

Una metodología híbrida (MSF/PMBOK) en el desarrollo de sistemas de información

Alberto Israel Padilla Zavala

Fundación Arturo Rosenblueth, Ciudad de México. México.

Resumen

Las bases de conocimiento que genera un desarrollador se enriquecen con cada sistema creado. El desarrollo de software deja de ser un proceso artesanal en el momento en que se aplica una metodología de desarrollo. Este documento plasma una metodología híbrida de desarrollo basada en la metodología de Microsoft® Solution Framework (MSF) con apoyo en la metodología de administración de proyectos contenida en el PMBOK, creando una base sólida de conocimientos útil en el desarrollo de próximos proyectos de tecnologías de información. Esta metodología híbrida se divide en cinco etapas: Visión y Alcances, Planeación, Desarrollo, Estabilización e Implantación. En cada una se describen los procesos y entregables generados obteniendo así una base documental que servirá de referencia para los próximos proyectos a desarrollar.

Abstract

Each software created enriches the knowledge bases generated by a developer. Software development ceases to be an artisan process the moment a development methodology is applied. This document represents a hybrid development methodology based on the Microsoft® Solution Framework (MSF) methodology supported by the project management methodology contained in the PMBOK, creating a solid base of knowledge useful in the development of future IT projects. This hybrid methodology contains five stages: Vision and Scope, Planning, Development, Stabilization, and Implementation. Each phase describes the processes and deliverables generated, thus obtaining a documentary base that will serve as a reference for the next projects to be developed.

Palabras clave: Modelo; Híbrido; MSF; PMBOK; Software.

Keywords: Model; Hybrid; MSF; PMBOK; Software.

1. INTRODUCCIÓN

Los modelos de desarrollo de software sirven como base para llevar el control en el proceso que se sigue para crear un sistema de información. Entender el ciclo de vida de un software permite controlar las diferentes etapas de desarrollo hasta su implementación y consiguiente mantenimiento. Dichos modelos proveen una serie de herramientas a fin de llevar a buen término el desarrollo mismo, sin embargo, en la práctica, la aplicación de un solo modelo puede dejar expuesto algún déficit de información o dejar endeble alguna parte del sistema, dejando procesos expuestos a la generación artesanal de los elementos por parte de los desarrolladores.

Al comenzar utilizando el modelo MSF para la generación de software, que es un modelo que se centra en la generación de los procesos y en la fase propia del desarrollo del software y, sin embargo, deja de lado los aspectos de documentación del proyecto; esta debilidad se puede contrarrestar con el modelo para la administración de proyectos desarrollada por el Project Management Institute el cual provee procesos, modelos y plantillas útiles para gestionar cualquier proyecto, incluidos los de desarrollo de software.

Dos modelos que fácilmente se pueden complementar (MSF – PMBOK) y que permiten generar una metodología híbrida de desarrollo aplicable a todos los proyectos de tecnologías de información. La

experiencia en desarrollo de sistemas de información permite este aprendizaje y una consiguiente base de conocimientos aplicables en los desarrollos subsecuentes, que se va fortaleciendo con el tiempo.

2. METODOLOGÍAS

La metodología desarrollada por la compañía de Microsoft® es un modelo aplicable al desarrollo de software que se centra en los procesos. Apoyaremos esta metodología con el modelo de administración de proyectos contenido en el PMBOK, por lo que se utilizarán herramientas de ambos modelos para tener un modelo híbrido que permita tener una base más sólida en cuanto al desarrollo del propio software, pero también en la elaboración de los documentos de soporte y entregables del mismo.

2.1 Microsoft Solution Framework

Como ya se mencionó, la metodología MSF de la compañía de Microsoft® es un modelo que se centra en los procesos y es transparente para cualquier lenguaje de programación. Esta metodología, desarrollada en 1994, ha servido de base para el desarrollo de software, y que se basa en un modelo cíclico, práctico y adaptable a cualquier proyecto de sistemas de información [7].

Todo proyecto de desarrollo de software es separado en cinco principales fases como se muestra en la figura 1:



Figura 1. Fases de la metodología MSF

A continuación, se describen cada una de las etapas:

Visión. En esta etapa se pretende dar una vista general del proyecto. Identifica las tareas y los entregables que permiten al equipo cumplir con los requerimientos y objetivos del proyecto.

Planificación. En esta fase se realiza la preparación de la especificación funcional, diseño de la solución, planes de trabajo, costos y tiempos estimados o calendarización para los entregables. Implica la recolección y el análisis de los requerimientos de negocio, de usuario, operacionales y del sistema.

Desarrollo. La meta de la fase de desarrollo es la construcción de los elementos y entregables de la solución, incluidos la infraestructura y la documentación. Básicamente es el desarrollo del software utilizando uno o varios lenguajes de programación.

Estabilización. La solución se pasa a un entorno real de aplicación. Aquí se determina que la solución cumple los criterios de calidad necesarios para pasar a la fase de implantación. Se reproduce la solución en condiciones reales y el equipo se concentra en detectar y priorizar errores.

Implantación. Se instalan los componentes, se comienza a usar el proyecto y se obtiene la aprobación del cliente. El equipo entonces menciona las lecciones aprendidas.

