

DESARROLLANDO UN SISTEMA PARA FLUJOS DE TRABAJO (WORKFLOW)

Joffre Moreira-Pico ¹, Javier López-Zambrano ¹, Karla Cevallos Molina,

Jeniffer Murillo Montesdeoca

¹ Carrera de Computación, Escuela Superior Politécnica Agropecuaria de Manabí
Manuel Félix López, Campus Politécnico El Limón, Km 2.7, Calceta, Ecuador

RESUMEN

Este artículo se realizó con el objetivo de orientar a quienes se inician en el desarrollo de sistemas de flujo de trabajo (Workflow), mediante la explicación de los primeros pasos, técnicas, metodología y demás consideraciones que se deben tener en cuenta para desarrollar un Workflow de calidad. La metodología de generación de interfaces se realizó con base en la experiencia propia de los autores en el desarrollo de éste tipo de sistemas aprovechando las definiciones web stack, en donde se obtuvo un Workflow de alta calidad, mediante una ardua investigación y la aplicación de MEIDAW (Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow); utilizando para la codificación (después de un análisis de lenguajes y herramientas de programación) el patrón de desarrollo MVC (Modelo, Vista, Controlador) en el IDE Visual Studio 2013 bajo el lenguaje C#, además del motor de vistas razor con HTML5 y CSS3, para obtener una buena experiencia de usuario y un sistema eficiente. Esto permite concluir que, mediante el uso de estos métodos y herramientas detallados en la metodología, los desarrolladores sin experiencia en sistemas Workflow, pueden obtener buenos resultados.

Palabras clave: Metodología, Interfaces del Workflow, Flujo de trabajo, BPMN 2.0, WfMC, MEIDAW.

INTRODUCCIÓN

De acuerdo con Lee *et al.* (2011), las Tecnologías de Información y Comunicación (TIC) permiten a las instituciones mejorar su rendimiento a través del uso de herramientas para la automatización de procesos en todas las áreas funcionales: entrada, producción, conversión y salida; las mismas que deben brindar el soporte necesario para alcanzar altos estándares de funcionabilidad, esto se puede lograr a través de Sistemas de Gestión de Flujos de Trabajo (Workflow Management Systems).

La principal característica de los sistemas Workflow según Marante y Santana (2010), es que permiten la realización inteligente de las tareas, utilizando menos recursos por medio de un tratamiento automatizado y eficiente de la información, asegurando la adaptabilidad a los cambios del entorno de un negocio, lo que se contrapone a la falta de libertad y rigidez de las aplicaciones basadas en plataformas. Sin embargo, de acuerdo a lo que indica Alvarado (2011), las características de éstos sistemas varían en función del tipo de Workflow, que como señala el autor citado son:

- **Workflow procedimental (producción o administrativo):** Su orientación es fija porque corresponde a procesos de negocios conocidos de la empresa y sujetos a procedimientos preestablecidos.
- **Workflow ad hoc:** En este caso la orientación del Workflow es dinámica, debido a que se basa en un modelo de trabajo de grupo en el cual los participantes deciden hacia donde se debe dirigir el flujo de trabajo.

Cuando se incursiona en el desarrollo de un Workflow es normal que exista cierta incertidumbre por parte del equipo de trabajo, al no saber qué es lo primero que se debe hacer, qué herramientas utilizar o qué lenguaje de programación es mejor, o si con lo que se va a realizar se cumple con las expectativas de calidad. Hay una gran variedad de análisis sobre los lenguajes de programación existentes y estándares para la documentación de requerimientos, los cuales según Cervantes y Gómez (2012) son primordiales para realizar un sistema, además de propuestas metodológicas para el desarrollo de software; sin embargo, estos lineamientos no están enfocados al desarrollo e implementación de aplicaciones de flujo de trabajo para los procesos empresariales.

Este artículo tiene como finalidad orientar a los desarrolladores que se están iniciando en la realización de sistemas de gestión de flujo de trabajo, a través de una metodología de generación de interfaces para la automatización de procesos de negocio; donde se detallan las herramientas y métodos que pueden ser utilizados para obtener un Workflow de calidad; a partir del modelo genérico para la elaboración de estos sistemas expuesto por la Workflow Management Coalition (WFMC), y la implementación de la metodología MEIDAW.

