

# Kiosco electrónico de servicios automatizados para trabajadores

## Desarrollo de un prototipo funcional aplicando el proceso unificado

MSC. Sofía Margarita Manríquez Orduño  
Departamento de Sistemas y Computación  
Instituto Tecnológico de La Paz  
La Paz, B.C.S., México  
sofia.manriquez@itlp.edu.mx

Dr. Marco Antonio Castro Liera  
División de Estudios de Posgrado e Investigación  
Instituto Tecnológico de La Paz  
La Paz, B.C.S., México  
mcastro@itlp.edu.mx

**Resumen**—En este artículo se presenta el desarrollo de un prototipo funcional de un kiosco electrónico de servicios automatizados para los trabajadores del Instituto Tecnológico de La Paz siguiendo el marco de trabajo establecido por el Proceso Unificado. En el Instituto existen diversos procesos administrativos que requieren que el trabajador presente documentos oficiales, estos documentos eran generados manualmente y frecuentemente presentaban errores de redacción, de veracidad en la información o bien no eran oportunos. Para dar solución a lo anterior se desarrolló un kiosco electrónico en el que es posible generar de forma inmediata, sin errores y en una primera versión, nueve documentos de índole administrativo y que en un futuro puede implementarse en el resto de los institutos tecnológicos distribuidos a lo largo y ancho de la república mexicana. Además se presenta una propuesta técnica y económica del diseño físico del kiosco electrónico.

**Palabras claves**—*kiosco electrónico; proceso unificado; procesos administrativos; marco de trabajo; prototipo funcional*

### I. INTRODUCCIÓN

La sociedad de la información es una nueva forma de organización social que capacita a la ciudadanía para acceder y generar información, además de poder compartirla y obtener nuevo conocimiento desde cualquier lugar y en cualquier momento[4].

Esta nueva sociedad plantea retos que modifican la manera de entender las relaciones personales y profesionales, las actividades culturales, los trámites de gobierno, la comunicación y el entretenimiento. Ello implica también la aparición de nuevos modelos de negocios centrados en el conocimiento y en la innovación que favorecerán la competitividad de un país.

En México para dar impulso a esta nueva sociedad, los tres niveles de gobierno han establecido acciones que responden a esta nueva forma de organización social, un ejemplo de ello son los llamados “Kioscos electrónicos”, los cuales proporcionan mecanismos ágiles de consulta de información sobre trámites gubernamentales. La aplicación de los kioscos electrónicos se ha diversificado a muchas áreas de la sociedad actual[4]. El Instituto Tecnológico de La Paz no es la excepción y en este trabajo se describen las actividades llevadas a cabo para el desarrollo del kiosco electrónico.

Una de las primeras actividades que se llevó a cabo fue la identificación de un proceso de desarrollo adecuado para este tipo de proyecto, adoptando al proceso unificado como proceso de desarrollo[2] y al lenguaje unificado de modelado[3] como técnica de modelado para construir los planos del software.

A través del kiosco electrónico un trabajador puede generar un total de nueve documentos de los cuales cuatro provienen del departamento de recursos humanos y cinco del departamento académico correspondiente. Para identificar el grupo de documentos a implementar en esta primera versión se diseñó y aplicó una encuesta a 60 trabajadores de los diferentes departamentos que conforman la estructura orgánica del Instituto Tecnológico de La Paz.

La puesta en marcha del kiosco electrónico elimina la problemática inherente a la obtención de documentos administrativos, ya que estos son generados de forma automática e inmediata, sin errores en su contenido y de manera oportuna. Además de que dignifica al usuario del kiosco otorgándole servicios de primer nivel y sin depender de la voluntad de un servidor público.

Además del desarrollo del software se presenta una propuesta técnica y económica del diseño físico del kiosco, convirtiendo a este proyecto en una alternativa factible de ser implementado en el resto de los Institutos Tecnológicos que conforman al Instituto Nacional de México.

### II. METODOLOGÍA

#### A. El proceso unificado

Es una metodología orientada a objetos de tercera generación, que se caracteriza por integrar varios métodos. Es el comienzo de la unificación de los métodos de Booch, Rumbaugh y Jacobson en una metodología que establece un marco de trabajo de procesos que guía las actividades para el desarrollo de sistemas orientados a objetos de distintos tamaños y complejidad. Es un ejemplo de un modelo de proceso de desarrollo moderno e híbrido, ya que reúne elementos de todos los procesos genéricos, iteraciones de apoyo y buenas prácticas en la especificación y el diseño[1].