2.2 Metodología de administración de proyectos PMBOK

Dentro de la administración de proyectos, el PMBOK ofrece una serie de procesos ya aceptados dentro de las mejores prácticas, por lo que se puede mencionar que es un estándar reconocido a nivel internacional por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE Std. 1490-2003). Este estándar es aplicable a un amplio rango de proyectos, por lo que puede ser fácilmente adaptable a cualquier desarrollo de software [9].

Hay cinco procesos y nueve áreas de conocimiento en la mayoría de los proyectos según la metodología del PMBOK. Los cinco grupos de procesos básicos se muestran en la figura 2:



Figura 2. Procesos básicos del PMBOK

1. Inicio: Aquí se define el proyecto o una fase del mismo.
2. Planeación: Define los objetivos, y establece el plan de acción requerido para lograrlos, así como el alcance del proyecto.
3. Ejecución: Integra a los recursos humanos y materiales que se necesitan para llevar a cabo el plan de gestión para el proyecto.
4. Control y Monitoreo: Mide y supervisa regularmente el avance a fin de identificar las variaciones respecto del plan de gestión del proyecto, de tal forma que se tomen medidas correctivas cuando sea necesario para cumplir con los objetivos.
5. Cierre: Formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase del mismo.

Es importante mencionar que para iniciar una fase del proyecto no es necesario terminar la anterior, sino que se pueden superponer entre ellas, e interactuar entre fases.

En el PMBOK los procesos son definidos como entes que tienen Entradas (documentos, información, planes, diseños, etc.), Herramientas y Técnicas (mecanismos aplicados a las entradas) y Salidas (documentos, productos, bases de datos, etc.) [1].

Las nueve áreas del conocimiento mencionadas en el PMBOK se muestran en la figura 3:



Figura 3. Áreas de conocimiento del PMBOK

A continuación, se definen las áreas de conocimiento:

1. Gestión de la Integración de Proyectos: Esta fase incluye las tareas requeridas para asegurar que los diferentes elementos que conforman un proyecto estén adecuadamente coordinados.
2. Gestión del Alcance del Proyecto: Describe los procesos que definirán qué hacer y qué no hacer en el proyecto para tener éxito.
3. Gestión del Tiempo del Proyecto: Describe los procesos que se necesitan para que el proyecto se complete a tiempo.
4. Gestión de la Calidad del Proyecto: Describe los procesos que aseguran que el proyecto va a satisfacer las necesidades para las cuales ha sido creado, además de lo establecido en las políticas de calidad.
5. Gestión de Costos del Proyecto: Se establecen los lineamientos para no sobrepasar el presupuesto asignado al proyecto.
6. Gestión de los Riesgos del Proyecto: Se identifican, analizan y se toma acción sobre los riesgos que se presenten en el desarrollo del proyecto.
7. Gestión de los Recursos Humanos del Proyecto: Incluye los procesos para mejorar la efectividad de los actores involucrados en el proyecto.
8. Gestión de las Comunicaciones del Proyecto: Aquí se administra toda la información del proyecto: se asegura la generación, recolección, distribución, almacenamiento y destino final de todos los datos.
9. Gestión de las Adquisiciones del Proyecto: Se definen los procesos para adquirir bienes y servicios ajenos a la organización que realiza el proyecto.

La importancia del PMBOK radica en los estándares que posee, que, al ser avalados internacionalmente permiten orientar de manera eficaz la gestión de proyectos, logrando mejores resultados, sin importar a qué se dedica la empresa, o a qué va enfocado el proyecto [1].

3. MODELO HÍBRIDO DE DESARROLLO

La estructura de desarrollo cíclica, proporcionada por el modelo MSF, se complementa con la documentación, plantillas, buenas prácticas y herramientas proporcionadas por el PMBOK que aplican para cada etapa, para el seguimiento de los proyectos, enfocándose en los entregables y los resultados. De esta manera, el modelo MSF, enfocado mayoritariamente en el producto de software generado, estará respaldado por una serie de lineamientos y guías de procesos proporcionados por el PMBOK.

Este modelo propio de desarrollo, aplicable en cualquier proyecto de desarrollo de software, quedará como referencia y base de conocimiento para posteriores desarrollos de sistemas de información.

Este modelo híbrido se enfocará en los pasos a seguir para obtener, además de un producto final, la documentación de respaldo que garantice la calidad del proyecto. Se tendrán documentos aprobados por todos los actores donde se establezca el alcance del proyecto, evitando así reprocesos y pérdida de tiempo.

La documentación generada al utilizar este modelo híbrido permitirá tener una estandarización en cuanto a la recolección de la información, al diseño y desarrollo del sistema, la codificación y las pruebas y finalmente a la documentación presentada al cliente o dueño del proceso.

El modelo incluirá las cinco fases presentadas en la figura 4, mostrando además los elementos incluidos en cada una, permitiendo establecer la respuesta a las siguientes preguntas:

1. Visión: ¿Qué se requiere?
2. Planeación: ¿Qué elementos tendrá mi producto? ¿Cómo se va a desarrollar?
3. Desarrollo: ¿Se está desarrollando correctamente?
4. Estabilización: ¿Cumple con los requisitos?
5. Implantación: ¿Qué aprendimos?

