIISH) que permitirá organizar el proceso de migración por fases desde un análisis modular del sistema heredado a migrar, el establecimiento de procesos de negocio e implementación de modelos de datos, así como la definición de etapas de pruebas piloto y la posibilidad de generar módulos que permitan mantener la interoperabilidad entre los nuevos sistemas desarrollados y los sistemas activos. MADIISH se aplica para la migración de sistemas obsoletos que pueden ser de misión crítica, alta disponibilidad, de gran flujo de datos o para sistemas orientados a la planificación de recursos empresariales (Enterprise Resource Planning, ERP).

KEYWORDS:

Modular Analysis, Business Process, Agile Development, Systems Interoperability.

ABSTRACT

This paper describes an iterative, incremental and agile software process for legacy systems migration (MADIISH), which allows to organize the migration process in phases, including a modular analysis of the legacy system, the establishment of the business processes that drive the migration, implementation of data models, as well as the definition of stages of initial tests and the possibility of generating modules that allow the interoperability between the new modules and the active (old) modules. MADIISH can be applied to the migration of obsolete systems that can be mission critical, or system that require high availability and a large data flow. MADIISH is especially suited for the migration of Enterprise Resource Planning (ERPs).

Recibido: 28 de junio del 2017 • Aceptado: 5 de enero del 2018 • Publicado en línea: 28 de agosto del 2018

1. INTRODUCCIÓN

Se denominan sistemas heredados a aquellos sistemas informáticos que han permanecido en funcionamiento durante un largo periodo de tiempo dentro de una empresa o conjunto de organizaciones, que han evolucionado bajo las características de las tecnologías en el que fueron desarrollados y que al pasar del tiempo llegan a sus límites [1]. Es en este caso que la organización debe decidir si mantener el funcionamiento del sistema, o realizar una migración hacia el uso de nuevas tecnologías. Las causas más frecuentes para decidir la migración de un sistema se debe a la incompatibilidad con nuevas arquitecturas de hardware, la imposibilidad de crecimiento y que el soporte técnico es muy limitado o el personal conocedor de la tecnología heredada no está disponible, con lo que se incrementan los costos del mantenimiento, se disminuye la capacidad de escalabilidad y se reducen los índices de competitividad en el mercado ([1], [2], [3]). A los problemas anteriores se adiciona en muchas ocasiones el aislamiento de los orígenes de datos (si se carece de un gestor de datos relacional), lo cual puede provocar inconsistencias en los resultados de búsquedas, redundancia y deficiencia en la confiabilidad de los datos [4]. Durante la migración de un sistema heredado, frecuentemente no se cuenta con manuales o esquemas de la arquitectura del sistema, o algún modelo que permita visualizar el flujo de los procesos para comprender el negocio, lo que dificulta el proceso de migración, recurriendo a periodos de análisis donde se deben definir los procesos y diseñar los esquemas de datos [1] [2]. Durante este proceso es necesaria la elaboración de módulos interoperables entre el nuevo sistema y el sistema heredado para permitir el flujo de datos mientras se realiza la migración [3].

2. METODOLOGÍAS DE SOFTWARE

Una metodología de desarrollo de software consiste en hacer uso de herramientas, métodos, modelos para el desarrollo y técnicas para estructurar, planear y controlar el proceso de desarrollo de sistemas de información [5]. La gran mayoría de las metodologías tradicionales tales como cascada, prototipos, incremental y espiral suelen ser demasiado documentadas, esto para que los programadores que estarán dentro de la planeación del proyecto puedan entender perfectamente la metodología y en algunos casos el proceso de negocios

, razón por la cual hacen pesada cualquier migración. Estas metodologías suelen consistir en:

1. Realizar un análisis de los requisitos: Consiste en documentar lo que el software deberá realizar al término de su migración.
2. Diseño del sistema y programa: Es la realización de un prototipo y los algoritmos a utilizar sin codificar.
3. Codificación: Se realiza la escritura del código necesario para el desarrollo del software.
4. Ejecución de pruebas: Para valorar y localizar posibles errores o validaciones que no hayan sido consideradas. De igual forma permitirá conocer los tiempos de ejecución y la veracidad de los resultados obtenidos.
5. Verificación: Una vez que fue probado por el programador, se instalará para que el usuario realice pruebas reales.
6. Mantenimiento del sistema nuevo: Normalmente no se prueban todos los posibles casos por lo que siempre habrá que corregir errores y realizar actualizaciones.
7. Amplia documentación en todo momento.

