

UNIVERSIDAD DISTRITAL

“FRANCISCO JOSE DE CALDAS”

TRABAJO FINAL

ESPECIALIZACION EN PROYECTOS INFORMATICOS

Modelo de gestión del conocimiento para la dirección de proyectos de desarrollo de software

Autores

Jonathan Daniel Briceño Barrera

Pablo Ernesto Villate León

Director

Víctor Hugo Medina

Bogotá 2018

Nota de aceptación

_____
_____
_____
_____

_____
Firma del Director del Proyecto

_____
Firma del Revisor del Proyecto

Bogotá, 13 de noviembre de 2018

Resumen

La información es uno de los elementos más valorados por las organizaciones, esta se define como la estructuración de datos mediante el conocimiento. En este orden de ideas la información vendría siendo parte fundamental de la gestión del conocimiento.

La gestión del conocimiento ha tomado fuerza en los últimos años, por lo tanto, se tienen como referencia los modelos de transferencia del conocimiento y se rescata lo más importante de cada uno de ellos, con el objetivo de proponer un modelo derivado que unifica esos conceptos clave. Dentro del contenido del trabajo se destaca que el punto central del modelo propuesto será la gestión del conocimiento con el uso de las lecciones aprendidas, por lo que se definen conceptos claves inherentes al tema, también se evalúan documentos de registro de lecciones aprendidas facilitados por dos empresas, con el compromiso de no revelar sus nombres. Esto fue de gran ayuda para definir un modelo que cumpliera con el objetivo principal del trabajo.

Finalmente, se procede a diseñar un modelo entidad relación para construir una base de datos y una arquitectura que plantee los lineamientos para la implementación del prototipo, con esto se da por finalizado el objetivo del trabajo y se plantean trabajos futuros para dar continuidad a la implementación y pruebas de la herramienta final.

PALABRAS CLAVE: *Prototipo*, *Gestión, conocimiento, framework, marco de trabajo, información, gestión del conocimiento, lecciones aprendidas, dirección de proyectos, desarrollo de software, modelo.*

Abstract

Information is one of the most appreciated assets for organizations, it's defined as the data structuring though knowledge. Therefore information would be an important part of knowledge management.

Knowledge management has been gaining force (strenght) in the last years, therefore knowledge transfering models are taken by reference to propose a derivated model which unifies the key concepts of each one. In the document content we highlight that the proposed model's central point will be the knowledge management using Learned Lessons, that's why key concepts inherent with the subject are defined, documents with registered learned lessons issued by 2 companies are evaluated too, with the commitment to not reveal their names. This was a great help in order to define a model that meets this document main objective.

Finally, we proceed to design a entity relationship model in order to build a database and an architecture that presents the criteria for the prototype implementation, with this we finish the document objective and also we propose future works to give continuity to the implementation and test of the final tool.

KEYWORDS: *Prototype, model, knowlegde, management, Project Management, Framework, information, knowledge management, learned lessons, project management, software development, model.*

Agradecimientos

Los autores expresan sus agradecimientos a:

La Universidad Distrital por darnos el espacio para seguir creciendo académicamente.
Los profesores Víctor Medina y Luis Montenegro por sus aportes en el desarrollo y perfeccionamiento del proyecto.
Los profesores que en el transcurso de la especialización brindaron su apoyo y conocimiento incondicionalmente.
Nuestros familiares por el apoyo incondicional en las jornadas de trabajo.

Tabla de contenido

Resumen 4

Abstract 5

Introducción 12

Capítulo 1 Descripción de la investigación 13

1.1 Planteamiento del problema 13

1.2 Pregunta de investigación 13

1.3 Sistematización del problema 13

1.4 Objetivos 14

1.4.1 Objetivo general 14

1.4.2 Objetivos específicos 14

1.5 Justificación 15

1.6 Hipótesis 15

1.7 Alcance y limitaciones 15

1.8 Limitaciones 15

1.8.1 Alcance 16

1.9 Metodología 16

1.9.1 Levantamiento de información 16

1.10 Organización del trabajo 16

1.10.1 Capítulo 1 16

Capítulo 2 Marco teórico 18

2.1 Capital Intelectual 18

2.2 Modelos de gestión del conocimiento 18

2.1.1 Proceso de creación de conocimiento 20

2.1.2 Framework de la organización Inteligente 22

2.1.3 Knowledge Management Asessment Tool KMAT 23

2.1.4 Espiral de TIC 24

2.3 ¿Por qué se debe integrar el ciclo de vida del conocimiento al proceso de desarrollo de software? 25

2.4 Las lecciones aprendidas, ¿por qué es importante tenerlas en cuenta? 28
2.5 Análisis de formatos de captura de lecciones aprendidas 28
2.6 Machine Learning 29
Capítulo 3 Modelo de Gestión de Conocimiento 30
3.1 Situación actual 30
3.2 Definición de aspectos generales para el marco de trabajo 31
3.2.1 Estructura del área 31
3.2.2 Recursos 31
3.2.3 Capital humano 31
3.2.4 Flujo de información 31
3.2.5 Métricas 32
3.2.5.1 Cantidad de defectos reportados por desarrollo 32
3.2.5.2 Criticidad de los defectos 32
3.2.5.3 Tiempo de solución de defectos 32
3.2.5.4 Tiempo de desarrollo vs tiempo estimado 32
3.2.5.5 Índice de rendimiento de costos 32
3.2.5.6 Número de incidentes reportados 32
3.2.5.7 Número de incidentes resueltos 32
3.2.5.8 Número de incidentes reabiertos 32
3.2.5.9 Feedback de los usuarios 33
3.2.5.10 Registro de productos internos 33
3.2.5.11 Entorno 33
3.3 Definición del modelo de gestión de conocimiento 33
3.3.1 Esquema funcional 33
3.3.1.1 Factor procesal 36
3.3.1.2 Factor humano 37
3.3.1.3 Lecciones aprendidas 37
3.3.1.4 Variables de afectación 37
3.3.1.5 Machine Learning 37
3.3.2 Arquitectura 37
3.3.2.1 Descripción de la herramienta 38

3.3.2.2 Modelo Entidad-Relación 39
3.3.2.3 Modelo Relacional 40
3.3.2.4 Infraestructura 45
Capítulo 4 Evaluación y discusión del modelo de Gestión del Conocimiento 48
4.1 Conclusiones 49
4.1.1 Trabajos de investigación futuros 50
Referencias Bibliográficas 51

Tabla de figuras

Fig. 1 Ciclo De Vida Del Conocimiento 20
Fig. 2 Sánchex Díaz, Madery, Modelo de Creación del Conocimiento, Nonaka y Takeuchi 2005 21
Fig. 3 Barragán Ocaña, Alejandro. 2009. Modelo de la organización inteligente de Choo. 22
Fig. 4 Amaya, Karina;et al, 2006,. Modelo KMAT de Andersen 24
Fig. 5 Pérez, Dressler, et al. 2007. Espiral de TIC para los procesos de gestión del conocimiento. 25
Fig. 6 Cadena de alimentación y uso de lecciones aprendidas 34
Fig. 7 Modelo de Gestión de Lecciones Aprendidas propuesto 35
Fig. 8 Modelo de Gestión de Lecciones Aprendidas propuesto con asociación de indicadores 36
Fig. 9 Modelo Entidad-Relación propuesto 40
Fig. 10 Modelo Relacional propuesto 41
Fig. 11 Infraestructura Física 46
Fig. 12 Infraestructura Lógica del Modelo 46

Índice de tablas

Tabla 1 Ejemplos de registro de lecciones aprendidas 43
Tabla 2 Arquitectura conceptual propuesta de la solución 45

Introducción

En el siguiente documento se plantea un modelo de gestión de conocimiento basado en lecciones aprendidas. Primero se abordan conceptos básicos como la transferencia del conocimiento, así como los modelos propuestos por algunos autores.

Se realiza una exploración de algunos conceptos que serán importantes y su relación con el modelo a desarrollar. Estos conceptos son: el desarrollo de software y la gestión del conocimiento, lecciones aprendidas y machine learning. Toda esta base teórica permitió desarrollar un modelo que cumple con el objetivo principal de este trabajo.

La descripción del modelo se hace de manera incremental. Es decir, se parte de un concepto general que permite observar la abstracción más simple de la solución hasta llegar a un nivel de mayor detalle, que facilite el entendimiento de cada elemento allí contenido.

Finalmente se propone una arquitectura base para una posterior implementación que se trata en el apartado de trabajos futuros ya que no forma parte del alcance del proyecto.

Capítulo 1 Descripción de la investigación

1.1 Planteamiento del problema

Las organizaciones buscan la implementación de un modelo para la gestión del conocimiento en cada una de sus áreas, no obstante, cada área de la organización maneja temas que son completamente diferentes pero que convergen para que la organización funcione correctamente, por esta razón es importante definir un marco de trabajo dentro de cada área, por lo tanto, se deben considerar ciertos factores para ver la viabilidad de su implementación, entre los más importantes encontramos:

- Las personas
- El tiempo
- El presupuesto

El área de dirección de proyectos de una empresa cuenta con amplia información de la que podrían beneficiarse los proyectos actuales y futuros, no obstante, en varias organizaciones esta documentación queda almacenada, no se transfiere y tampoco se divulga el conocimiento obtenido, ni las lecciones aprendidas. Debido a esto generalmente se incurre en errores que hubieran podido prevenirse o evitarse de haber tenido un modelo de gestión del conocimiento en el área.