Por lo tanto, se pretende convertir el desarrollo de software en un proceso formal que incluya la documentación necesaria para generar un producto de calidad y respaldado por todo un modelo de desarrollo que garantice que los requisitos de usuario sean cubiertos y verificables, a fin de que, en un futuro, el software pueda ser escalado o actualizado, pero que sobre todo se obtenga un producto durable.

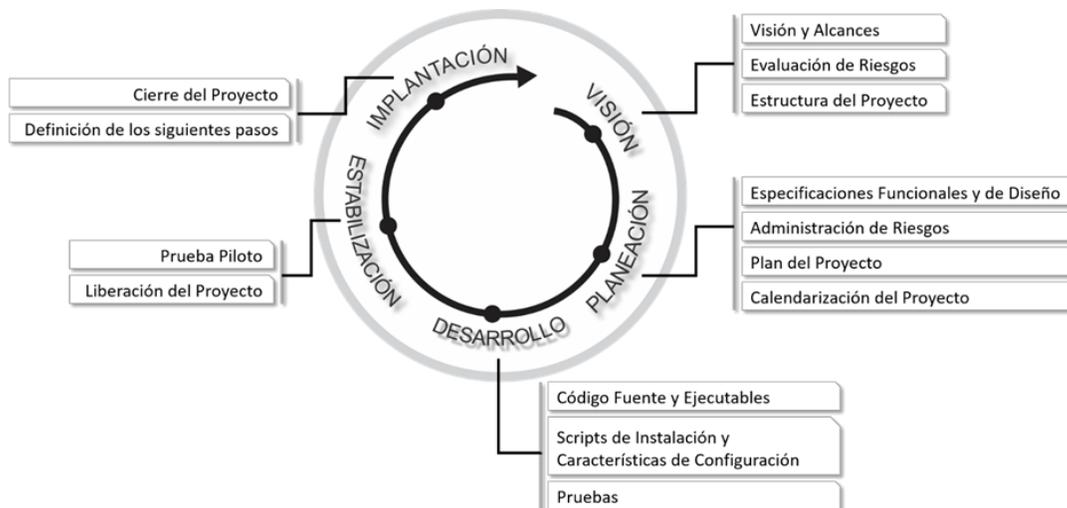


Figura 4. Modelo Híbrido de Desarrollo

A continuación, se mostrarán los procesos generados en cada una de las cinco etapas. El reforzar el modelo MSF con las actividades de gestión de proyectos de PMBOK permitirá tener una base sólida de conocimientos que servirá para posteriores desarrollos. A partir de este punto, se describirá a detalle la metodología empleada en la realización de un proyecto.

3.1 Visión

Esta fase se centra en establecer una visión general del proyecto. La importancia de conocer las necesidades del cliente y las expectativas del negocio nos permite establecer un alcance del proyecto y comenzar a trabajar, estableciendo los roles específicos del equipo de desarrollo [2].

3.1.1 Visión y alcances

Esta sección incluye el análisis del proceso para detectar las oportunidades del negocio. Se define una solución y se hace un análisis de usabilidad. Finalmente se establece el alcance del proyecto con las tareas que realizará el sistema y las que quedan fuera de dicho alcance, y se establece la estrategia de diseño.

Dentro de este punto debemos definir los siguientes elementos:

1. Oportunidades del negocio. Focalizar la problemática que se quiere corregir, enlistando los puntos a favor, los beneficios cuantitativos y la visión a largo plazo del proceso, utilizando el software.
2. Descripción de la solución. Definir los objetivos del sistema a desarrollar, las metas, los supuestos y restricciones que tendrá.
3. Análisis de Usabilidad. Definir los perfiles de los usuarios del sistema y los procesos a los que tienen acceso. En este punto se deberán establecer los escenarios de cada proceso a un nivel de entendimiento del usuario.
4. Alcance del Proyecto. Se definen todos los procesos que incluirá el software a desarrollar, así como aquellos procesos que están fuera del alcance del mismo. Hay que contemplar los escenarios en caso de que haya un cambio considerable en el desarrollo durante alguna de las etapas y establecer los criterios de aceptación una vez que esté terminado el proyecto.
5. Estrategia de Diseño. Definir el ambiente en que funcionará el sistema: sistema operativo, configuración de la red, plataforma de acceso, etc.

Algunos de los entregables, como el mostrado en la tabla 1, muestra los perfiles de usuarios y los procesos a los que tiene acceso.

Tabla 1. Perfiles de usuario

Usuario	Tareas principales
Administrador	Generación de roles de permiso para usuarios Creación de usuarios Administración de lotes Generación de órdenes de producción Cierre de órdenes de producción Cierre contable Impresión de reportes Consulta de seguimiento de lotes
Jefe de producción	Consulta de órdenes de producción Impresión de órdenes de producción
Gerente general	Consulta de órdenes de producción Consulta de seguimiento de lotes Impresión de reportes
Usuario de apoyo	Captura de información de los catálogos de: Proveedores Almacenes Materias primas Recetas

En la figura 5 se muestra el diseño de la aplicación que presenta la estructura de la red local con los equipos que tendrán acceso al sistema.