Los aspectos claves que definen al proceso unificado son:

- a) Está dirigido por casos de uso,
- b) Está centrado en la arquitectura y
- c) Es iterativo e incremental.

El proceso unificado está dirigido por casos de uso ya que el proceso de desarrollo sigue un hilo que avanza a través de una serie de flujos que parten de los mismos, esto es, los casos de uso se especifican, se diseñan, se implementan y se prueban. El modelo de casos de uso es la base para que el desarrollador cree una serie de modelos de diseño e implementación. Los desarrolladores revisan los modelos sucesivos para que sean conforme a los casos de uso. Los ingenieros de prueba validan la implementación para garantizar que se implementan correctamente los casos de uso.

Que el proceso unificado esté centrado en la arquitectura significa tener una descripción del sistema en construcción desde diferentes vistas que contengan los aspectos dinámicos y estáticos del mismo. La arquitectura representa el diseño de la forma que tendrá el sistema atendiendo a los casos de uso y sus posibles evoluciones. La arquitectura inicia con los casos de uso más importantes (representan el 5%-10% del total), por lo tanto, el arquitecto de software debe crear primero una arquitectura no específica de los casos de uso, por ejemplo plataforma en cuanto a hardware, sistema operativo, manejador de base de datos, protocolos de comunicación, bloques de construcción reutilizables; segundo, especificar en detalle un subconjunto de casos de uso en términos de subsistemas, clases y componentes; tercero tener una arquitectura estable a través de la maduración de los casos de uso.

El proceso unificado es iterativo e incremental, esto proporciona la estrategia para desarrollar un producto en pasos pequeños manejables:

- Planificar un poco,
- Especificar, diseñar e implementar un poco e
- Integrar, probar y ejecutar un poco cada iteración.

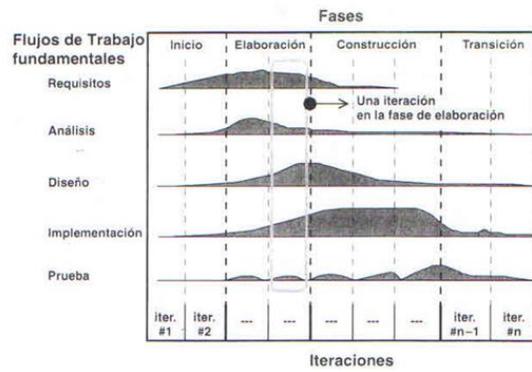
Es útil dividir el trabajo en miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones son pasos en el flujo de trabajo, y los incrementos son crecimiento del producto. Para una efectividad máxima, las iteraciones deber ser controladas; esto es, deben seleccionarse y ejecutarse de una forma planificada[2].

La Fig. 1 muestra en la columna de la izquierda los flujos de trabajo –requisitos, análisis, diseño, implementación y prueba-. Las curvas son una aproximación de hasta donde se llevan a cabo los flujos de trabajo en cada fase. Cada fase se divide en iteraciones o mini-proyectos. Una iteración típica pasa por los cinco flujos de trabajo.

Fig. 1. Flujos de trabajo del proceso unificado.

### B. Lenguaje unificado de modelado (UML, Unified Modeling Language)

Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema, proporciona una



forma estándar de describir los planos de un sistema, cubriendo tanto las cosas conceptuales tales como procesos del negocio y funciones del sistema, como las cosas concretas, tales como las clases escritas en un lenguaje de programación específico, esquemas de bases de datos y componentes software reutilizables.

El lenguaje unificado de modelado es un lenguaje muy expresivo, que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas. Por ser un lenguaje, es tan sólo una parte de un método de desarrollo de software. UML es independiente del proceso, aunque para utilizarlo óptimamente se debería usar en un proceso que fuese dirigido por casos de uso, centrado en la arquitectura, iterativo e incremental [3].