MATERIALES Y MÉTODOS

Existen muchas herramientas de desarrollo de software, y la elección se basa en las características y ventajas de cada una, enfocadas a los requerimientos del sistema que se va a realizar. Para la elaboración de la plataforma de generación de aplicaciones de flujo de trabajo de procesos administrativos, implementado por los autores de este documento, se utilizaron computadores con especificidades básicas para la etapa de desarrollo. Además, se realizó un análisis donde se definen las herramientas que pueden ser utilizadas para desarrollar un Workflow administrativo como solución web, por lo cual se hace uso de la web stack, también denominado web application stack o conjunto de soluciones (solution stack), como parte de la estrategia previa para el desarrollo de páginas y aplicaciones web.

Esta arquitectura (web stack) hace referencia al método de apilamiento de un conjunto de herramientas para el desarrollo web, se compone de:

- **Sistema operativo:** Es la interfaz central entre el hardware fundamental del servidor (procesador, unidad de memoria, espacio web) y los otros componentes de software. Su función es recurrir a los recursos de hardware para garantizar la funcionalidad de las aplicaciones, en especial del servidor web. Algunos de los sistemas sobre los que se puede levantar un web stack son Linux, Windows, Unix y Mac OS.
- **Servidor web:** Es un software instalado en el sistema operativo que entrega documentos al cliente que los solicita, mediante el protocolo de transmisión HTTP. Las soluciones de servidor web más utilizadas incluyen a Apache, Microsoft IIS y nginx.

- **Base de datos:** Dentro de la base de datos se almacenan, los datos necesarios para que el servidor web, en el momento requerido, pueda recurrir a ellos. Por lo cual, el servidor emite peticiones con ayuda de las extensiones del servidor, que la base de datos procesa y devuelve a las mismas. La elección de un gestor de base de datos depende del software del servidor web y al lenguaje de programación utilizado. Los más usados son, Oracle, MySQL, MongoDB, SQLserver, entre otros.
- **Intérprete de lenguaje de programación:** Los intérpretes de lenguaje actúan de dos maneras:
 - **Cliente:** en el navegador que realiza la petición.
 - **Servidor:** la programación en la parte del servidor hace posible los cálculos necesarios para representar las páginas.

Los interprete más utilizados son: PHP, ASP.NET, Java, Ruby, Perl o Python.

Los componentes de software de un web stack pueden combinarse de forma individual, por lo que existe un gran número de versiones, cada una para diferentes fines.

Los autores utilizaron la web stack WISA, cuyo interprete de lenguaje de programación es C#, por lo cual se empleó el IDE de programación Visual Studio Community con una licencia gratuita para desarrolladores individuales, proyectos de código abierto, investigación académica, educación y pequeños equipos profesionales; y como servidor web se utilizó Internet Information Service (IIS 8.5) del sistema operativo Windows. Ajustándose también a los requerimientos de infraestructura del lugar donde se implementó.