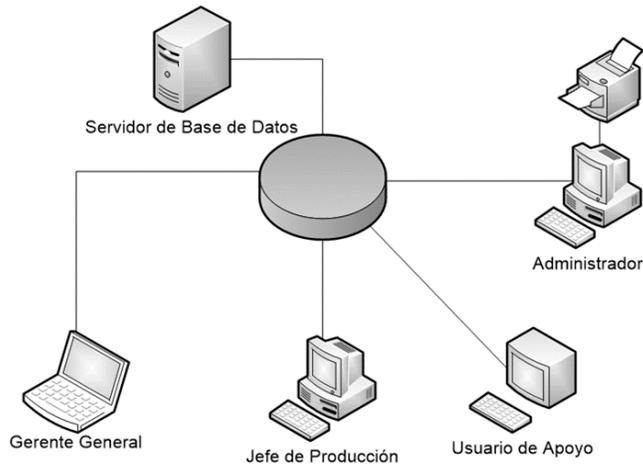


Figura 5. Estructura de organización de equipos.

3.1.2 Evaluación de riesgos

Se deberá definir un listado de los riesgos detectados que pueden comprometer el desarrollo del sistema. Cada riesgo deberá tener los siguientes elementos:

1. Descripción. Un breve texto que explique el riesgo detectado.
2. Escala de probabilidad, impacto y exposición. Valores numéricos que permitirán tomar acciones correctivas.
3. Acción de mitigación. Breve texto que describa qué se debe hacer para que no se presente el riesgo.
4. Acción de contingencia. Breve texto que describa qué se debe hacer en caso de que se presente el riesgo.
5. Responsable. Definir los actores responsables de las acciones de mitigación y de contingencia.

En la figura 6 se muestra cómo quedaría representada la información de los riesgos.

#	Riesgo	Escala de Probabilidad	Escala de Impacto	Exposición	Acción de Mitigación	Acción de Contingencia	Responsable
1	Definición errónea del alcance y límites del proyecto	20	10	200	Participación de los usuarios		Project Manager
2	Mala interpretación de requerimientos	35	10	350	Revisar los requerimientos con el usuario		Project Manager
3	Retraso en la entrega	35	10	350	Poner especial atención en la fase de Planeación	Contratar recursos adicionales	Project Manager
4	Mala abstracción de los componentes	5	10	50	Verificar los componentes y mapearlos con los requerimientos	Corrección de los componentes	Desarrollador
5	Asignación de tareas ajenas al proyecto	70	10	700	Dedicación de tiempo exclusivo a la realización del proyecto	Contratar recursos adicionales	Project Manager
6	Tiempo de disponibilidad limitada por parte del personal del corporativo	20	7	140	Definir un calendario de visitas	Adelantar otros módulos	Project Manager

Figura 6. Tabla de evaluación de riesgos.

3.1.3 Estructura del proyecto

Se definen todas las actividades que se desarrollarán, así como el o los responsables de cada una. En la figura 7 se muestra la asignación de responsabilidades a cada actor.

Actividades del Proyecto	Responsables				
	1	2	3	4	5
1. Visión					
Visión y Alcances	R				I
Evaluación de Riesgos	R				I
Estructura del Proyecto	R				I
2. Planeación					
Especificaciones funcionales y de diseño	R	P			
Administración de Riesgos	R				
Plan del Proyecto	R		P	P	I
Calendarización del Proyecto	R	P	P	P	I
3. Desarrollo					
Código Fuente y Ejecutables	I	R		P	
Scripts de Instalación y características de configuración	I	R	P	P	I
Pruebas	I	P		R	
4. Estabilización					
Elaboración de la Prueba Piloto	P	I	P	R	P
Liberación del Proyecto	R		P		A
5. Implantación					
Cierre del Proyecto	R	I	P	I	A
Definición de los siguientes pasos	R				I
R = Responsable P = Participante I = Se le informa A = Autoriza					

Figura 7. Asignación de responsabilidades.

3.2 Planeación

Durante la fase de planeación se prepara la especificación funcional, se trabaja en el proceso de diseño y se crea el plan de trabajo, además de especificar la calendarización de los entregables [3].

3.2.1 Especificaciones funcionales y de diseño

Esta sección incluye aquellos aspectos de requerimientos, de diseño y los escenarios del sistema, de manera que funcione como:

1. Un contrato entre el cliente y el programador que incluye el alcance y las características que contendrá el sistema.
2. Un documento que describe de una manera técnica las expectativas del cliente.
3. Una base para el programador sobre lo que va a desarrollar.
4. Un documento que servirá para comprobar que el sistema cumple con los requisitos del cliente.