1.2 Pregunta de investigación

¿Cómo fortalecer el área de dirección de proyectos aprovechando las lecciones aprendidas?

1.3 Sistematización del problema

- ¿Es posible implementar un modelo de gestión del conocimiento de forma rápida?
- ¿Es posible que el personal se adapte a una nueva metodología de trabajo?
- ¿Qué beneficios trae gestionar y divulgar el conocimiento?
- ¿Cuáles son los principales parámetros a tener en cuenta para definir un modelo de gestión del conocimiento?

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar un modelo de gestión de conocimiento que fortalezca la dirección de proyectos de software en una organización basado en las lecciones aprendidas.

1.4.2 Objetivos específicos

- Revisión de modelos de gestión de conocimiento existentes.
- Conocer los métodos de captura de las lecciones aprendidas que se utiliza actualmente dentro de la organización.
- Establecer los actores que participan en el levantamiento de lecciones aprendidas.
- Diseñar una base de datos para la gestión del conocimiento para el manejo de lecciones aprendidas.
- Diseñar un mecanismo para la divulgación de las lecciones aprendidas en la organización

1.5 Justificación

El conocimiento es considerado uno de los activos más importantes con los que cuenta una empresa, dicho activo forma parte de cada uno de sus integrantes y permite la realización de sus labores de forma eficiente. En las empresas de desarrollo de software cada proyecto genera diversos conocimientos que podrían ser aplicados a aquellos que se tengan en curso o los que se tendrán en un futuro, pero a pesar de que dicho conocimiento sea documentado esto no es garantía de que será socializado internamente o de fácil acceso para el resto de la organización. Por lo tanto, es importante evaluar diferentes marcos de trabajo Si, para gestión de conocimiento que existen en la actualidad y que permitan solventar esta situación y de esta manera sacar el mayor provecho de los resultados obtenidos por cada proyecto realizado.

La *justificación* es de *tipo práctica*, ya que se espera proponer un marco de trabajo que apoye la gestión de conocimiento en el área de dirección de proyectos.

1.6 Hipótesis

De acuerdo a lo planteado, se presenta la siguiente hipótesis:

“Un modelo de gestión de conocimiento en la dirección de proyectos de software aprovecha las lecciones aprendidas en la dirección de proyectos de software dentro de la organización”

1.7 Alcance y limitaciones

A continuación, se describen las limitaciones que afectan al proyecto y el alcance esperado de su ejecución.

1.8 Limitaciones

Por el tiempo de ejecución del proyecto desde su propuesta hasta su puesta en ejecución, el proyecto se limitará a realizar la investigación y proponer un modelo que se adecúe al área de dirección de proyectos de software.

1.8.1 Alcance

El diseño de un modelo propuesto de gestión de la información para el área de gestión de proyectos de desarrollo de software.

1.9 Metodología

Para la realización del modelo de gestión de conocimiento se identifica un tipo de investigación exploratoria, se espera que estudiando el estado actual de la gestión de conocimiento se obtenga toda la información requerida y con base en esto proponer un modelo que fortalezca el área de dirección de proyectos de software.

1.9.1 Levantamiento de información

Consulta de documentación: Se realiza la consulta de información para la creación del método propuesto. Se logra que dos organizaciones permitan revisar el proceso de gestión de conocimiento que llevan a cabo, ambas llevan a cabo este procedimiento mediante unos formatos en Excel, propuestos por cada una de ellas. Sin embargo, las organizaciones dejan en claro que sus nombres y los documentos no deben ser revelados para dar cumplimiento a cláusulas de privacidad internas.

Búsqueda y análisis de información acorde: Se realiza investigación y análisis de información acorde a la documentación proporcionada por las organizaciones, de esta se complementan los conocimientos acerca de modelos aplicados al área de dirección de proyectos.

1.10 Organización del trabajo

A continuación, se explica el proceso realizado para llevar a cabo el desarrollo de los capítulos que componen el proyecto.

1.10.1 Capítulo 1

Se da inicio a la investigación, se define el problema y en general se dan todos los lineamientos a seguir para desarrollar el proceso investigativo del trabajo. Se definen los objetivos a cumplir y por lo tanto el alcance del proyecto el cual consiste en definir un modelo de gestión de conocimiento basado en lecciones aprendidas. Finalmente se define la metodología que se utilizó para el desarrollo del trabajo.

1.10.2 Capítulo 2

Se realiza revisión bibliográfica de las bases teóricas fundamentales para el desarrollo del proyecto propuesto. Se estudia la teoría de transferencia del conocimiento, rescatando los conceptos más importantes de cada uno, para finalmente generar un diagrama propio. También se estudian los principales modelos de gestión de conocimiento, las lecciones aprendidas al igual que la teoría principal del aprendizaje de máquina.

1.10.3 Capítulo 3

Se define el modelo como tal, tomando como referencias primarias el modelo de gestión de conocimiento “Espiral de TIC” de Pérez y Dressler (2006) en el cual las TIC son el motor de la gestión del conocimiento y el factor humano es incluido en todas las etapas del proceso. Adicionalmente se tuvieron en cuenta formatos usados por dos empresas distintas para realizar el registro de lecciones aprendidas, de allí se sacaron ejemplos para complementar algunas definiciones y conceptos.

De esta forma se pudieron definir todos los componentes del modelo base: los factores humano y procesal, los indicadores o variables de afectación y el entorno dentro del cual se encuentra el modelo. El componente que puede marcar un factor diferencial respecto de otros modelos es la capa de Machine Learning que, mediante el uso de un algoritmo de recomendación y la información del proyecto activo, será capaz de extraer y mostrar las lecciones aprendidas más relevantes al mismo.

Continuando con el desarrollo del modelo se definen indicadores que lo afectan estos podrían verse alterados de acuerdo al entorno y las características de la organización donde se aplique.

Finalmente se propone la estructura de datos y la arquitectura para realizar una implementación del modelo.

1.10.4 Capítulo 4

Se revisan los conceptos aplicados durante el desarrollo del trabajo y se dan algunas recomendaciones, también se justifica la selección de las tecnologías propuestas para la implementación de una herramienta soportada por el modelo propuesto.

Se realizan las conclusiones teniendo en cuenta los objetivos planteados en el capítulo 1 y finalmente se definen los trabajos futuros para continuar con la investigación propuesta en este proyecto.

Capítulo 2 Marco teórico

En este capítulo se profundiza el estudio de las bases teóricas fundamentales permiten establecer el desarrollo del presente proyecto.

2.1 Capital Intelectual

Las organizaciones con aproximaciones a la utilización de modelos de gestión de conocimiento extienden su capital incluyendo un concepto denominado *capital intelectual* el cual se divide en 3 componentes:

- Capital humano, representado por el conocimiento, habilidades y competencias de los integrantes de la organización.
- Capital del cliente, también conocido como capital relacional, es el valor de las relaciones de la organización con sus clientes, la lealtad de estos, los canales de distribución, marcas, licenciamiento y franquicias.
- Capital estructural, son los procesos, sistemas de información, estructuras y propiedades intelectuales independientes de las personas que los crearon.

Desde la generación del concepto se podría decir que se han estructurado algunos mecanismos para facilitar esta actividad, actualmente estos mecanismos son conocidos como modelos y su objetivo principal es dar unos lineamientos básicos para garantizar un flujo de trabajo eficiente y aprovechar al máximo todos los aspectos y ventajas de esta actividad. Desde la década de los 90’s se empezaron a proponer distintos modelos de acuerdo a las necesidades organizacionales, estos a su vez se han venido modificando y/o mejorando, a continuación, se presentarán algunos propuestos en el desarrollo de la gestión del conocimiento.

2.2 Modelos de gestión del conocimiento

Se entiende por gestión del conocimiento a la coordinación sistemática y deliberada de las personas, tecnología, procesos y estructura organizacional con el fin de generar valor a partir de la reutilización e innovación dentro de una organización.

Los principales objetivos de la gestión del conocimiento son:

- Minimizar la pérdida de memoria corporativa.
- Identificar áreas y recursos críticos de conocimiento.
- Facilitar la transición del personal en retiro a sus sucesores.

- La creación de una serie de métodos a ser utilizados dentro de la organización para detener la potencial pérdida de capital intelectual.