## III. DESARROLLO

Los flujos de trabajo que definen al proceso unificado y los que se siguieron durante el desarrollo del proyecto son los siguientes:

### A. Captura de requisitos.

Este flujo de trabajo permitió identificar las necesidades del usuario y describir el sistema de información objetivo de una manera clara y precisa para que el cliente y los desarrolladores lleguen a un acuerdo sobre lo que debe y no debe hacer el sistema.

El resultado de este flujo de trabajo fue el modelo de requisitos. La Fig. 2 resume los elementos que lo conforman.

### B. Análisis.

Este flujo de trabajo permitió resolver los requisitos formulados en un lenguaje más preciso, añadir detalles que son esenciales para los desarrolladores y definir la arquitectura de clases base.

El resultado de este flujo es el modelo de análisis. La Fig. 3 sintetiza los elementos que lo conforman.

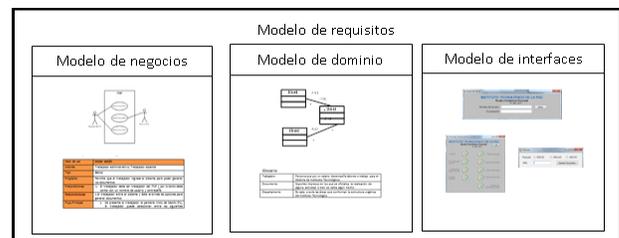


Fig. 2. Componentes del modelo de requisitos.

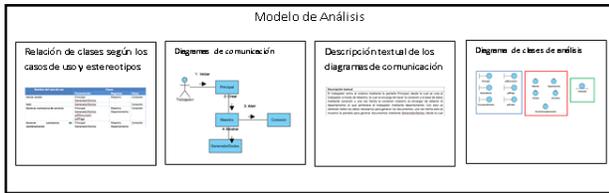


Fig. 3. Componentes del modelo de análisis.

C. Diseño.

El modelo de análisis no es lo suficientemente formal para generar código fuente, por esta razón es necesario elaborar el modelo de diseño. En este flujo se refinó el modelo de análisis hasta que el material tomara una forma que los programadores puedan implementar. Un aspecto importante de este flujo fue:

- Incremento de las operaciones y clases apropiadas.
- Selección del lenguaje de programación.
- Reutilización de componentes.
- Decidir los componentes de hardware y software necesarios para la ejecución de la aplicación.

La Fig. 4 muestra el modelo de diseño resultante de este flujo de trabajo.

D. Implementación.

En este flujo de trabajo se llevó a cabo la construcción del sistema bajo el entorno de desarrollo integrado Microsoft Visual Studio 2010 Ultimate y C# como lenguaje de programación. Para la creación de los documentos en formato PDF se usó la librería SharpPDF.

La Fig. 5 modela el subsistema de implementación y en él se muestra como los elementos de diseño como las clases se implementaron en términos de componentes para la capa de presentación, capa de negocios y capa de datos.

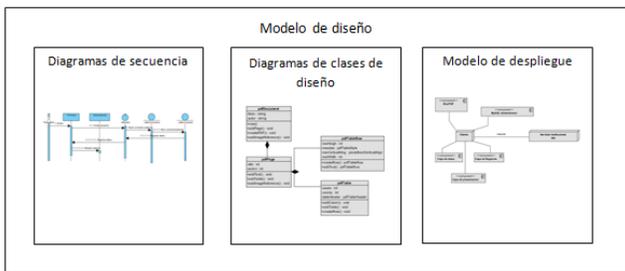


Fig. 4. Modelo de diseño

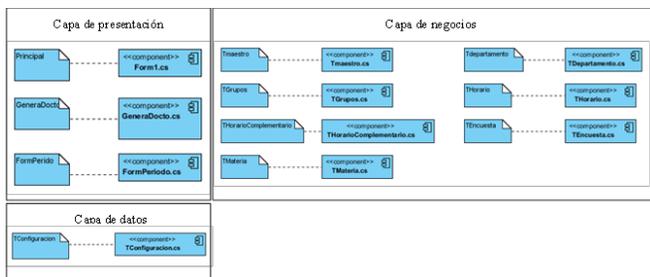


Fig. 5. Subsistema de implementación

E. Prueba.

Este flujo de trabajo sirvió para descubrir y corregir los errores cometidos durante el diseño y la implementación. Como estrategia de prueba se llevaron a cabo pruebas de unidad, pruebas de integración y pruebas del sistema. Las pruebas de especificación y las de estructura fueron las técnicas de pruebas utilizadas.