En cambio, las metodologías ágiles como lo es Scrum y Kanban [6] consisten principalmente en:

1. Trabajar para obtener software funcional en lugar de demasiada documentación.
2. Colaboración con el usuario para la comprensión rápida de sus procesos de negocios.
3. Se tiene la posibilidad de hacer cambios de planes en cualquier punto del proyecto evitando la planeación extensa, lo que permite iniciar la programación.

Sin embargo, aunque sus ventajas son interesantes, estas metodologías también presentan inconvenientes que hay que asumir cuando se decide trabajar con ellas. Estas son:

1. Falta de documentación del diseño.
2. Problemas de comunicación debido a la participación del usuario donde se pueden dar malas interpretaciones en lo hablado y no documentado.
3. Existe una fuerte dependencia de las personas, si el usuario no tiene el tiempo suficiente disponible puede alentar el desarrollo del proyecto.
4. Falta de reusabilidad derivada de la falta de

documentación.

Estos inconvenientes son contemplados y existe un enfoque especial a los mismos dentro de la metodología de migración MADIISH.

3. METODOLOGÍAS PARA LA MIGRACIÓN DE SISTEMAS HEREDADOS.

En la actualidad se han diseñado varias metodologías para la migración de un Sistema Heredado (SH), tal es el caso de la metodología de apoyo basada en el uso de las herramientas KDD (Knowledge Discovery in Databases) donde se señala que uno de los factores de éxito de un proyecto de migración es el entendimiento del SH, esto es, entender tanto el modelo de datos como el modelo de negocios que trata de cubrir el mismo [7]. Mediante esta estrategia se pretende reconstruir algunos aspectos básicos del SH a migrar, de modo que sea posible entender el modelo de datos del SH, entender el modelo de negocios que intentaba cubrir el SH y determinar el nivel de calidad de los datos del SH.

Esta metodología propone apoyar la recuperación de requisitos de un SH, basada en el uso de herramientas de minería de datos; dichas herramientas se basan en un esquema a lo cual se identifica la estructura y el objetivo de la realización de la minería de datos, la cual pretende "excavar" entre los datos para hallar información oculta y que posiblemente sea de gran utilidad en la toma de decisiones [7] [8]. La técnica de la minería de datos se basa principalmente [9] en los siguientes incisos:

- Selección de los datos.
- Preprocesamiento o enriquecimiento de datos.
- Transformación o codificación de datos.
- Extracción del conocimiento (minería de datos).
- Evaluación de los resultados

La propuesta menciona que la importancia de la minería de datos en el desarrollo de la metodología basada en herramientas KDD tiene como fundamento principal que los datos siempre estarán presentes y esa es la característica que ha sido explotada en dicha investigación, por lo que se menciona en la propuesta misma: si se posee los datos, entonces que estos sean la fuente que provea de conocimiento sobre el SH [7]. Por lo que el punto es el cómo se puede extraer el conocimiento sobre el SH que supone está implícito en los datos, cuyo objetivo es, justamente, encontrar co-

nocimiento oculto entre los datos; según lo menciona su autor. Las etapas que comprende la propuesta de la metodología anterior son las siguientes:

- o Etapa 1: Obtención de los datos del SH y la documentación asociada a ellos.
- o Etapa 2: Selección de una herramienta de KDD.
- o Etapa 3: Compatibilización del formato de los archivos o tablas con las entradas para la herramienta KDD y selección de atributos.
- o Etapa 4: Análisis de los datos utilizando funciones de preprocesamiento.
- o Etapa 5: Minado de los datos.
- o Etapa 6: Chequeo de los resultados con el usuario.
- o Etapa 7: Recomposición del modelo de datos del SH.