Las etapas del proceso o ciclo de vida de la gestión del conocimiento son las siguientes:

- Creación o adquisición del conocimiento, en esta etapa el conocimiento es creado internamente por los trabajadores, adquirido mediante tercerización o comprado de una fuente externa.
- Modificación del conocimiento, en esta etapa la información obtenida es modificada para adaptarse a las necesidades inmediatas y posiblemente futuras de los trabajadores.
- Uso inmediato, la información obtenida es utilizada para cumplir algún propósito.
- Se archiva el conocimiento, consiste en el almacenamiento de la información en un formato que se mantenga en el tiempo y que pueda ser utilizado por los trabajadores en el momento de ser requerido. Para archivar el conocimiento se pueden usar impresiones, distintos formatos digitales o almacenamiento en sitios de internet.
- Transferencia, consiste en el paso de la información almacenada de un lugar a otro, esta maneja variables como costo, seguridad y costo de la transferencia que depende del formato en el que esté almacenada la información.
- Traducción/Reasignación, la información pasa de su formato original a otro que se adecúe más a su nuevo propósito. Por ejemplo, pasar una tabla de datos a un gráfico, un archivo de audio a un sonograma.
- Acceso del usuario, por lo general se puede definir por el rol del trabajador que desee tener acceso a la información, el total de información disponible recolectada sólo puede ser accedida por personal con el rol adecuado.
- Eliminación, del total de información recolectada, aquella a la que no se le va a dar un uso posible es eliminada, el método para seleccionar cuál información mantener y cual eliminar debe estar regido por políticas de la empresa y reglamentación gubernamental.

El proceso anterior puede resumirse en la figura 1

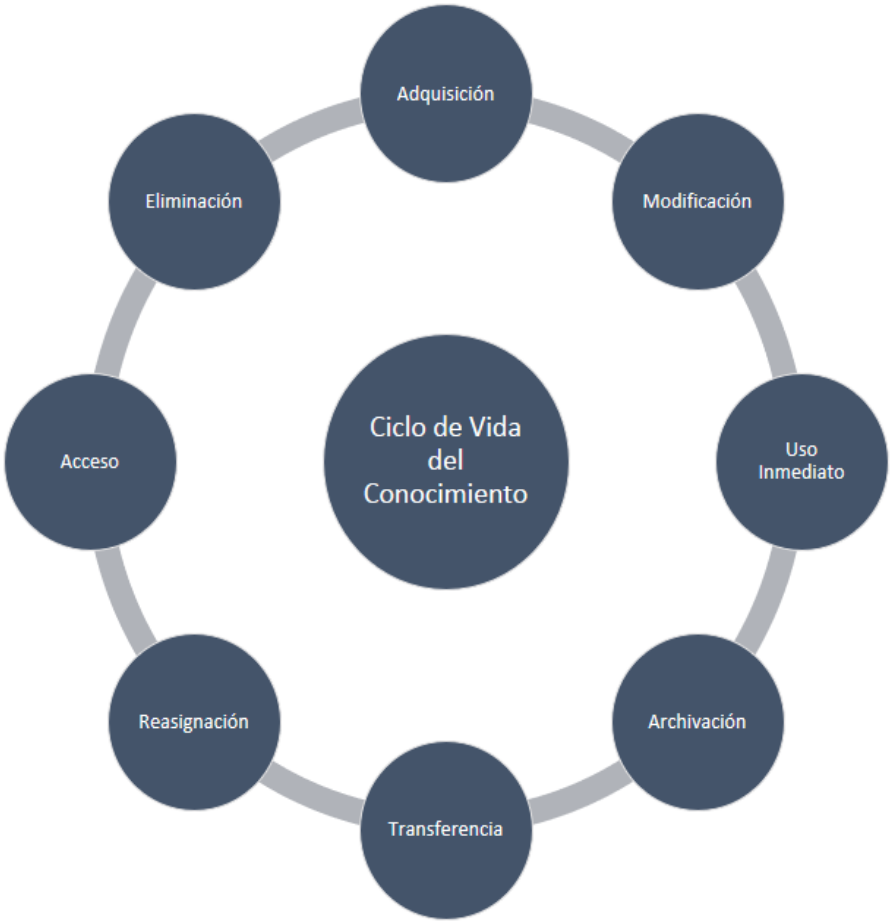


Fig. 1 Ciclo De Vida Del Conocimiento
Fuente: Elaboración propia

2.1.1 Proceso de creación de conocimiento

Nonaka y Takeuchi dividen el conocimiento en tácito, que es el que toda persona posee, no es medible ni tangible ya que es interno; y explícito, que por el contrario es tangible, se caracteriza porque se puede expresar mediante cualquier mecanismo de comunicación (escrito, visual, auditivo, etc.). La base del modelo es la relación entre estos dos "tipos" de conocimiento y sus procesos de conversión, son cuatro:

- Socialización, es la conversión de conocimiento tácito a tácito, consiste en la transferencia de conocimiento de persona a persona, se puede dar directa (por medio de una charla, una capacitación, un taller) o indirectamente (generalmente por observación). Sin embargo, el aspecto más importante de esta actividad es la experiencia colectiva.

- Exteriorización, es la conversión del conocimiento tácito a explícito, consiste esencialmente en compartir el conocimiento personal por medio de cualquier mecanismo entendible para otras personas (escrito, hablado, por señas, braille, etc.), el mayor logro de esta etapa es la transformación de la información en conceptos, a menudo no tan claros pero que promueven la reflexión y la investigación.
- Combinación, es la conversión de conocimiento explícito a explícito, consiste en un ordenamiento, clasificación, combinación y depuración de conceptos obtenidos a través de cualquier medio eficiente de comunicación, con el único objetivo de generar más conocimiento.
- Interiorización, es la conversión de conocimiento explícito a tácito, se da generalmente con la puesta en práctica, se podría decir por ejemplo que se analizan las lecciones aprendidas en experiencias anteriores para evitar cometer los mismos errores y aplicar las prácticas, teorías, métodos o sistemas que tuvieron un efecto positivo. De esta manera se va construyendo un camino (en práctica podría ser un *roadmap*) de cómo ejecutar una tarea específica de forma eficaz y eficiente.

Hay que tener en cuenta que este proceso es cíclico, nunca termina, la figura 2 ilustra mejor el modelo:

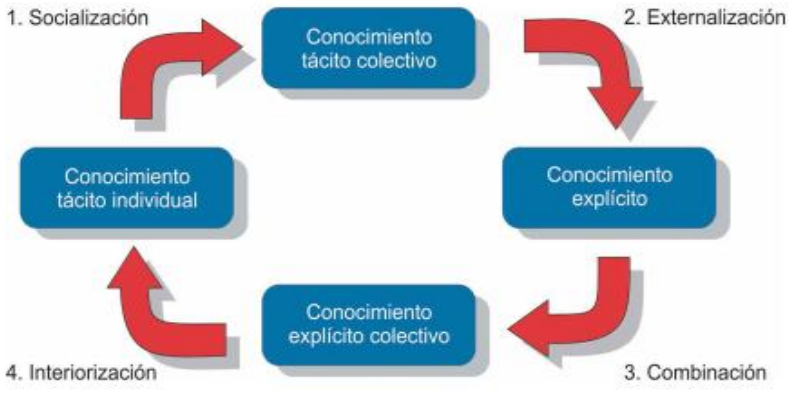


Fig. 2 Modelo de Creación del Conocimiento, Nonaka y Takeuchi
Díaz, Madery 2005

2.1.2 Framework de la organización Inteligente

Choo (2006) propone un modelo que describe la gestión de conocimiento dentro de las características de un modelo cognoscitivo y de capital intelectual. Este modelo tiene como base: la toma de sentido, la creación de conocimiento y la toma de decisiones.

- Toma de Sentido, inicia cuando ocurre un cambio a nivel del ambiente organizacional lo que genera modificaciones y alteraciones en el flujo de trabajo y experiencias para los integrantes de la organización.
- Creación de conocimiento, parte de la relación entre el conocimiento tácito y el conocimiento explícito y del diseño de procesos que permitan la transformación del primer tipo de conocimiento al segundo.
- Toma de decisiones, compromete a una organización a tomar una serie de acciones, es por eso que ésta debe estar regida bajo reglas, premisas y rutinas.

En la figura 3 se representa de forma más clara la propuesta establecida por Choo:

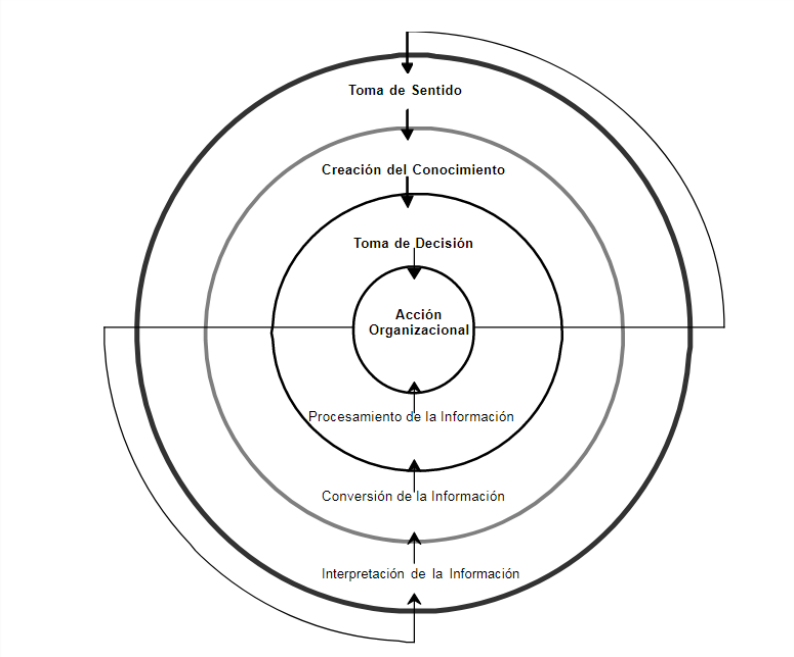


Fig. 3 Modelo de la organización inteligente de Choo
Barragán, 2009.