IV. RESULTADOS

La Fig. 6 y Fig. 7 muestran el resultado del prototipo funcional del kiosko electrónico de servicios para los trabajadores del Instituto Tecnológico de La Paz, en el que es posible generar nuevos documentos administrativos de interés para los trabajadores, logrando con ello un proceso oportuno ya que, se reduce el tiempo de espera para la obtención de los documentos y se incrementa la confiabilidad y veracidad de la información contenida en los documentos. También se presentó la propuesta técnica y económica del diseño físico del kiosko.



Fig. 6. Pantalla para generar documentos

PERIODO	NOM	MATERIA	GRUPO	EST	EST
Ene-Jan	1	ADMINISTRACION DE LA FUNCION INFORMATICA	A	LI	4
Ene-Jan	2	FUNDAMENTO DE PROGRAMACION	A	OC	10
Ene-Jan	3	PROGRAMACION	B	OC	10
Abr-Jun	4	FUNDAMENTO DE PROGRAMACION	A	OC	10
Abr-Jun	5	FUNDAMENTO DE PROGRAMACION	C	OC	10
Abr-Jun	6	FUNDAMENTO DE PROGRAMACION	A	OC	10

Fig. 7. Constancia de grupos atendidos generada en el kiosko

V. CONCLUSIONES Y TRABAJO FUTURO

Con el desarrollo de este proyecto se concluye que la implementación de kioscos electrónicos en procesos administrativos es de gran utilidad ya que mejora la calidad en la generación de documentos pues éstos son oportunos y confiables.

La aplicación del proceso unificado en el desarrollo de este proyecto fue una decisión acertada, ya que hizo que el desarrollo se convirtiera en una actividad más dinámica y

flexible. Al ser un proceso iterativo e incremental simplificó el desarrollo permitiendo ir atendiendo áreas de interés específicas. La construcción de modelos a diferentes niveles de abstracción permitió que el desarrollo del software fuera una actividad menos caótica. La estructuración de componente en capas proporcionó gran flexibilidad y adaptabilidad a los cambios durante la fase de implementación.

Algunas mejoras que se propone añadir al proyecto en el futuro cercano incluyen la incorporación de mecanismos de autenticación, creación de una versión web del kiosko, generación automática y masiva de documentos, añadir mecanismos que permitan identificar posibles nuevos documentos susceptibles de ser incorporados o bien cuáles deberán ser analizados para mejorar el proceso de generación de los mismos e incluir un módulo de configuración de diseño de documentos.

## VI. REFERENCIAS

- [1] I. Sommerville, Ingeniería del software, 7a. ed. Pearson Addison Wesley, 2005.
- [2] I. Jacobson, G. Booch, and J. Rumbaugh, El proceso unificado de desarrollo de software. Addison Wesley, 2000, p. 438.
- [3] G. Booch, J. Rumbaugh, I. Jacobson, El lenguaje unificado de modelado, Addison Wesley, 1999, p. 464.
- [4] Academia de Ingeniería, A.C., “El gobierno digital y la sociedad de la información y el conocimiento”, 2014. [On line]. Available: [http://www.ai.org.mx/ai/images/sitio/2014/05/20140513\\_mgz\\_gobierno\\_digital.pdf](http://www.ai.org.mx/ai/images/sitio/2014/05/20140513_mgz_gobierno_digital.pdf). [Accessed : 18-Feb-2014].
- [5] R. S. Pressman, Ingeniería del Software, Un enfoque práctico, 6ta. ed. Mc Graw Hill, 2005, p. 958.
- [6] M. G. Piattini, J. A. Calvo Manzano, J. Cervera, and L. Fernández, Análisis y diseño de aplicaciones informáticas de gestión, Una perspectiva de ingeniería del software. Alfaomega Ra-Ma, 2004, p. 710.
- [7] S. R. Schach, Ingeniería de software clásica y orientada a objetos, 6ta. ed. Mc Graw Hill, 2006, p. 581.
- [8] C. Fontela, UML Modelado de software para profesionales. Alfaomega, 2011, p. 165.