Dichas etapas generan salida de información que determinará el proceso de la etapa inmediata; a su vez, continuamente se evalúan los resultados para verificar su integridad y que exista confiabilidad de los mismos.

Por otro lado, como parte de los trabajos de migración de sistemas heredados y las nuevas tendencias de desarrollo de software orientado a arquitecturas basadas en la nube (Cloud Computing), se propuso un esquema particularmente para la migración de sistemas hacia la nube. El autor menciona que, en el flujo de trabajo propuesto, no se considera el análisis de seguridad como una tarea independiente, si no integrada a cada una de las tareas involucradas a la migración. Es por esto por lo que el consumidor tiene la responsabilidad de alinear cada actividad de la migración a sus políticas de seguridad y asegurarse que el contrato del proveedor abarque estas políticas [3].

El esquema propuesto, Figura 1, incluye 13 procesos que deben ser llevados a cabo por el consumidor, proveedor y desarrollador del servicio.

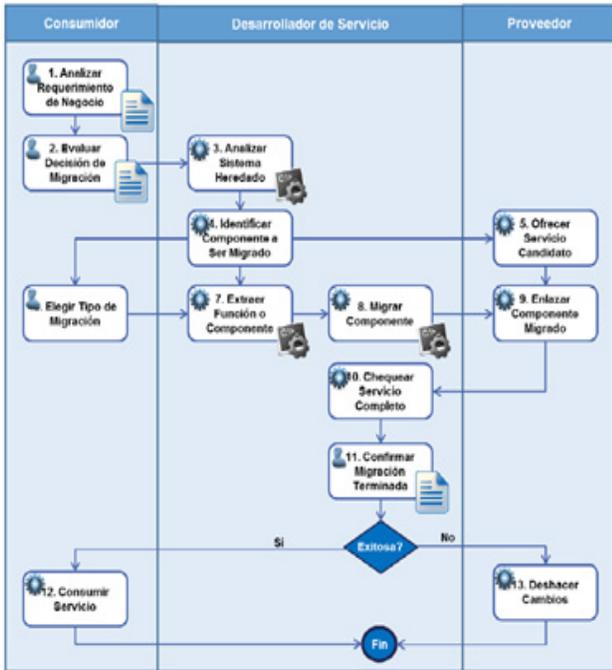


Figura 1.- Esquema de proceso de migración de sistemas heredados hacia Cloud Computing [3].

Las propuestas de las metodologías anteriores y la experiencia obtenida durante la migración de un sistema heredado han formado parte de los fundamentos del desarrollo de la metodología MADIISH.

4. METODOLOGÍA MADIISH

La metodología MADIISH está basada en las principales características de las siguientes metodologías:

1. Metodología iterativa debido a que se pueden realizar modificaciones constantes sobre un mismo subsistema hasta la entrega satisfactoria al usuario final.
2. Metodología incremental ya que se migra subsistema por subsistema.
3. Metodología ágil por la participación constante del cliente para reducir la documentación.

La metodología MADIISH presenta una estructura como se muestra en la Figura 2:



Figura 2.- Estructura general de la metodología MADIISH

La cual consta de las siguientes siete fases:

- Fase C: Se define de forma abstracta la conformación del supra sistema en sus diferentes sistemas, así como la de sus sistemas en sub sistemas, lo cual permite establecer la relación entre cada sub sistema para facilitar la determinación de la jerarquía de migración y la interoperabilidad de dicho sub sistema o sistema con otros. Esta fase se denomina fase cero debido a que durante la migración se realiza una vez, al inicio de la migración. A su vez, permite el reconocimiento de las necesidades del cliente y se plantean las estrategias necesarias para poder llevar a cabo el proceso de migración.
 - Fase 1: Se elige un sistema a migrar tomando en cuenta los siguientes criterios:
 - o Importancia
 - o Sencillez
 - o Conveniencia
 - o Complejidad
 - o Contingencia
 - o Selectividad

A su vez, se realiza un análisis de entradas y salidas de datos para determinar la dependencia de otros sub sistemas. De igual forma se definen los nuevos requerimientos de usuario y se comprende mediante el análisis de procesos de negocio la operación del sistema, esquematizando el conocimiento sobre un diagrama de entradas y salidas de datos, permitiendo la eficiencia de los nuevos procesos.