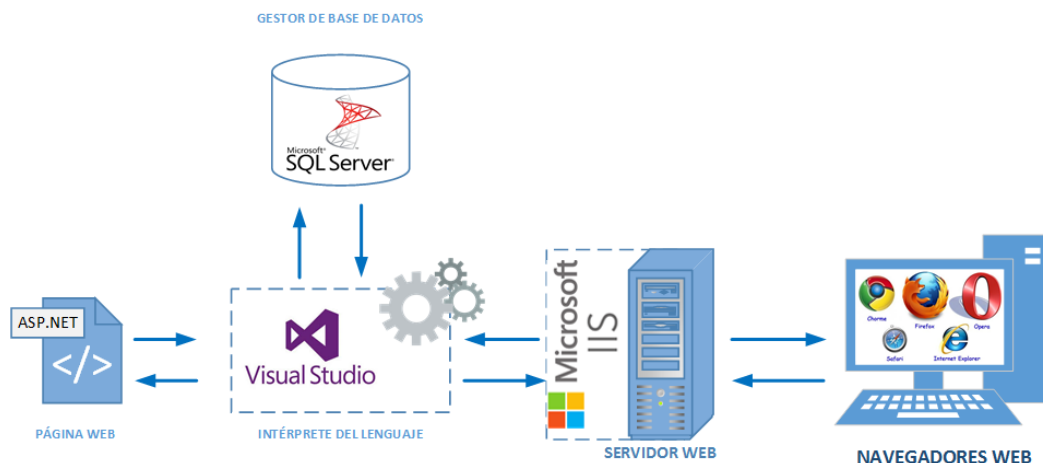


Figura 1. Web Stack WISA **Elaboración:** Autores

Tabla 1. Web Stack

WEB STACK	SISTEMA OPERATIVO	SERVIDOR WEB	GESTOR DE BASE DE DATOS	INTERPRETE DEL LENGUAJE DE PROGRAMACIÓN
STACK-LAMP	LINUX	APACHE	MYSQL	PHP
STACK-WISA	WINDOWS SERVER	INTERNET INFORMATICON SERVICES (IIS)	SQL SERVER	ASP.NET C#
STACK – MEAN	LINUX	NODE.JS	MONGODB	JAVASCRIPT

Elaboración: Autores

Como parte del seguimiento a las actividades realizadas, los autores definieron el tipo de Workflow a desarrollar (procesos administrativos), utilizando la metodología MEIDAW (Metodología Evolutiva Incremental para Desarrollo de Aplicaciones de Workflow), establecida por Mendoza y Wilfredo (2001) citado por Flores y Rufino (2007), como una propuesta para mejorar el proceso de desarrollo de sistemas Workflow, enfocada en los procesos de negocio y la flexibilidad para manejar los cambios de los mismos, la misma que cuenta con dos fases:

- **Fase 1:** denominada “Orientación al proceso”, ésta es una planificación de lo que se va a realizar, y debe estar evidenciada por medio de documentaciones como: documento de incrementos (donde se incluye la planificación, ejecución, control e informe de avances), plan de proyecto (requerimientos generales y estructura de funcionamiento) y documento de alcance de requerimientos.
- **Fase 2:** es la “Orientación al producto”, donde se debe realizar la ejecución de todo lo que se planificó en la fase 1, a través de actividades como: análisis, diseño y desarrollo del sistema, además se realizan pruebas para asegurar el correcto funcionamiento, así como también los documentos formales (manual de usuario y manual del programador), y finalmente la implantación del sistema. Se debe tener en consideración que como se trata de una metodología evolutiva e incremental, cada incremento o avance en el producto, es evidenciado en el documento de incrementos de la fase 1.

RESULTADOS Y DISCUSIÓN

La metodología de generación de interfaces de aplicaciones de flujo de trabajo (Workflow) realizado a partir de la experiencia de los autores en la elaboración de este tipo de sistemas se muestra en la figura 1, donde el primer paso es la identificación del tipo de Workflow a realizar dependiendo de la orientación que este tenga.