Para completar la especificación funcional se deberán tener en cuenta los siguientes aspectos:

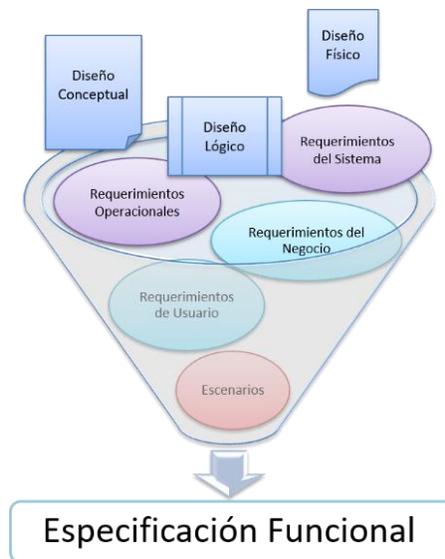


Figura 8. Aspectos de la Especificación Funcional

Como se muestra en la figura 8, son varios los elementos que conforman la especificación funcional:

1. Escenarios. Definición de los casos operacionales del sistema. Incluye precondiciones de funcionamiento, los pasos a seguir para el funcionamiento del sistema y la salida esperada.
2. Requerimientos de usuario. Son entrevistas a los usuarios finales del sistema a fin de conocer el detalle de la operación de los procesos.
3. Requerimientos del negocio. Las entrevistas son ahora al personal de nivel superior para conocer qué información requieren ellos.
4. Requerimientos operacionales. Se hace una investigación sobre el escenario en donde operará el sistema: ambiente de red, tipo de conexiones, sistemas operativos utilizados, hardware y herramientas de software adicionales.
5. Requerimientos del sistema. Se definen los aspectos de hardware y software que necesitará el sistema para funcionar correctamente.
6. Diseño conceptual. Se definen los tipos de usuarios y los módulos que contendrá el software, así como los roles y permisos que tendrán dichos usuarios sobre el sistema.
7. Diseño lógico. Se comienzan a crear las entidades y objetos que formarán el sistema. Esta información será muy útil para los desarrolladores.
8. Diseño físico. Se establecen los componentes que formarán tanto la base de datos como el software, así como las capas de comunicación entre ellos.

En la figura 9 se muestra un caso de uso, un entregable útil para que el desarrollador conozca el flujo de la información.

Proyecto	Sistema MULTICO
Versión	0.1
Caso de Uso	Nuevo Proveedor
Actores	Usuario de Apoyo (UA)
Precondiciones	<ul style="list-style-type: none"> • UA debe tener instalado el sistema MULTICO • UA deberá tener un usuario de acceso al sistema • El usuario debe pertenecer a un Rol de Acceso que tenga permisos sobre el catálogo de Proveedor • UA debe tener la información del proveedor
Flujo Operacional	<ul style="list-style-type: none"> • UA ejecuta el sistema MULTICO • UA selecciona su usuario e ingresa su contraseña • UA accede al catálogo de Proveedores desde el menú del sistema MULTICO • UA presiona el botón "NUEVO" • UA ingresa los datos del proveedor • UA presiona el botón "GUARDAR"
Salida Esperada	<ul style="list-style-type: none"> • Nuevo Proveedor ingresado a la base de datos
Representación	

Figura 9. Representación de un caso de uso.

En la figura 10 se muestra el diseño conceptual con los usuarios del sistema, así como los diferentes módulos que tendrá el mismo.

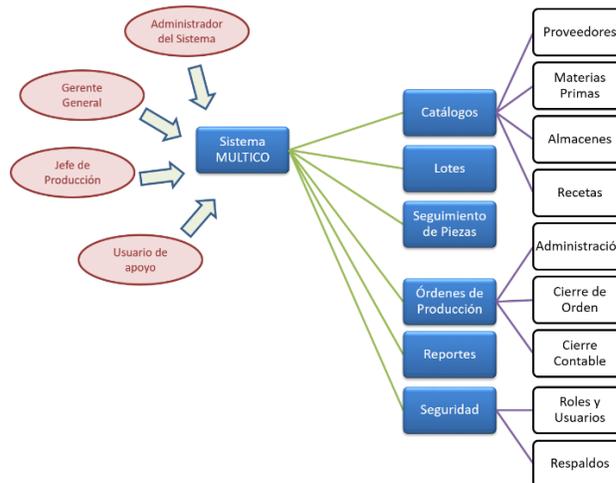


Figura 10. Diseño conceptual.

Mientras que en la figura 11 se muestra la representación del diseño lógico y físico de una parte del sistema.