- Fase 2: Se realiza el modelado la base de datos o actualización de la misma en caso de existir una previa de algún subsistema migrado anteriormente,

el desarrollo de interfaces de usuario, la programación para la comunicación entre interfaces con base de datos y la generación de una versión prototipo para realizar pruebas y mostrar al usuario.

- Fase 3: Se implementa un ambiente de pruebas lo más semejante al ambiente de producción en donde se pueda ejecutar un prototipo con información actualizada, si son los resultados esperados se podrá continuar con la siguiente fase, de no serlo se deberá regresar al análisis y esquematización de entradas y salidas de datos. Es aquí otra de las características de MADIISH, permite regresar dentro del ciclo de migración a un punto en el que se pueda determinar las fallas presentadas según el avance obtenido.

- Fase 4: Una vez superadas las fases anteriores es necesario implementar el nuevo sistema en el ambiente de producción por lo que se debe anular el funcionamiento del subsistema heredado migrado, actualizar la base de datos del nuevo sistema e implementar finalmente el nuevo sistema. Se tiene contemplado la habilitación de módulos de interoperabilidad de datos en caso de que la desconexión afecte al flujo de datos entre sistemas.

- Fase 5: Se realiza una documentación del sub sistema o sistema migrado según sea el caso en la cual se describen los procesos de negocios, la documentación del código fuente y la descripción grafica del sistema.

- Fase T: Elaboración de la documentación final, consiste en unir toda la documentación y retroalimentar los manuales o procesos en los que se encuentre ambiguo su contenido.

De acuerdo con la estructura y el flujo de trabajo que conforma MADIISH, Tabla 1, se generó un análisis de las ventajas, desventajas y las semejanzas que tiene esta metodología contra las propuestas de los artículos descritos anteriormente:

Tabla 1.- Análisis de ventajas, desventajas y similitudes de MADIISH con otras propuestas metodológicas.

Actividad	MADIISH	Metodología de migración de sistemas heredados basada en KDD	Metodología de migración de sistemas heredados hacia Cloud Computing
Estructura iterativa e incremental	Es por naturaleza incremental e iterativa en cada una de sus fases	Los procesos son secuenciales, al final de las pruebas pueden regresar a la primera fase.	Los procesos son secuenciales, al final de las pruebas pueden regresar a la primera fase.
Procesos ágiles	La organización del trabajo y la entrega continua de resultados vuelve ágil los procesos	No se menciona una entrega continua de resultados al cliente.	El cliente puede probar el funcionamiento de los servicios a través del consumo de los mismos.