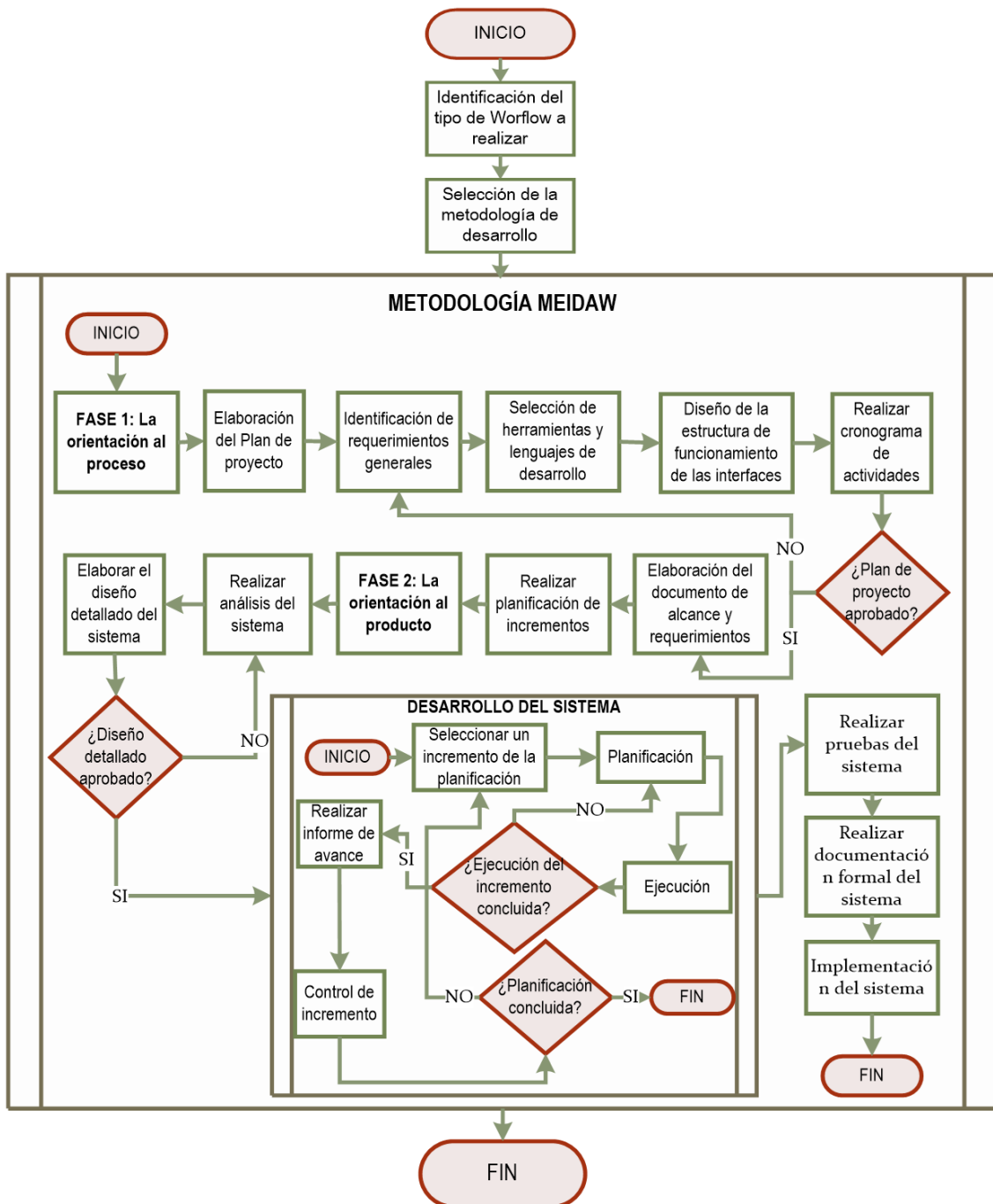


Figura 2. Metodología para la generación de interfaces de Workflow. **Elaboración:** Autores

Para la elección de la metodología de desarrollo se debe tomar en consideración, que cuando se requiere realizar un sistema de gestión de flujo de trabajo, intervienen nuevos requisitos que no se encuentran contemplados en las metodologías tradicionales. Chogollo (2011) hace referencia en que la elección de los métodos, técnicas y procedimientos para el desarrollo de sistemas Workflow se debe realizar con base en las necesidades que tienen las organizaciones de mejorar su estructura de negocio. Por lo cual, los autores implementaron la metodología MEIDAW, como referencia para la elaboración de este tipo de sistema y que se centra en los procesos de negocio y la flexibilidad de los mismos.