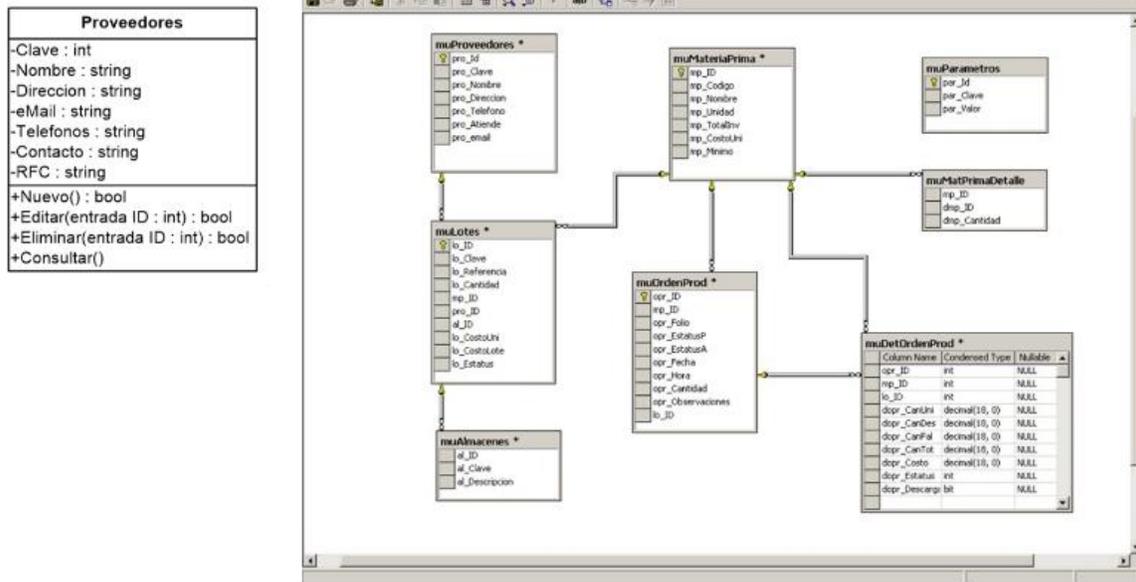


Figura 11. Diseño lógico y físico.

3.2.2 Administración de riesgos

Este apartado incluye maximizar las probabilidades y consecuencias de sucesos positivos y minimizar las probabilidades y consecuencias de sucesos adversos a los objetivos del proyecto.

La administración de riesgos contempla las siguientes actividades:

1. Inspección de seguridad. Se tiene que verificar o inspeccionar condiciones de la infraestructura que da soporte al proyecto (equipos, red, medio ambiente) que pueda afectar la ejecución del proyecto.
2. Observaciones de seguridad. A través de esta actividad será posible verificar que las tareas asociadas al proyecto se están desarrollando según los procedimientos establecidos, lo cual garantiza que no habrá daños a los recursos, menor desarrollo, baja calidad, derroche, retraso o demora.
3. Contactos personales. Son pequeñas reuniones que se llevarán a cabo con el responsable del sistema en la empresa, para tratar temas específicos relacionados con las actividades del proyecto y el desarrollo del mismo.

Si un riesgo es detectado y es considerable, se deberá informar de inmediato al responsable del sistema, el cual convocará a una reunión, ya sea presencial o a distancia, para analizar el impacto, redefinir el alcance si es necesario y para definir el plan de manejo del riesgo. Si el riesgo detectado es de menor consideración, sólo se tomará una decisión en el momento de la detección.

3.2.3 Plan del proyecto

En este punto se definen las políticas sobre las cuales todos los actores involucrados en el proyecto deberán sustentarse. Los planes incluidos son:

1. Plan de costos. Definir detalladamente cada aspecto económico involucrado en el desarrollo del proyecto.
2. Plan de comunicaciones. Se establece la línea de comunicación entre todos los actores, así como los medios a emplear para ello.
3. Plan de desarrollo. Se definen los lenguajes, programas, estándares de codificación, control de versiones, orden de desarrollo de los componentes, los estándares de diseño, colorimetría y todo lo concerniente al proceso de desarrollo del software.
4. Plan de pruebas. Este plan va dirigido a encontrar bugs en la aplicación y para ello hay que usar listas de verificación a fin de que permita a los desarrolladores trabajar en la solución.
5. Plan de operación. Incluye lo concerniente al uso del sistema y su configuración en caso de cambio de equipos.
6. Plan de entrenamiento. Se definen los tipos de capacitación que se le dará a los diferentes tipos de usuarios detectados a fin de que usen la herramienta generada.

3.2.4 Calendarización del Proyecto

Con todos los actores involucrados en el desarrollo del proyecto se define un calendario de actividades, entregables e hitos. Este calendario servirá como pauta a seguir para los desarrolladores. En la figura 12 se muestra el uso de un software como Microsoft© Project para crear el calendario del proyecto.

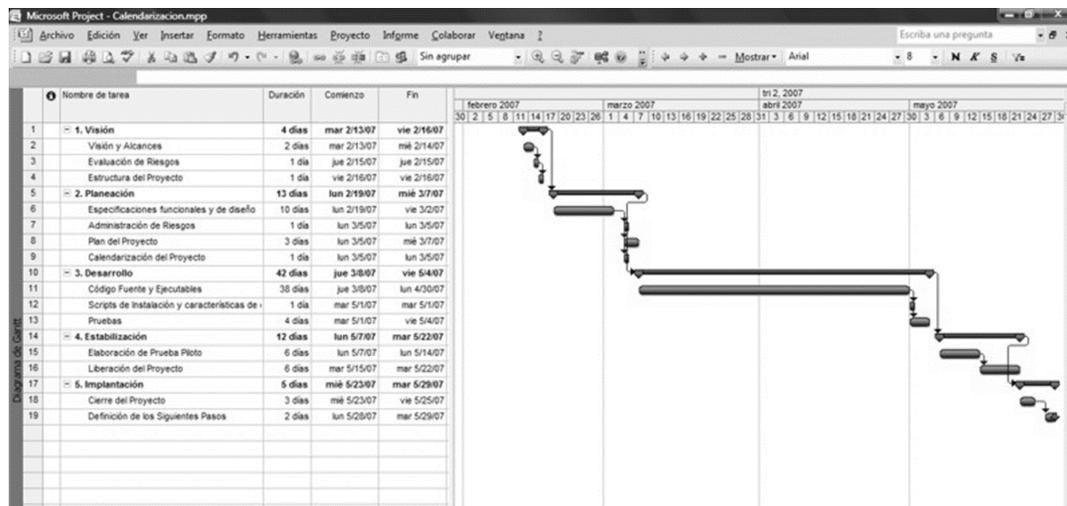


Figura 12. Calendarización del proyecto.