Estudio del caso y obtención de datos del sistema heredado	Fase C, la cual permite obtener información acerca del sistema heredado y plantear las estrategias de trabajo para la migración	Corresponde a la fase 1, donde se obtienen los datos del sistema heredado y se analiza la documentación del mismo	Corresponde al consumidor (cliente) en el cual se plantea un análisis inicial de requerimientos del negocio para poder evaluar una decisión de la estrategia de migración.
Definición de módulos que integran el sistema a migrar	Fase C en la cual se identifican cada módulo y composición del sistema heredado.	Integra esta tarea durante el estudio del caso y obtención de datos del sistema heredado.	Durante el desarrollo del servicio se contempla un análisis del sistema heredado de manera general y se definen los componentes del sistema
Proceso de selección de módulo a migrar	Fase 1, con el apoyo de una tabla de criterios, el cliente tomará una decisión.	No existe una tabla de criterios. Se realiza la migración de acuerdo con la decisión del cliente.	No existe una tabla de criterios. Se identifica el componente a ser migrado durante el desarrollo del servicio.
Análisis de procesos de negocio y proceso de comprensión de funciones	Fase 1, se determinan las entradas y salidas de datos en donde se definen los procesos de negocio; se realiza un análisis de requerimientos y se plantean mejoras de los mismos si se requiere, comprendiendo así las funciones del negocio.	A través de la Fase 2, es seleccionada una herramienta KDD para poder realizar el análisis de procesos de negocio a través de los datos que brinda el sistema heredado.	Durante el desarrollo del servicio, se extraen las funciones de los componentes seleccionados, para ser analizados y verificar el flujo de datos que conforma el componente. No se contempla la comprensión de funciones como una característica de esta metodología.
Modelado, diseño de interfaces y generación de prototipos funcionales	Fase 2, se modelan las bases de datos, si existiese un modelo base, se tomaría como referencia. Se generan las interfaces y se construyen prototipos funcionales.	En las fases 3 y 4 se realizan labores de determinación de las tablas que van a participar en la prueba de generación de conocimiento, así como el análisis de los datos que serán sometidos a la prueba; pudiendo determinar la generación de interfaces de usuario y programación de acuerdo con la lógica de los datos.	Existe una etapa de migración de componentes durante el desarrollo del servicio, se enlazan los servicios y se verifica que el mismo esté disponible para generar pruebas funcionales con el consumidor.
Ambiente de pruebas y generación de resultados	Fase 3, se programa un ambiente de pruebas similar a la de producción, se involucran los usuarios y se muestran los resultados obtenidos para su análisis.	Durante la fase 5 se ejecuta el minado de datos y se extraen los resultados; por lo que en su fase 6 se realiza la verificación de los mismos con el usuario; así como también en su fase 7, en base a los resultados obtenidos, se realiza la recomposición del modelo de datos del SH	El consumidor realiza pruebas del servicio y en base a los resultados obtenidos se determina si la prueba fue exitosa o no, para mantener o deshacer cambios.

Implementación e interoperabilidad	Fase 4 en donde se realiza la instalación del subsistema migrado una vez superadas las pruebas; en caso de que se necesite mantener el flujo de datos entre el sistema nuevo con el heredado, se realizan instalaciones de módulos que mantendrán la interoperabilidad de datos hasta que se complete la migración.	No se contempla una fase de instalación o generación de módulos de interoperabilidad de datos entre sistemas.	No se contempla una fase de instalación o generación de módulos de interoperabilidad de datos entre sistemas.
Continuidad de la migración y generación de documentación	Durante la fase 5 se analiza el próximo subsistema a migrar y se generan los documentos finales como manuales de usuario y técnicos.	La generación de documentación final no se menciona en esta metodología.	La generación de documentación final no se menciona en esta metodología.

5. CASO DE ESTUDIO

La metodología MADIISH fue aplicada en la migración de un supra sistema (Sistema Integral de Control) de una empresa mexicana, la cual se conformaba en diferentes sistemas y subsistemas que fueron determinados en una reunión con el personal involucrado y el equipo de migración tomando los criterios establecidos en esta metodología. A partir de los cuales, por el nivel de importancia para los altos mandos de la empresa deciden que se diera inicio en el sistema de Almacén iniciando por el subsistema de Compras, el cual es de alta prioridad debido a que su funcionalidad es proveer el control del inventario de su almacén, el cual presentaba irregularidades en sus resultados y constantes fallas durante su operación en tiempo de producción (cierres inesperados, lentitud, inconsistencia entre la base de datos y lo que muestra en pantalla), lo que limitaba la usabilidad con el usuario final y lo volvía un subsistema inestable.