2.1.3 Knowledge Management Asessment Tool KMAT

Este modelo fue creado por Arthur Andersen Consulting en cooperación con la American Productivity and Quality Center y se basa en el modelo de gestión de conocimiento organizacional en el cual sus principales procesos son: compartir, crear, identificar, recolectar, adaptar, organizar y aplicar. Se apoyan en cuatro habilidades: liderazgo, cultura, tecnología y medición.

- Liderazgo, abarca asuntos relacionados con la estrategia de la organización, cómo ésta define su negocio y utiliza su conocimiento para fortalecer sus competencias.
- Tecnología, establece cómo la organización recolecta, almacena y disemina la información mediante la tecnología. También se refiere a las herramientas tecnológicas usadas para facilitar la comunicación entre sus colaboradores.
- Cultura, refleja cómo la organización incentiva el aprendizaje, la innovación y la forma en que se incentiva el conocimiento dentro de la organización el cual debe estar enfocado en generar valor a los clientes.
- Medición: Abarca la forma en que la organización mide el capital intelectual con el que cuenta y cómo se distribuyen los recursos para motivar su crecimiento.

La interacción de estos procesos se refleja en la figura 4.

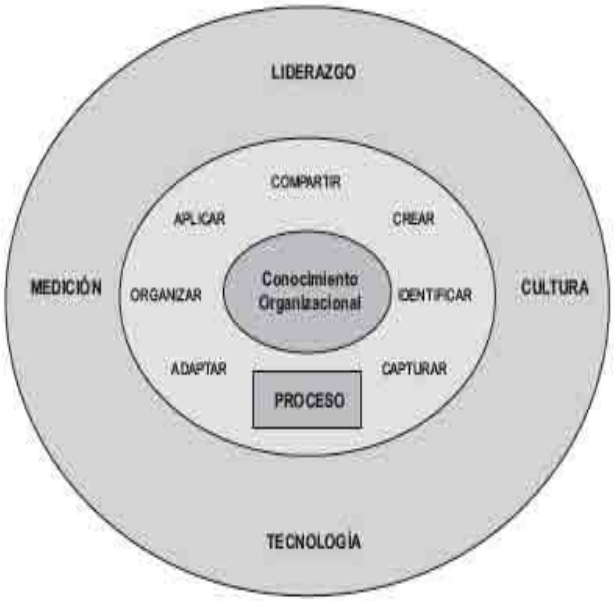


Fig. 4 Modelo KMAT de Andersen
Amaya, Karina;et al, 2006.

2.1.4 Espiral de TIC

De acuerdo con Pérez y Dressler (2006) Las TIC son el motor de la gestión del conocimiento, su correcta utilización permite “hacer reaccionar al resto de factores que intervienen en la gestión del conocimiento” y potencian en gran medida el resto de procesos permitiendo que el conocimiento se transmita y se enriquezca. Básicamente lo que se pretende con este modelo es incluir en cada etapa planteada en el modelo de Nonaka y Takeuchi (1995) los elementos y herramientas tecnológicas que permitan realizar de forma más eficiente cada una de estas etapas planteadas hace más de 20 años.

En conclusión, este modelo plantea una convergencia entre la teoría y la práctica intentando ampliar el impacto en la gestión del conocimiento, se puede tomar como referencia el uso de Internet, lo cual apunta a una gestión del conocimiento más globalizada y aplicable en un entorno ajeno a la organización, por ejemplo cuando se realiza teletrabajo o cuando se realizan viajes de negocios. El modelo se puede explicar de acuerdo a la figura 5.

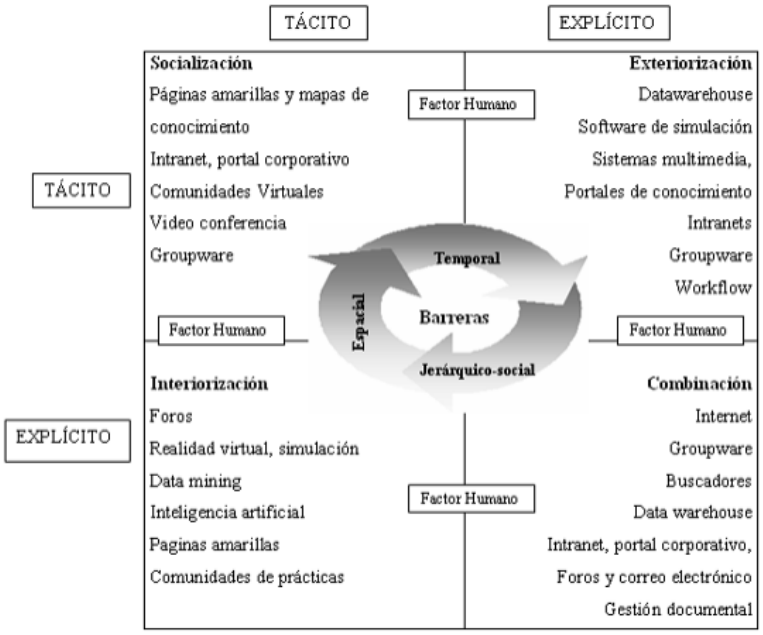


Fig. 5. Espiral de TIC para los procesos de gestión del conocimiento.
Pérez, Dressler, et al. 2007

2.3 ¿Por qué se debe integrar el ciclo de vida del conocimiento al proceso de desarrollo de software?

Shongwe demostró que una compañía de desarrollo de software depende del conocimiento que tiene a su disposición para desarrollar software de alta calidad y de esta forma ser competitivos en la industria del software (Sabri y Alfifi, 2017). Es decir, el conocimiento es uno de los elementos más importantes en el desarrollo de software en general, desde el área técnica hasta el área de dirección. Técnicamente el conocimiento se usa para evitar el reproceso, si una persona implementó una solución eficientemente esta debe quedar documentada para poder ser consultada y utilizada en un proyecto o solución futura de ser necesario, esto evitará todo el proceso de construcción que se hizo en un principio. En el otro extremo la dirección de proyectos usa toda la información del proyecto, la idea es tener este insumo disponible durante la ejecución de otros proyectos buscando cumplir los siguientes objetivos:

1) Evitar el reproceso, todas las tareas, desarrollos, soluciones, etc. hechas pueden ser reutilizadas o modificadas de acuerdo a la necesidad que se tenga.
2) Toma de decisiones, la información obtenida durante la ejecución de un proyecto es el mejor insumo para la toma de decisiones. Con base en experiencias anteriores se puede tomar la decisión de afrontar o de lo contrario evitar

comprometerse en proyectos que puedan generar pérdidas. Estas decisiones son sustentadas con el historial de los proyectos previos.

3) Gestionar el nivel de calidad del producto o servicio, permite evitar errores durante la ejecución del proyecto, ahorrar tiempo y por ende costos si se tiene un buen mecanismo de documentación del proyecto, el resultado será un producto hecho eficientemente.
4) Mejorar el rendimiento del equipo de trabajo.

Básicamente la gestión del conocimiento debería estar involucrada en todos los aspectos de la organización para mejorar la calidad de los procesos. Su propósito es aumentar el conocimiento de la gente, que este sea compartido, usado y reusado, como se mencionó anteriormente en todas las áreas de la organización, varios autores recalcan la importancia de la transformación del conocimiento, cuando pasa de ser tácito a ser explícito.

Lo anterior aplicado al desarrollo de software como tal, el cual pasa por un proceso de construcción, dentro del cual el equipo de trabajo comparte su experiencia y conocimiento para de generar un producto final, este último debe cumplir con los requerimientos establecidos al iniciar el proyecto. De lo contrario, si el conocimiento no fuese compartido es muy probable que el producto resultante no sea de óptima calidad, o en otro caso, el mantenimiento del producto o la búsqueda y solución de errores, serían una tarea tediosa y complicada de ser designada a una persona que no estuvo presente durante la ejecución del proyecto.

La gestión del conocimiento cumple un papel muy importante en el proceso de desarrollo de software, especialmente en el ciclo de vida de desarrollo del sistema, su principal objetivo es facilitar el flujo de información para cada una de sus fases (Sabri y Alfifi, 2017).