3.3 Desarrollo

Esta etapa involucra la construcción de los componentes del sistema, por lo que incluye la codificación, diseño de interfaces y pruebas unitarias. Aquí se documenta el desarrollo del software y es la base para los encargados del hacer las pruebas para que puedan ir utilizando el sistema y encontrando errores [4].

El software va tomando forma y es momento de mostrar algunos avances a manera de prototipo funcional como se muestra en la figura 13 con la creación de la interfaz de usuario.

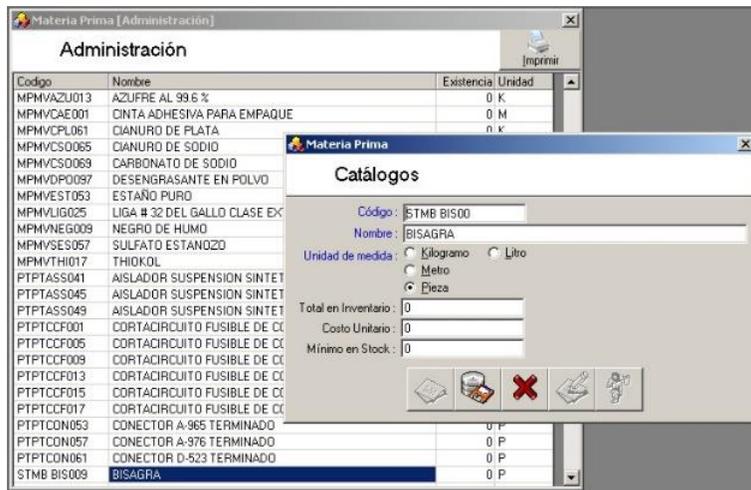


Figura 13. Desarrollo de la aplicación.

Se generan los scripts necesarios para la instalación del software y se hacen las pruebas necesarias en el código a fin de evitar algún error. Se lleva un control de las pruebas unitarias como se muestra en la figura 14.

Proceso	Proveedores	
Módulo	Catálogo de Proveedores	
Fecha	15/Mar/2017	
Verificado por	Tester	
Clasificación	Descripción	Estatus
Lógica	• Cada ciclo tiene su fin de ciclo	OK
	• Las funciones almacenan correctamente los valores de las variables	OK
	• Las sentencias de comparación son correctas y permiten seguir el flujo del programa	OK
	• El uso de banderas lógicas permite seguir el flujo del programa	OK
	• Las banderas están inicializadas	OK
	• Los índices de los ciclos se incrementan (o decrementan) correctamente	NA
Llamadas a Funciones	• El nombre de la función que se está llamando es correcto	OK
	• La llamada de la función contiene los valores de los parámetros requeridos	OK
	• La llamada de la función contiene o no los valores de los parámetros opcionales	OK
	• La llamada de la función contiene los parámetros	OK

Figura 14. Pruebas unitarias.

3.4 Estabilización

El trabajo en esta etapa se centra en probar la aplicación en un ambiente real [5]. Esto incluye los siguientes pasos:

3.4.1 Prueba piloto

Se crea una prueba piloto que incluye:

1. Entorno de prueba. Se definen usuarios que probarán los diferentes módulos del sistema, de preferencia que conozcan el proceso a fin de alimentarlo con información real.
2. Gestión de incidencias. Los detalles encontrados se documentan y se entregan al desarrollador a fin de corregirlos y adaptarlos, según el caso, como se muestra en la figura 15.

Una vez que se termina con las pruebas y ya no hay detalles, se versiona el sistema a fin de crear un entregable.

Sistema	Sistema MULTICO para Control de Órdenes de Producción
Proceso	Lotes
Módulo	Catálogo de Lotes
Sección	Declaración de Variables
Fecha	14/May/2007
Verificado por	Tester
Descripción del error:	No se está cumpliendo el estándar de codificación al nombrar las variables
Observaciones:	El prefijo de las variables tipo Integer es <u>int</u> ; El prefijo de las variables tipo Currency es <u>cur</u> ; El prefijo de las variables tipo String es <u>str</u> .

Sistema	Sistema MULTICO para Control de Órdenes de Producción
Proceso	Proveedores
Módulo	Catálogo de Proveedores
Sección	Campo Teléfono(s)
Fecha	17/May/07
Verificado por	CP José Martínez Felipe
Descripción del error:	No cabe la información completa de los teléfonos de un proveedor. El proveedor tiene 4 diferentes líneas de teléfono.
Observaciones:	Aumentar el tamaño del campo a 50 caracteres.