El proceso de migración en base a la metodología MADIISH (Tabla 2), presenta un proceso iterativo en cada una de sus fases por lo que es fácil notar el incremento en módulos realizados por cada iteración desde la fase 1 a la 5, lo que permitió una constante comunicación con el usuario mostrando cada avance y funcionamiento del nuevo sistema. El usuario durante esta migración está consciente de que la migración del sistema conllevaba un cambio en la forma de utilizar las nuevas funciones y que debe familiarizarse con la nueva tecnología. Unas de las resistencias al cambio durante este proyecto fueron las siguientes:

- El usuario está acostumbrado a la ubicación de los campos del sistema heredado y la forma en cómo debe capturar la información, por lo que en el nuevo

sistema se deberá presentar la distribución de campos de una manera tal que el usuario se familiarice rápidamente con la captura de información.

- El uso del ratón o mouse para navegar por el sistema, ya que en el sistema heredado toda la navegación era realizada mediante el uso del teclado sin la posibilidad del uso del ratón, ahora deberán hacer uso de él en algunas opciones del nuevo sistema.

- El temor de que los datos proporcionados por el nuevo sistema estén incorrectos debido a la depuración de registros solicitado (eliminar duplicidad de registros, asignación de claves de producto, completar información de inventario que era necesaria para realizar cálculos, entre otras sugerencias).

- La posibilidad de que ciertas validaciones especiales que fueron programadas en el sistema heredado dejen de funcionar debido a la migración, esto sucede a menudo cuando se desarrollan funciones muy específicas solicitadas alguna vez por los usuarios y que en el nuevo sistema no sean consideradas debido a la falta de fuentes de información o manuales del sistema heredado.

- La adquisición de infraestructura nueva es una de las resistencias del usuario en este proyecto, ya que se debió adquirir nuevas impresoras láser para la emisión de reportes (el sistema heredado las emitía por impresoras de matriz) y la adquisición de mayor espacio en memoria del servidor.

- Para el personal de sistemas implicó la capacitación del área de desarrollo y la contratación de personal capacitado para el uso de tecnologías nuevas.

Durante el desarrollo del proyecto, se trató de involucrar al personal operativo y de área para que en cada avance se mostraran las nuevas funciones del sistema y se tuviera la oportunidad de dialogar, admitir sugerencias y mejoras para que el usuario se fuera familiarizando y depositase mayor confianza al nuevo sistema; así como la necesidad de diseñar dos módulos que permitieran la interoperabilidad entre los sistemas heredados con el nuevo sistema (Figura 3) durante el transcurso de la migración hasta que los subsistemas dependientes del nuevo sistema se migraran y así se permitiera la desactivación de dichos módulos de interoperabilidad. Estos módulos fueron necesarios para permitir al usuario final la comparación de resultados entre el sistema heredado y el nuevo sistema a partir de cargar la misma información de entrada en ambos subsistemas, logrando obtener los resultados esperados libre de errores y dudas. Esto fue de suma importancia

para desactivar el subsistema heredado y entrar en completa operación con el nuevo sistema.



Figura 3.- Definición modular del sistema de Almacén

Tabla 2.- Proceso de migración del Sistema de Compras aplicando MADIISH.

Fases	Actividades
C: Análisis modular	Se determinó la interacción que tiene el subsistema de compras en conjunto con los subsistemas contiguos como son Notas de Entrada a Almacén y Resurtidos, Figura 3. Las relaciones de estos subsistemas componen el sistema de Almacén.
1: Análisis de procesos de negocio	Se identificaron las entradas y salidas cada uno de sus procesos, definiendo a través de ellos un proceso de negocio que ayudó a comprender de manera lógica el flujo de los datos y la manera en que se procesarán las peticiones del usuario.
2: Modelado, programación y generación del prototipo	Se diseñaron y elaboraron los modelos de datos tomando como referencia los archivos de datos DBF con los que el sistema heredado interactúa. Se generaron las interfaces de usuario y se implementó la lógica del proceso de negocio obtenida en la fase 1; es posible durante las iteraciones que se localicen módulos internos como el proceso de compras en tránsito (Figura 4).
3: Pruebas y resultados	Se contempló un periodo de pruebas en la cual se entregan prototipos del proyecto, así como la demostración de los objetivos alcanzados y la ejecución de pruebas piloto para observar los resultados obtenidos con respecto al proceso de compra del sistema heredado.
4: Implementación	En esta fase se integraron los módulos de compras en tránsito y los módulos como productos en promoción y productos especiales; los módulos interoperables permitieron obtener los datos de los archivos DBF para la alimentación temporal del nuevo sistema (Figura 5).
5: Migración continua y documentación	El subsistema de compras está dividido en pequeños módulos, por lo que, por cada iteración de la metodología, se incrementan sus funciones y continua el proceso de migración hasta abarcar todos los módulos disponibles dentro del subsistema de compras.
T: Elaboración de documentación final	Se prepararon los manuales y documentos definitivos, así como la entrega de los procesos de negocios generados durante la migración.