Según el autor (Sabri y Alfifi, 2017), hay que considerar y reconocer varios aspectos para tener una mejor perspectiva de la gestión del conocimiento, estos factores son listados a continuación:

- La colaboración de las personas, su actitud positiva frente al flujo de conocimiento.
- La participación de las personas, deben estar en un ambiente donde puedan usar sus conocimientos y compartir su experiencia sin restricciones.
- Apoyo en la construcción de relaciones cercanas entre empleados y empoderarlos para que todos aprendan de todos.
- Una plataforma para gestionar todo el proceso de generación de conocimiento.
- Crear estrategias para la gestión del conocimiento a corto y largo plazo.
- Proteger la información recogida.

- Soporte técnico a la infraestructura técnica.

En el desarrollo de software la gestión del conocimiento es importante, al ser una disciplina guiada por el conocimiento aplicado, está en constante evolución de acuerdo al avance en la tecnología y a los cambios en el mercado.

De acuerdo con Hansen (1999), existen dos estrategias para la gestión del conocimiento:

- Codificación, consiste en sistematizar y almacenar la información que representa conocimiento para la compañía.
- Personalización, se logra al estimular el flujo del conocimiento mediante el intercambio entre las personas, por ejemplo, un listado de personas que pueda ser clasificado por la experticia y nivel de cada uno, de esta forma se les puede solicitar información de ser necesario.

También es importante tener en cuenta ciertos factores (DINGSØYR, 2002) al momento de implementar un modelo de gestión del conocimiento dentro de una organización, en el caso particular, para el área de dirección de proyectos de software:

- Cultura orientada a compartir conocimiento, implica el uso de aplicaciones o herramientas de gestión del conocimiento y depende de la motivación de las personas. Debido a las labores diarias se puede pasar por alto el registro de conocimiento en la aplicación y si esto fuere una tarea obligatoria, puede generar el registro de conocimiento sin valor.
- Enfoque estable, se requiere que la gestión del conocimiento sea un proceso estable en el tiempo, a pesar de cambios organizacionales, para empezar a generar valor en la organización.
- Desarrollo incremental, debe realizarse por iteraciones e ir evolucionando conforme estas avanzan, también se deben tener en cuenta los comentarios de los empleados de la organización, cualquier aporte es válido.
- Acoplarse a objetivos de negocio, para que el área de dirección de proyectos y a futuro la organización inicie el proceso de implementación o adquisición de soluciones para la gestión del conocimiento, con la condición que debe ser originado por una necesidad real, si ésta no existe no se puede justificar que los colaboradores se involucren en su uso.

2.4 Las lecciones aprendidas, ¿por qué es importante tenerlas en cuenta?

Se entiende por lección aprendida toda experiencia, tanto positiva como negativa, que puede ser utilizada para mejorar el desempeño dentro de una organización. Las lecciones aprendidas parten del aprendizaje “bottom-up” o “de abajo hacia arriba”, es decir donde un trabajador aprende algo que podría ser útil y de interés para toda la organización.

Para el entendimiento de las lecciones aprendidas, se deben estudiar los tres tipos de aprendizaje que las engloban:

- Aprendizaje Individual, los trabajadores mediante la realización de sus labores utilizan sus experiencias personales para mejorar los procesos que realizan.
- Aprendizaje mediante comunicación: Este tipo de aprendizaje ocurre a través de la interacción de colegas, donde las experiencias individuales son compartidas, permitiendo un grupo de aprendizaje, esto implica que las lecciones aprendidas por una persona puedan ser aplicadas por el resto de la organización.
- Desarrollo de un repositorio de conocimiento, se encarga del almacenamiento de las lecciones aprendidas de tal manera que puedan ser consultadas cuando sea necesario, en este tipo de aprendizaje se reemplaza la comunicación entre colaboradores por un proceso de recolección, almacenamiento y recuperación de la información.

Cuando exista una nueva lección aprendida se deben resolver ciertos interrogantes, con el fin de validar que sea de utilidad para el área de dirección de proyectos.

- ¿Es la lección aprendida realmente nueva?
- ¿Es consistente con la información previamente almacenada?
- ¿La lección aprendida se puede considerar aislada o está relacionada con otras lecciones previamente almacenadas?
- ¿La lección es lo suficientemente general para ser considerada útil?

2.5 Análisis de formatos de captura de lecciones aprendidas

La captura y almacenamiento de lecciones aprendidas en una de las empresas consultadas se realiza sólo al final de los proyectos, son diligenciadas por cada uno de los involucrados en el proyecto teniendo en cuenta la experiencia de cada quien, finalmente son presentados en la reunión de cierre del proyecto. El archivo donde se consolidan las lecciones se guarda en un repositorio virtual al que sólo tienen acceso los integrantes del proyecto por lo que no trasciende a la organización.

En el otro caso de estudio, se cuenta con un documento transversal a todos los proyectos realizados, las lecciones aprendidas son diligenciadas por el director de proyectos, después se consulta con el equipo de trabajo para generar un documento consolidado. Estas lecciones aprendidas se encuentran en un repositorio que sólo puede ser consultado por otros integrantes del área de dirección de proyectos.

Por lo tanto, en ninguna de las dos organizaciones aplica correctamente el concepto de gestión de conocimiento, este está reservado sólo para un grupo de personas por lo tanto se podría facilitar un entorno donde se reincida en generación de errores comunes y aplicación de malas prácticas.

2.6 Machine Learning

Según Portugal, Alencar, y Cowan (2018) Machine Learning o aprendizaje de máquina consiste en el uso de computadores para simular el aprendizaje humano y así adquirir conocimiento. Los seres humanos poseen la capacidad de aprender de la experiencia debido a su habilidad de raciocinio, los computadores aprenden mediante la aplicación de algoritmos.

El concepto formal de Machine Learning, propuesto por Michalski, Carbonell y Mitchell en 1985 es el siguiente: Se dice que un programa de computador aprende de la experiencia E con respecto a una tarea T con medición de rendimiento P, si el rendimiento en la tarea T, medido por P, mejora con la experiencia E.

Para Machine Learning existen diferentes tipos de algoritmos de acuerdo al tipo de aprendizaje a utilizar:

- **Aprendizaje supervisado:** Ocurre cuando el algoritmo es alimentado con datos de entrenamiento y respuestas correctas, el algoritmo aprende basado en estos datos y aplicar el conocimiento obtenido sobre datos reales.
- **Aprendizaje no supervisado:** El algoritmo de Machine Learning no cuenta con datos de prueba, se alimentan con un set de datos reales, debe identificar patrones ocultos y aprender de dichos datos.
- **Aprendizaje reforzado:** Ocurre cuando el aprendizaje de los algoritmos se basa en un feedback externo ya sea una entidad pensante o el entorno.

Capítulo 3 Modelo de Gestión de Conocimiento

Durante los años 90 se empezó a notar que al reutilizar la información de proyectos previos tenía un impacto muy positivo en el desarrollo de los subsecuentes. Es en este momento que se empieza a hablar de gestión del conocimiento y empiezan a surgir teorías y estándares para comprender que es y cómo se hace. Así como la tecnología va avanzando el concepto va evolucionando a la par, su aplicación empieza a realizarse en cada vez en más áreas y el desarrollo de software no fue la excepción.

Actualmente el desarrollo de software es una de las actividades más ejercidas, en un mundo globalizado surgen constantemente necesidades tecnológicas que son solucionadas mediante esta actividad. Durante varios años se han venido perfeccionando técnicas, herramientas, frameworks, metodologías, lenguajes. En fin, un sin número de elementos que día a día facilitan y optimizan el trabajo de un desarrollador de software, eliminando tareas repetitivas para que éste se concentre en plasmar la lógica del negocio, esta es la tarea más compleja durante la ejecución de un proyecto de desarrollo de software lo que genera defectos, reprocesos, redundancia de código, demoras, aumento de costos y en algunos casos incumplimiento al cliente.

A continuación, se dará un panorama de cómo está acoplada la gestión del conocimiento con el desarrollo de software y se describirán algunas herramientas, su funcionamiento y cómo permiten mejorar los procesos dentro del área de dirección de proyectos.

3.1 Situación actual

Las distintas áreas de una organización se pueden considerar Fábricas de Experiencia, ya que el resultado de las operaciones que realizan diariamente genera conocimiento. Dicho conocimiento debe ser modelado, estructurado, organizado y almacenado para que pueda ser reutilizado por los demás integrantes de una organización, el repositorio de dicha información se conoce como Base de Experiencia.

En el caso particular del desarrollo de software, se puede decir que los proyectos actuales se basan en las Bases de Experiencia generadas por proyectos pasados (ya sea en forma de lecciones aprendidas) y debido a que el desarrollo de software es una disciplina con un flujo constante de nuevas experiencias y conocimientos, se ve beneficiada por los conceptos de Fábricas de Experiencia y Bases de Experiencia.

En el proceso de desarrollo de software se generan gran cantidad de artefactos a lo largo de su ciclo de vida: modelos, diagramas, documentación técnica, documentación funcional, entre otros. Tal variedad de artefactos es beneficiosa a la hora de clasificar

los diferentes tipos de conocimiento generados, lo ideal es que una Base de Experiencia tenga un alto nivel de detalle y que abarque la experiencia de diferentes colaboradores en el desarrollo de actividades técnicas, administrativas y procedimentales.