Figura 15. Plantilla de observaciones.

3.4.2 Liberación del proyecto

Una vez que se tiene el sistema libre de errores considerables, se procede a liberar el proyecto, obteniendo las firmas de las autoridades correspondientes, así como de los involucrados en el proceso de desarrollo como se muestra en la figura 16. Finalmente se crea la documentación de apoyo del sistema.

Si se modificó el Plan del Proyecto, se agregan los cambios solicitados y se procede a firmar de común acuerdo entre las partes. Este documento es definitivo cuando la versión que se encuentra en ejecución se muestra estable.

Se generan los manuales correspondientes de usuario y de instalación, útiles para el encargado del proyecto en la empresa.

Nombre del Sistema	Sistema MULTICO para el Control de Órdenes de Producción	
Versión	1.0	
Fecha	22 de Mayo de 2017	
Firmas	Nombre	Firma
Project Manager	Alberto Israel Padilla Zavala	
Desarrollador	Alberto Israel Padilla Zavala	
Capacitador	Alberto Israel Padilla Zavala	
Tester	Luis Enrique Hernández	
Usuario Final	CP José Martínez Felipe	

Figura 16. Documento de aceptación de versión.

3.5 Implantación

Durante esta etapa del proyecto, se instala la versión final del sistema y el proyecto pasa a manos del personal encargado de operación y soporte. Se le hace una última revisión y se obtienen las impresiones del cliente [6].

3.5.1 Cierre del proyecto

Una vez que se corrigieron todos los detalles encontrados, se procede a la instalación del software en los equipos asignados. Se entregan los ejecutables, scripts, códigos fuente, respaldos, manuales y todos los documentos acordados en el Plan del Proyecto.

3.5.2 Definición de los siguientes pasos

Se documenta la experiencia que tuvieron los actores en la realización de este proyecto. Solo es un documento que servirá de base de conocimiento para proyectos posteriores.

4. CONCLUSIONES

La aplicación de un sistema híbrido para el desarrollo de software es el resultado de una necesidad de aplicar nuevas metodologías, como informático, para mejorar la eficacia con la que se construye el software, pero sobre todo el utilizar nuevas metodologías para el desarrollo de sistemas y salir del paradigma de la construcción artesanal de programas de computadora.

Es sano apoyarse en diferentes modelos y adaptarlos a una metodología de desarrollo a fin de salirse del vicio de tener que terminar los sistemas en un tiempo muy reducido, relegando la documentación del sistema.

Involucrar a los encargados del proyecto en la empresa es vital a fin de que se adentren en la metodología, tiempos y en las pruebas a fin de obtener un producto que cumpla con los requerimientos solicitados.

Al principio se menciona que las bases de conocimiento que genera un desarrollador se enriquecen con cada sistema creado. Ahora se puede decir que estas bases de conocimiento se convierten en la firma indeleble del desarrollador.

REFERENCIAS

- [1] Carlos Adolfo Padilla Domis (2019). El PMBOK como herramienta de dirección estratégica. Universidad Militar Nueva Granada / Facultad de Ciencias Económicas., p14. <https://repository.unimilitar.edu.co/bitstream/handle/10654/35242/CARLOS%20ADOLFO%20PADILLA%20SANTIS.pdf?sequence=1&isAllowed=y>
- [2] Damir Dobric (2006). Microsoft Solution Framework (MSF) Envisioning Phase Overview. Daenet Developers.de. / Daenet's .NET Community. Alemania. http://developers.de/blogs/damir_dobric/archive/2006/05/06/251.aspx Damir Dobric (2006). Microsoft Solution Framework (MSF) Planning Phase Overview. Daenet Developers.de. / Daenet's .NET Community. Alemania. http://developers.de/blogs/damir_dobric/archive/2006/05/06/252.aspx
- [3] Damir Dobric (2006). Microsoft Solution Framework (MSF) Development Phase Overview. Daenet Developers.de. / Daenet's .NET Community. Alemania. http://developers.de/blogs/damir_dobric/archive/2006/05/06/256.aspx
- [4] Damir Dobric (2006). Microsoft Solution Framework (MSF) Stabilizing Phase Overview. Daenet Developers.de. / Daenet's .NET Community. Alemania. http://developers.de/blogs/damir_dobric/archive/2006/05/06/257.aspx
- [5] Damir Dobric (2006). Microsoft Solution Framework (MSF) Deploying Phase Overview. Daenet Developers.de. / Daenet's .NET Community. Alemania. http://developers.de/blogs/damir_dobric/archive/2006/05/06/258.aspx
- [6] Giotis, T. C. (2007). How to deliver successful IT projects using MSF team model and MSF process model. Paper presented at PMI® Global Congress 2007—EMEA, Budapest, Hungary. Newtown Square, PA: Project Management Institute.
- [7] Microsoft (2002). Microsoft Solution Framework White Paper. <https://www.microsoft.com/en-us/download/details.aspx?id=3214>
- [8] Project Management Institute. 2017. Una guía a los Fundamentos de la Dirección de Proyectos, PMBOK Guide. 6a Edición.

Correo autor: israel.padilla@gmail.com