La metodología especifica cada una de las actividades que se deben realizar, como por ejemplo la elaboración del plan de proyecto en la fase 1, que incluye la identificación de requerimientos generales a partir del modelo genérico para la elaboración de Workflow expuesto por la WFMC, que es una organización que establece estándares que facilitan la creación, desarrollo y análisis de sistemas Workflow, basado en que todos contienen componentes genéricos que interactúan de manera definida. De acuerdo con Fresno (2010) para que estos componentes puedan interactuar se deben definir un conjunto de interfaces y formatos que permitan el intercambio de datos. Las consideraciones que se llevaron a cabo en esta macro-planificación fueron:

- **Identificación de requerimientos generales del proyecto:** Indica las funciones principales del sistema desde un punto de vista general, definiendo la prioridad de cada requerimiento; esta identificación se realizó con base en el modelo genérico de la WFMC para la elaboración de Workflow (Figura 2), el mismo que define el conjunto de interfaces que se deben realizar.

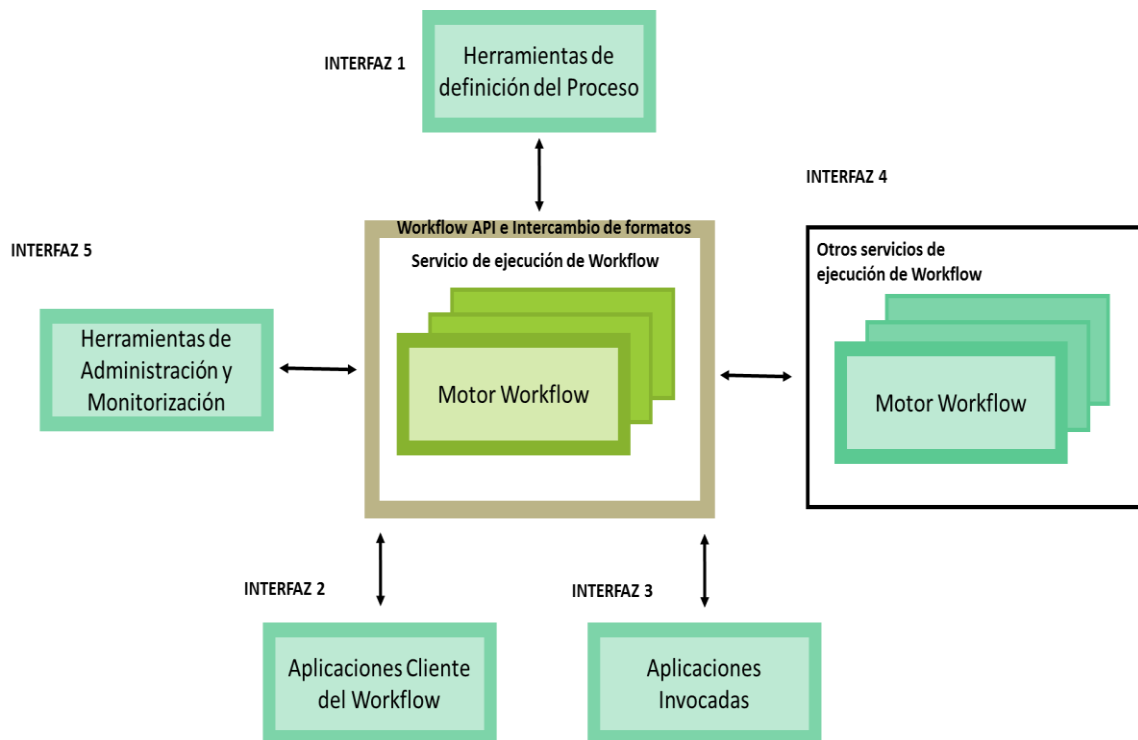


Figura 2. Modelo de referencia de Workflow

Elaboración: Autores

- Selección de los lenguajes y herramientas de desarrollo:** Se debe tomar en consideración que es importante conocer y analizar cuáles son los lenguajes más apropiados para la elaboración de sistemas Workflow, así como también los más utilizados en el área de desarrollo, debido a que entre más se conozca sobre un lenguaje de programación, más información se tiene del mismo y el sistema incrementa su escalabilidad permitiendo a otros desarrolladores trabajar en un lenguaje más común. Además, es fundamental escoger herramientas que soporten la transmisión de datos del flujo de trabajo y que sean de última generación.