3.2 Definición de aspectos generales para el marco de trabajo

A continuación, se definirán las variables, el entorno, los datos y otros aspectos relevantes que serán de vital importancia para la definición y posterior propuesta de un modelo genérico, que se ajuste a cualquier área de desarrollo de software.

3.2.1 Estructura del área

Si bien cualquier área de desarrollo es similar en varios aspectos, por experiencia propia se puede afirmar que también tienen características diferentes. Los procesos son llevados de manera distinta, los mecanismos de gestión de la información, las comunicaciones incluso las personas son muy diferentes, esto se debe propiamente al entorno, pues es la empresa quien define y regula todos los aspectos de las áreas que la componen.

3.2.2 Recursos

Por recursos se hace referencia a todo aquel artefacto, persona o elemento inherente al departamento, que genere directa o indirectamente conocimiento. Estos recursos deben ser identificados y validados, pues serán el inicio de la cadena dentro del marco de trabajo.

3.2.3 Capital humano

Es el conjunto de personas que conforman el área, encargados de realizar las tareas y los que mediante diferentes acciones generan, consumen y transmiten información a sus pares.

3.2.4 Flujo de información

Se debe determinar el ciclo de vida del conocimiento (ver Fig. 1), y cómo se va a compartir dentro del área. La gestión de la comunicación debe ser eficiente y la información entregada debe ser confiable. Esto implica que todo lo que se encuentre en el repositorio debe ser validado y depurado exhaustivamente para garantizar que el contenido entregado sea realmente útil y cumpla con el propósito general del modelo, facilitar la toma de decisiones.

3.2.5 Métricas

Las métricas facilitarán varios aspectos dentro del área de desarrollo de software, se podrá medir la efectividad del equipo, el nivel de satisfacción del usuario y la calidad de los productos entregados. Las métricas propuestas son:

3.2.5.1 Cantidad de defectos reportados por desarrollo

Se debe contar con una herramienta de captura y procesamiento de defectos, en esta el cliente registra y categoriza los errores por herramienta, por ejemplo, GLPI.

3.2.5.2 Criticidad de los defectos

Los defectos reportados deben poder ser discriminados por su criticidad (bloqueante, crítico, mayor, menor, estético), para establecer el nivel de calidad del producto desarrollado.

3.2.5.3 Tiempo de solución de defectos

De acuerdo a la criticidad del defecto se debe tener el registro del tiempo de respuesta y el tiempo invertido en solucionarlo.

3.2.5.4 Tiempo de desarrollo vs tiempo estimado

Se debe llevar el control de desfases del tiempo pactado para entrega de un proyecto contra el tiempo real de entrega, para evaluar las razones de la demora en caso de superar el tiempo y las razones de una estimación inexacta en caso de cumplir la meta en un tiempo mucho menor al estimado.

3.2.5.5 Índice de rendimiento de costos

Evaluar la relación entre el presupuesto inicial del trabajo contra el costo real en el transcurso y al finalizar el proyecto.

3.2.5.6 Número de incidentes reportados

Medir la cantidad de defectos o incidentes reportados y analizar si se han reducido conforme se ejecutan otros proyectos.

3.2.5.7 Número de incidentes resueltos

Medir la eficiencia en la resolución de incidentes, esto se logra verificando el estado y la aceptación de la solución por parte del cliente.

3.2.5.8 Número de incidentes reabiertos

El número de veces que el usuario reabre o regenera un bug.

3.2.5.9 Feedback de los usuarios

Mediante encuestas de satisfacción con preguntas tanto cerradas como abiertas, se debe poder establecer el nivel de satisfacción y la percepción del cliente con respecto al servicio de desarrollo prestado y el producto de software desarrollado.

3.2.5.10 Registro de productos internos

Varias compañías por iniciativa propia deciden realizar proyectos internos, es decir invierten de su propio capital para investigar y desarrollar productos o servicios que generan valor para la organización sin importar si fracasan o son exitosos.

3.2.5.11 Entorno

El área en cuestión no es un nodo aislado de la organización, por esta razón es importante identificar con qué áreas interactúa. Sin embargo, es importante identificar los entes externos a la organización con las que el área interactuará ya que serán generadores directos de información útil para el área y por lo tanto para la generación de conocimiento.

3.3 Definición del modelo de gestión de conocimiento

Con las variables definidas se puede proponer el modelo objeto de este trabajo, esto se realizará en 2 etapas, primero la definición de un esquema funcional, lo cual dará una visión global de toda la estructura del modelo y como segunda instancia se definirá una arquitectura tecnológica, la cual será el primer paso para iniciar la construcción de un prototipo y posterior implementación del modelo.

3.3.1 Esquema funcional

Consiste en identificar los factores que generan conocimiento, para este caso se encontrarán en dos categorías: procesal y humano. Dentro del factor procesal se generan los elementos para construir una lección aprendida, esta será el insumo que el factor humano usará al generar conocimiento. En la figura 1 se ilustra lo mencionado anteriormente.

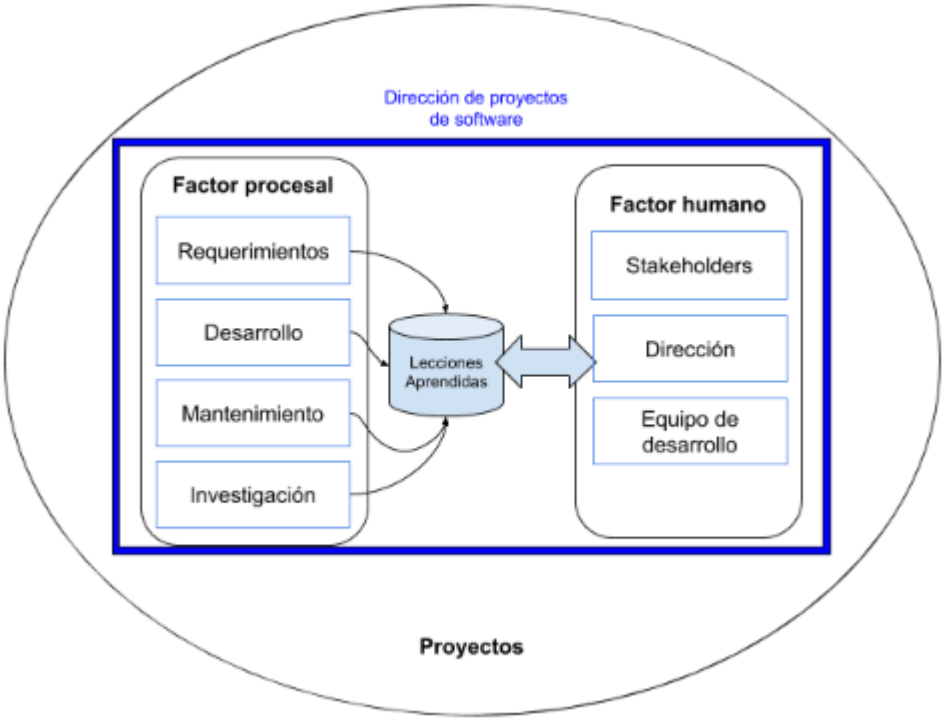


Fig. 6 Cadena de alimentación y uso de lecciones aprendidas
Fuente: Elaboración propia

Una vez se identifican los factores que afectan el modelo, y tomando como referencia la importancia del uso de tecnologías en el modelo de espiral de TIC de Pérez y Dressler en el apoyo a la gestión del conocimiento, se agrega una capa de machine learning que implementará un sistema de recomendaciones de acuerdo a la información ingresada al iniciar un proyecto. También se debe tener en cuenta que los factores definidos serán afectados por los indicadores. En la figura 7 se complementa el modelo con lo mencionado previamente.