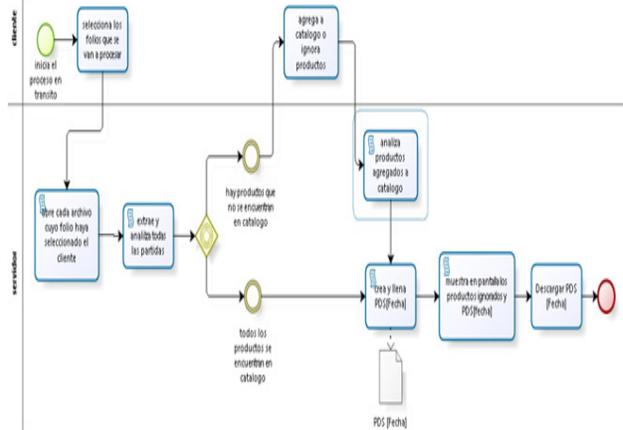


Figura 4.- Proceso de negocio del módulo de compras en tránsito, perteneciente al sistema de almacén

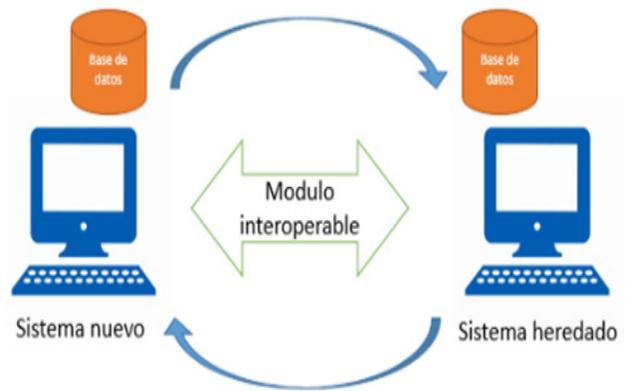


Figura 5.- El módulo interoperable provee flujo de datos desde el sistema de compras al sistema nuevo y viceversa.

6. CONCLUSIONES

Como resultado de la implementación de MADIISH se pudo constatar que la clasificación por sistemas, subsistemas, módulos y su conjunto de funciones permite obtener un amplio conocimiento de la estructura del sistema heredado, así como la determinación de sus procesos de negocio y la definición hacia nuevos modelos de datos a través de este conocimiento; logrando así el desarrollo de nuevos sistemas con mayor eficiencia en sus procesos y con mejor aprovechamiento de sus recursos. Esta dinámica ágil de desarrollo incremental e iterativa permite la generación de prototipos funcionales y periódicos de los módulos que se estén migrando, así como la posibilidad de la construcción de módulos interoperables que garantizan la coexistencia de los datos entre ambos sistemas mientras se realizan los trabajos de migración y no detener el flujo del negocio, lo que genera un valor agregado para la empresa permitiendo involucrarse paulatinamente a las nuevas operaciones de sistema migrado, reduciendo la resistencia a los cambios e incrementando las probabilidades de éxito de la migración.