Por ejemplo, en el caso del sistema desarrollado por los autores, se realizó un análisis comparativo entre las ventajas y desventajas de los 5 lenguajes más utilizados según el índice TIOBE (2016), en su última actualización de octubre 2016, mostrado en el cuadro 1. El gráfico 1 muestra el porcentaje de utilización de Java, C# y Python (obviando C y C++ debido a que C# es una evolución de estos lenguajes).

Cuadro 1. Índice TIOBE de los 5 lenguajes más utilizados

N.	LENGUAJE DE PROGRAMACIÓN	PUNTUACIONES
1	JAVA	18.75%
2	C	9.20%
3	C++	5.41%
4	C#	3.65%
5	PYTHON	3.56%

Elaboración: Autores

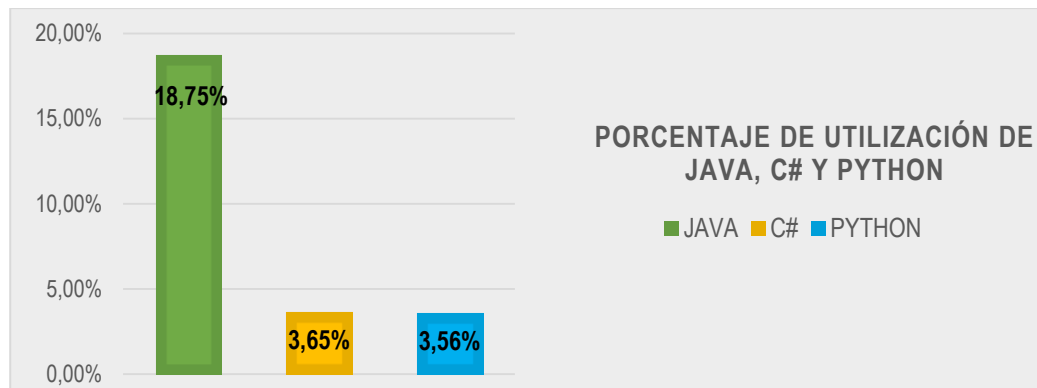


Gráfico 1. Porcentaje de utilización de Java, C# y Python en octubre 2016

Fuente: TIOBE (2016)

Elaboración: Autores

Con base en las características, ventajas y desventajas de los lenguajes mencionados anteriormente, los autores eligieron a C# para la codificación de las interfaces del Workflow, debido a que facilita el mantenimiento de grandes aplicaciones a través de su IDE Visual Studio y aumenta la velocidad de respuesta del servidor.

- **Diseño de la estructura de funcionamiento de las interfaces:** Como parte de la definición de los elementos de las interfaces, fue establecida la estructura del proyecto y para tener un mejor entendimiento de la forma en la que el usuario interactúa con las interfaces del sistema, se realizó la documentación de casos de uso. Los autores detallaron el funcionamiento de las interfaces del Workflow a través del diseño del diagrama de clases, debido a que éste representa la estructura estática del sistema, atributos, operaciones (o métodos); la figura 3 muestra un fragmento del diagrama de clases para el modelado de procesos,

que se encuentra asociado con cada elemento del flujo de trabajo para manejar la interacción con el usuario.