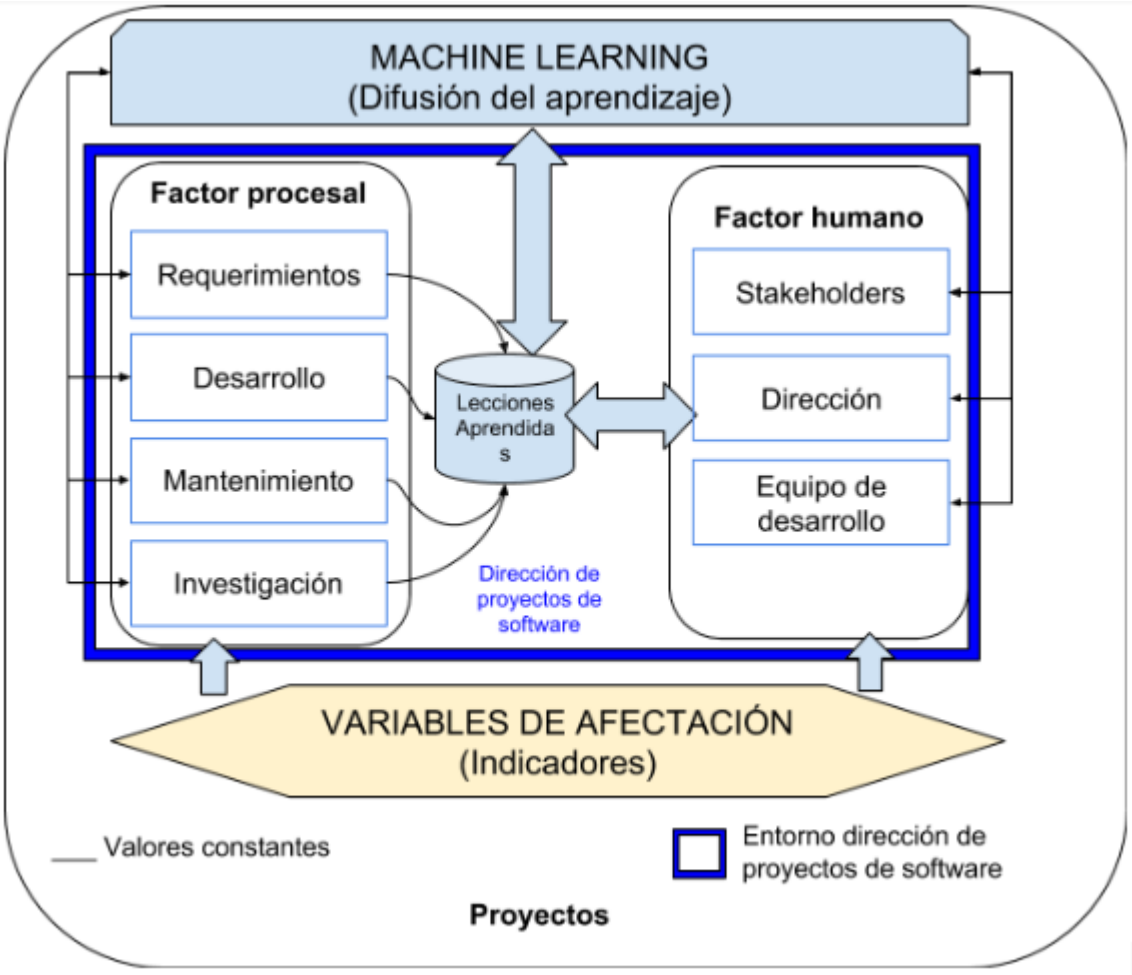


Fig. 7 Modelo de Gestión de Lecciones Aprendidas propuesto
Fuente: Elaboración propia

Con el esquema completo es importante definir qué indicadores afectan a los factores, estos son ilustrados en la figura 8.

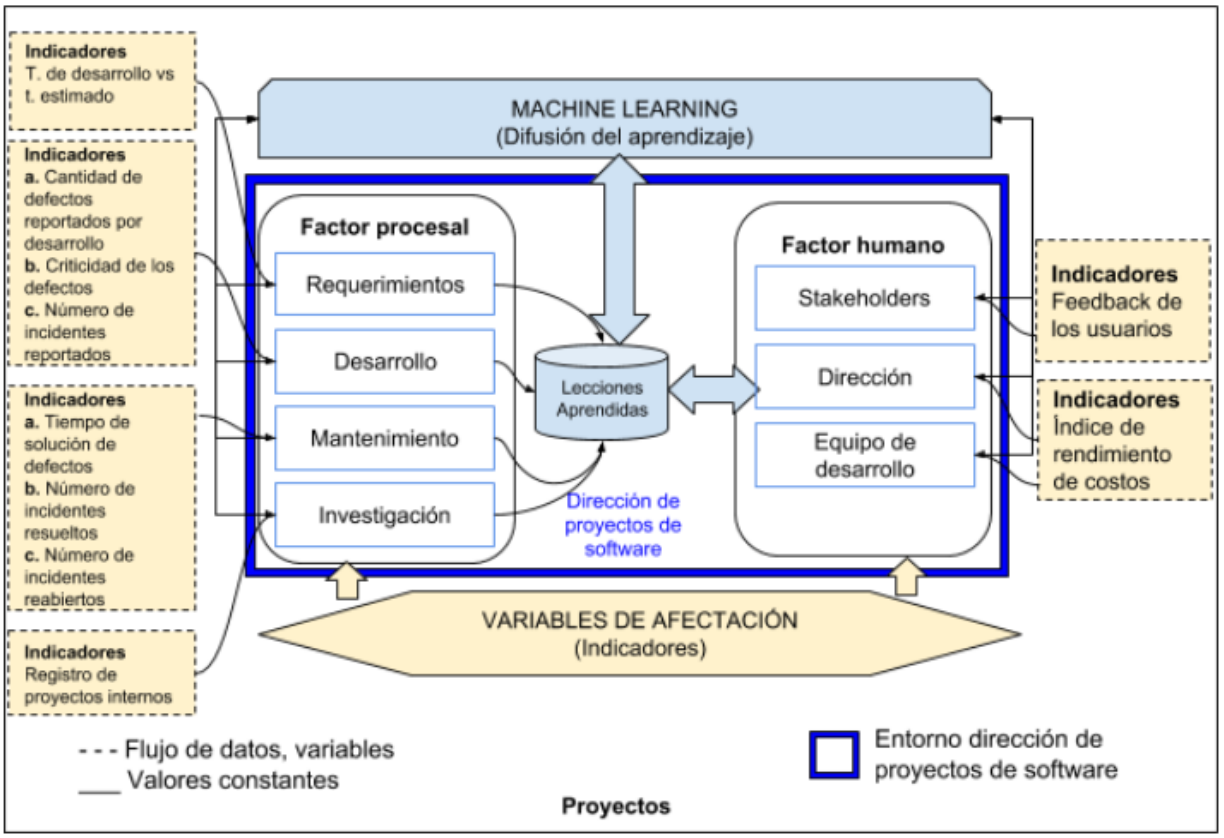


Fig. 8 Modelo de Gestión de Lecciones Aprendidas propuesto con asociación de indicadores
Fuente: Elaboración propia

3.3.1.1 Factor procesal

Son los diferentes procesos que al desarrollar sus actividades pueden generar lecciones aprendidas.

- Requerimientos. Son el centro de todo proyecto, los requerimientos indican las metas a cumplir, una vez que se desarrollen completamente se puede decir que el proyecto ha finalizado.
- Desarrollo. Es el procedimiento mediante el cual se aplican los conocimientos y el talento del equipo de desarrollo para analizar, calcular y crear elementos funcionales que cumplen con los requerimientos planteados.
- Mantenimiento. Son las tareas que se pueden presentar durante o al finalizar el desarrollo, consisten en la corrección de errores, limpieza o documentación de código.
- Investigación, se da cuando existen proyectos internos que generalmente se clasifican como “investigación y desarrollo”, están presentes en fábricas de software y su objetivo es crear productos que generen valor a la compañía. En ocasiones se da cuando en un proceso de desarrollo de software es necesario recurrir a fuentes

de información externas por falta de experiencia del equipo de trabajo o por simple desconocimiento de la temática.

3.3.1.2 Factor humano

Son los actores que realizan las actividades que componen los procesos y pueden utilizar o registrar lecciones aprendidas.

- Stakeholders. Son de suma importancia en el proceso ya que en muchos casos generan los proyectos. Por lo tanto, cualquier proyecto de desarrollo de software está destinado al fracaso si los usuarios y stakeholders no juegan un papel activo en todo su desarrollo.
- Dirección. Es el área encargada de validar y coordinar la ejecución de los proyectos, entre sus responsabilidades está también la toma de decisiones.
- Equipo de desarrollo. Es el talento humano encargado del desarrollo de los requerimientos funcionales de los distintos proyectos, son los que aportan el conocimiento.

3.3.1.3 Lecciones aprendidas

Entendiendo que una lección aprendida es la asimilación de una experiencia previa como conocimiento que puede ser compartido para mejorar el cumplimiento de tareas en la organización, dentro del modelo representan un insumo para el algoritmo de recomendación.

3.3.1.4 Variables de afectación

Son los valores encargados de definir el comportamiento del modelo. Las variables de afectación o indicadores son todas aquellas que como su nombre lo indica pueden llegar a afectar el modelo

3.3.1.5 Machine Learning

La capa de Machine Learning se encargará de analizar, filtrar y presentar la información, esto mediante un algoritmo de recomendación.

3.3.2 Arquitectura

La arquitectura consta de una estructura básica compuesta por una capa de datos y unos servicios soportados en una infraestructura tecnológica.