REFERENCIAS

- [1] Henrard, J., Hainaut, J., L., Cleve, A., Hick, J., M. Migration of legacy information systems, 2008.
- [2] Bisbal, J., Legacy Information Systems, issues and directions, 1999, 6(1), 103-111.
- [3] Zalazar, A., S., Migración de Sistemas Heredados a Cloud Computing, Argentine Symposium on Software Engineering, ASSE, 2014
- [4] Bradley, R., Moving from DBF to SQL Server, 2006, Broad Leal LLC.
- [5] Menendez, R., Barzanallana, A., Ingeniería del software: Metodologías de desarrollo, Informática Aplicada a la Gestión Pública. Recuperado el día 02, 06, 2017 de <http://www.um.es/docencia/barzana/IAGP/IAGP2-Metodologias-de-desarrollo.html>, 2011.
- [6] Sommerville, I., Ingeniería de Software, 2005, 7.
- [7] Caro, G., A., Bocca, J., Campos, D., Migración de Sistemas Heredados: Una metodología de apoyo basada en el uso de herramientas KDD (Knowledge Discovery in Databases), Revista ingeniería de Sistemas, 2002, 16(1), 51-60.
- [8] Imielinski, T., Swami, A., Agrawal R., Data Mining: A Performance perspective, IEEE Transactions on Knowledge and Data Engineering, 1993, 5(6).
- [9] Adriaans, P., Zantinge, D. Data Mining. 1996.
- [10] Barros, O. Reingeniería de procesos de negocio. 1994, Dolmen

SEMBLANZA



Daniel Torres Silva es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Veracruz (ITVER 2015) y Técnico en Informática por el Centro de Estudios Tecnológicos Industrial y de Servicios número 15 en Veracruz. Se desempeña como desarrollador y analista de software en la empresa Sistemas para Agentes Aduanales S.C. en Veracruz;

cuenta con certificación internacional de desarrollo de software basado en el conocimiento como Analista bajo la herramienta GeneXus, Actualmente aplica la metodología MADIISH para la migración de los sistemas de la empresa. Sus áreas de interés son: Lenguajes de programación, Ingeniería de software, Arquitectura de Computadoras, Arquitecturas Orientadas a Servicios (SOA) y Modelado de bases de datos.



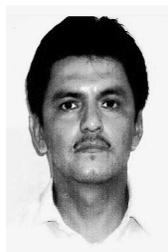
Juan Diego Ortiz Galván es técnico en sistemas computacionales por el Colegio Práctico de Computación Actualizada, A.C. (COPCA) e ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Veracruz (ITVER 2015). Actualmente es encargado de programación en el departamento de informática de la

empresa Rullán del Sur S.A. de C.V. ubicada en la ciudad de Veracruz dónde realiza diferentes actividades como lo es soporte a usuarios, análisis y modelado de procesos de negocios e implementación de la metodología MADIISH para la migración interna de la compañía. Sus intereses incluyen ingeniería de software, programación, base de datos, cómputo distribuido, redes e innovaciones tecnológicas.



Héctor Andrade Gómez es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Veracruz (ITVER 1985), Maestro en Ciencias Computacionales por el Instituto Tecnológico y de Estudios Superiores de Monterrey campus Morelos (1992) y Doctor en Ciencias Computacionales por la Universidad de Florida (2001). Ha trabajado en desarrollo de

software, administración de centros de cómputo y también en el área de soporte técnico. Ha sido profesor de planta del Instituto Tecnológico y de Estudios Superiores de Monterrey, campus Puebla. Actualmente es profesor investigador en el Departamento de Sistemas y Computación del Instituto Tecnológico de Veracruz. Sus áreas de interés incluyen: Lenguajes de Programación, Cómputo Móvil, Arquitecturas Orientadas a Servicios y Desarrollo Web.



Rafael Rivera López es Ingeniero en Sistemas Computacionales por el Instituto Tecnológico de Veracruz (ITVER 1989) y es Maestro en Ciencias de la Computación por el Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de Morelos (2000). Actualmente desempeña su labor como profesor e investigador en el Departamento de Computación y Sistemas

en las instalaciones del Instituto Tecnológico de Veracruz. Sus áreas de estudio y de interés incluyen la programación orientada a objetos (POO) y la aplicación de técnicas de optimización extraídas de la inteligencia artificial (IA), como son los algoritmos de cómputo evolutivo y de inteligencia colectiva.