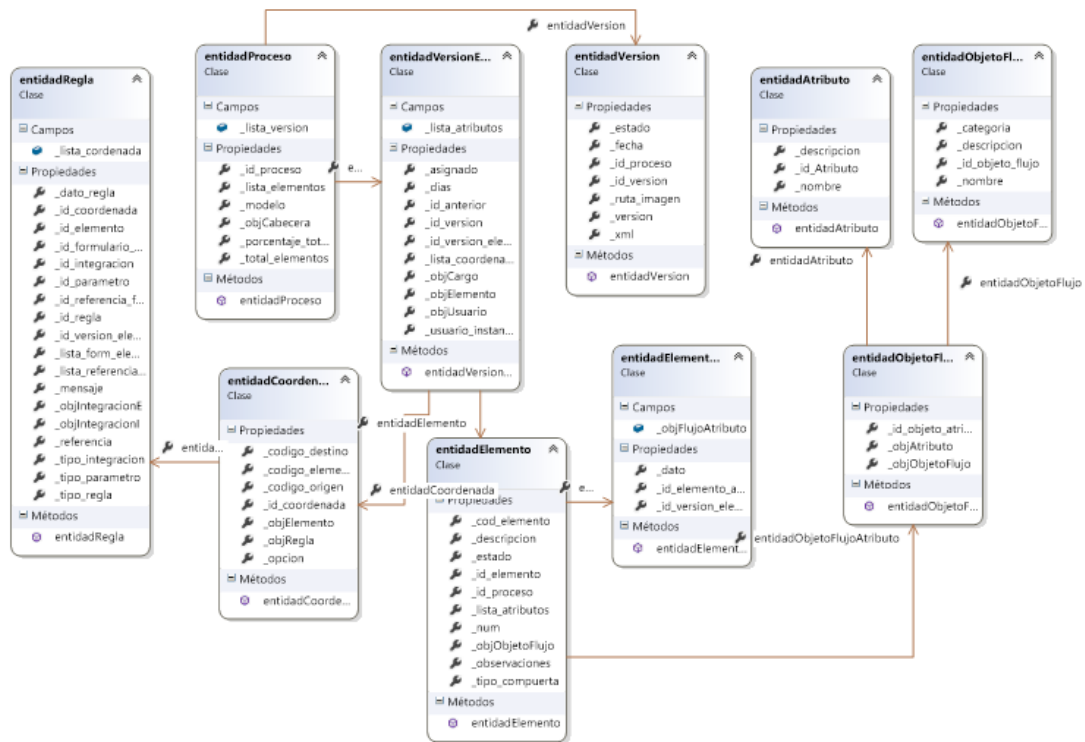


Figura 3. Diagrama de clases (Fragmento para el modelado de procesos)

Elaboración: Autores

Vidal *et al.*, (2014) en su investigación denominada “Propuesta y Aplicación de Diagramas de Clases UML JPI”, manifiesta que los diagramas de clases permiten crear un diseño conceptual de la información que será manejada en el sistema e identificar cada una de las clases que intervendrán, además de que permite definir la estructura funcional de las interfaces desde un punto de vista más lógico, a través de los objetos y atributos incluidos.

- **Cronograma de actividades:** Se realizó un cronograma especificando las actividades, responsables, tiempos, recursos, estimación de esfuerzo en horas y la complejidad.

La elaboración del documento de alcance y requerimientos se realizó de acuerdo al estándar IEEE 830, donde se definieron los requisitos funcionales del sistema; para luego realizar la planificación de los incrementos de los avances que se generaron en el sistema con una duración de 15 días, según lo indica la metodología establecida. Cada incremento del documento señalado involucra los siguientes procesos, que son efectuados en el desarrollo del sistema:

- **Planificación:** Donde se definieron cada una de las actividades de los requerimientos y la administración de los recursos de manera cronológica, teniendo en consideración las asignaciones, notificaciones, permisos, entre otros.
- **Ejecución:** Por cada planificación de un incremento se realizó una ejecución para evaluar el estado del sistema en cada avance.
- **Control de incrementos:** En cada incremento se realizaron Feedbacks (son consideradas retroalimentaciones del sistema) positivos y negativos, para suscitar desvíos favorables o para controlar fallas en el sistema.
- **Informes de avances:** Dentro de la documentación de incrementos se definieron informes de avances donde se indica: estado actual del sistema, actividades completadas y retrasadas.

Siguiendo el orden de la metodología, una vez culminada la fase 1 se debe realizar la fase 2 denominada “Orientación al producto” que incluye el desarrollo, implementación y pruebas, por lo cual se efectuó un análisis del sistema y un diseño detallado del mismo, donde se debe definir las características que deben tener las interfaces.

Entre las principales características de las interfaces para el levantamiento de procesos, está la utilización de una plantilla responsive (que se adapte a los distintos tipos de pantalla donde se visualizaran las interfaces). Para la diagramación de un flujo de trabajo los autores utilizaron elementos del estándar BPMN 2.0, BPMAmericas (2016) menciona que este estándar brinda el conjunto de eventos, tareas, compuertas y artefactos necesarios para el diseño de los procesos de negocio.

Los autores desarrollaron las interfaces del Workflow de procesos administrativos siguiendo el esquema de las metodologías tradicionales, incluyendo la planificación y ejecución de los incrementos de acuerdo a lo especificado en la planificación de los mismos dentro de la fase 1. Para finalizar con las etapas, se debe realizar las pruebas respectivas del sistema y la documentación formal de mismo, que pueden incluir manual de usuario, programador, diagrama de clases, entre otros.

CONCLUSIONES

La utilización de la metodología MEIDAW para la generación de interfaces de aplicaciones Workflow, permite obtener un sistema eficiente, encargado del flujo de trabajo de procesos administrativos; por lo cual los autores consideran que el equipo de trabajo que está iniciando en el desarrollo de éste tipo de sistemas, puede seguir las pautas y procedimientos planteados en éste documento, teniendo la certeza de que obtendrán un producto final de alta calidad, mediante la identificación de requerimientos generales del proyecto y la selección de lenguajes y herramientas de desarrollo que permitan la escalabilidad del sistema y el entendimiento de posibles nuevos desarrolladores.

Es primordial definir la estructura del sistema, para tener un mejor entendimiento del funcionamiento del mismo, todas estas actividades que componen a la metodología de desarrollo deben estar ligadas al cronograma de trabajo, donde cada miembro del equipo de desarrolladores debe ser responsable de las tareas que le sean asignadas.

BIBLIOGRAFÍA

- Alvarado, P. 2011. BONITA SOFT: Gestor de procesos de negocios. Bogotá, COL. Revista Universidad Nacional de Colombia. p 40 -53.
- BPMamericas, 2016. Estándares BPMN 2.0. CU. Revista CENIC. Vol. 41. p 1-9.
- Cervantes, J y Gómez, M. 2012. Taxonomía de los modelos y metodologías de desarrollo de software más utilizados. Distrito Federal, ME. Revista Universidades. Vol. 62. Núm 52. p 37-47.
- Choglo, J. 2011. Estudio de las Metodologías de automatización de procesos de negocio mediante tecnología BPM y su aplicación en el proceso de atención de requerimientos del Dpto. de Servicio Técnico de la Empresa "Almacenes Juan Eljuri". Tesis. Ing. Sistemas. Facultad de Informática. Universidad Politécnica Salesiana. Cuenca, EC.
- Flores, D. y Rufino, J. 2007. Implementación de un Sistema de Gestión del Flujo de Trabajo de los Procesos Administrativos del área de Promoción y Afiliación de la Sociedad Nacional de Industrias – SNI sede Lambayeque. Tesis. Ing. de Sistemas. Universidad Señor de Sipán. Chiclayo, PE. p 60-81

- Fresno, C. 2010. Workflows flexibles para procesos de desarrollo de software. Tesis. Máster en Ingeniería del Software, Métodos Formales y Sistemas de Información. Universidad Politécnica de Valencia. Valencia, ES. p 145
- Lee, Y; Chu, P; Tseng, H. 2011. Corporate performance of ICT-enabled business process re-engineering. Kaohsiung, TPE. Emerald Group. Vol. 111, núm. 5. p. 735-754
- Marante, M y Santana, W. 2010. Gestión de procesos de negocio. Enfoque conceptual orientado a los sistemas de información empresariales Ciencias de la Información. La Habana, CU. Revista Ciencias de la Información. Vol. 41, núm. 2. p. 11-15.
- Vidal, C; Rivero, S; López, L; Pereira, C. 2014. Propuesta y Aplicación de Diagramas de Clases UML JPI. Revista Información tecnológica. Vol. 25. p. 113-120
- TIOBE. 2016. Índice TIOBE de octubre 2016. 20 de oct. 2016. Formato HTML. Disponible en <http://www.tiobe.com/tiobe-index>