3.3.2.1 Descripción de la herramienta

La herramienta debe permitir almacenar toda la información que maneja la dirección de proyectos de software, a continuación, se listan los aspectos a tener en cuenta para la implementación de la herramienta:

- Se deben poder registrar las lecciones aprendidas de diferentes proyectos de una organización, así como el tipo de proyecto que se realiza.
- Las lecciones aprendidas pueden tener uno o más autores.
- La lección aprendida puede afectar a una o más etapas del proceso de desarrollo.
- La lección aprendida debe tener un identificador único, un usuario asociado, la fecha de creación debe ser definida de acuerdo al sistema, una descripción de porqué se genera la lección (sus causas), cómo impactó a los objetivos del proyecto, que acciones correctivas y/o preventivas se tomaron y las recomendaciones asociadas a la lección.
- Las causas de la lección se deben tipificar de la siguiente forma:
    - Defectos
    - Retrasos
    - Sobrecostos
    - Falta de personal calificado
    - Estimación deficiente
    - Levantamiento de requerimientos deficiente
    - Falta de liderazgo
    - Otra, ¿Cuál?
- Las acciones realizadas están categorizadas en: preventiva, correctiva o recomendación. Cada acción debe tener un único tipo de acción.
- Las lecciones aprendidas deben estar tipificadas para facilitar su búsqueda, también deben estar asociadas a una etapa específica del proyecto.
- Los tipos de lecciones aprendidas según Rowe, S. F. & Sikes, S. (2006) pueden ser:
    - Causa raíz recurrente
    - Mejores prácticas
    - Tamaño del proyecto
    - Recursos
    - Costos
    - Tipo del proyecto
    - Disponibilidad de recursos
    - Duración del proyecto
    - Repetición de lecciones

- Personas de diferentes proyectos pueden consultar las lecciones aprendidas de otros proyectos.
- El sistema debe solicitar una configuración inicial cuando se ejecute un nuevo proyecto. La información requerida para este incluye el tipo, nombre, fecha de inicio, fecha de finalización y responsable del proyecto.
- Cuando finalice la configuración inicial el sistema deberá mostrar las lecciones aprendidas asociadas a los parámetros ingresados.
- Las etapas del proyecto son:
    - Planeación
    - Desarrollo
    - Pruebas
    - Salida a producción
    - Soporte
- El sistema deberá solicitar un registro completo de cada proyecto nuevo. La información que se quiere registrar incluye:
    - Un identificador único del proyecto
    - Un tipo de proyecto
    - El nombre del proyecto
    - Las fechas de inicio y finalización
- El tipo de proyecto puede ser: investigación, implementación, soporte técnico y mantenimiento, desarrollo a la medida y consultoría.
- Al finalizar el registro del proyecto el algoritmo de recomendación deberá seleccionar y mostrar las lecciones aprendidas de acuerdo a los datos ingresados.

3.3.2.2 Modelo Entidad-Relación

Con base en los requerimientos y funcionalidades identificados se presenta el modelo Entidad Relación, presentado en la figura 9.

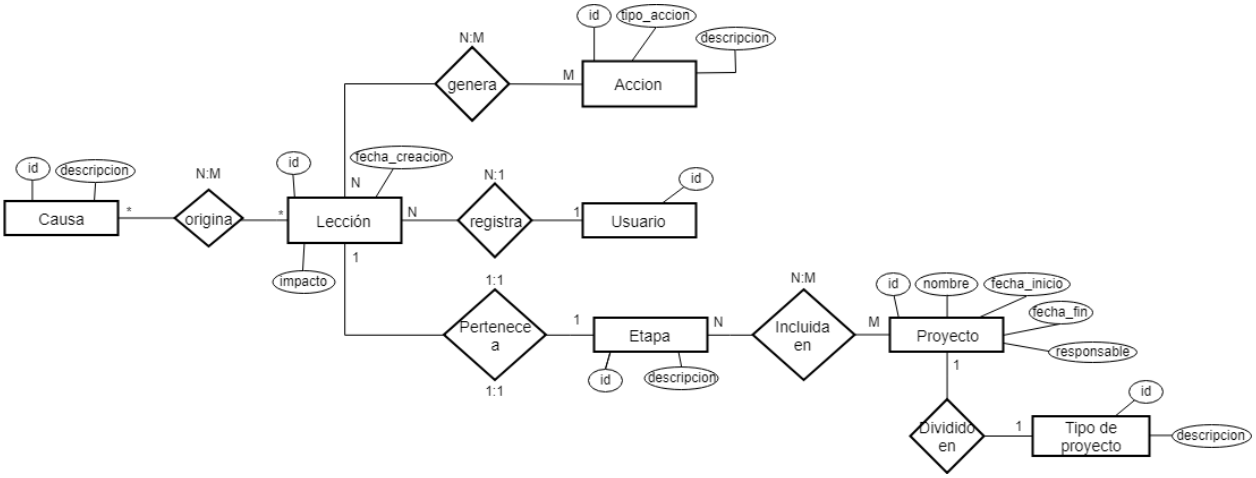


Fig. 9 Modelo Entidad-Relación propuesto
Fuente: Elaboración propia

3.3.2.3 Modelo Relacional

En la figura 10 se desarrollan las relaciones descrita en el modelo entidad relación y se normalizan las que son “muchos a muchos” (N:M).

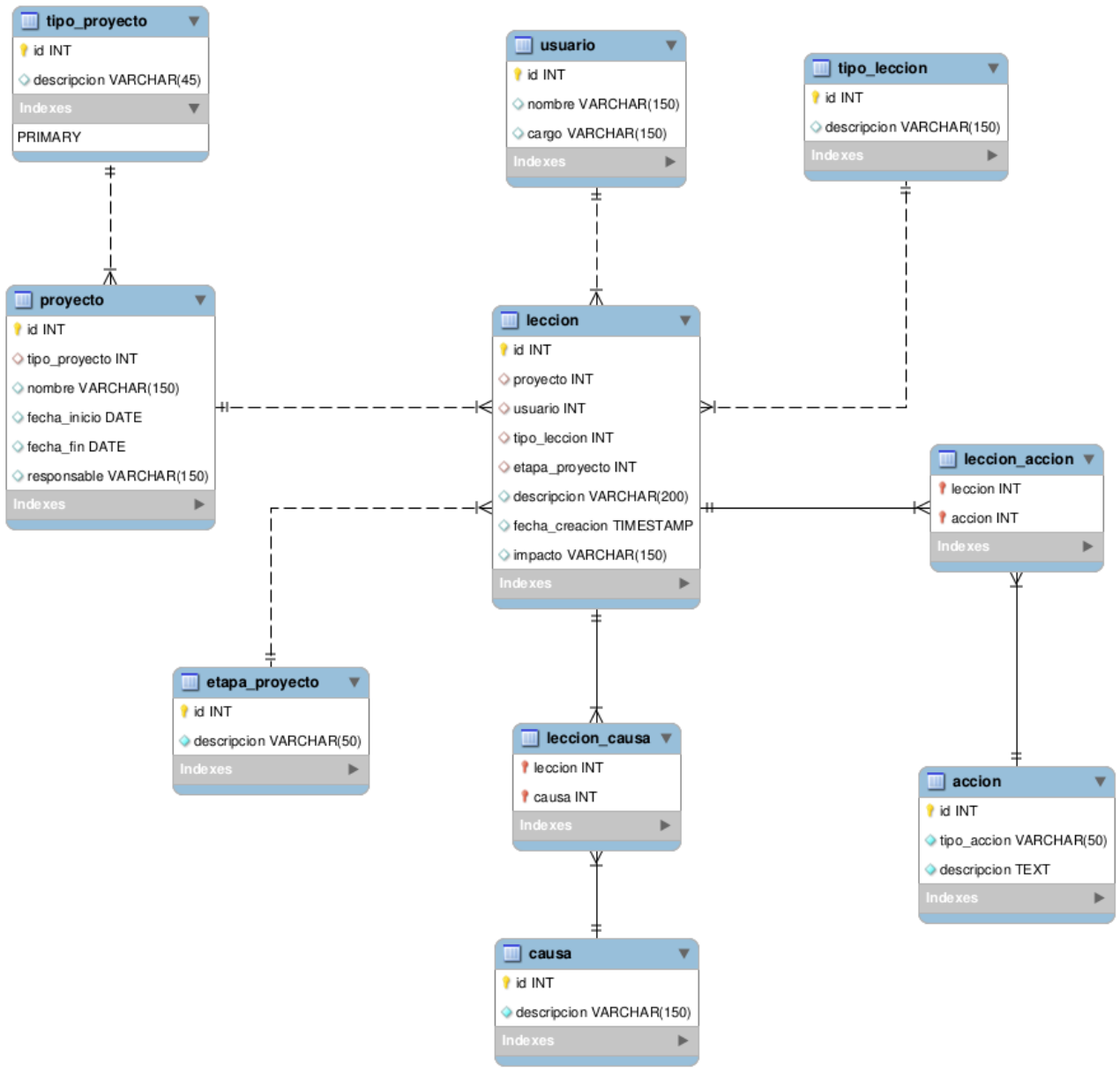


Fig. 10 Modelo Relacional propuesto
Fuente: Elaboración propia

- **Tipo proyecto**: Listado de los tipos de proyectos que se manejen en el área de desarrollo de software. Por ejemplo, proyectos de desarrollo, proyectos de mantenimiento.
- **Proyecto:** Representación de un proyecto a nivel del modelo de información que le permita asociarlo con varias lecciones aprendidas.
- **Usuario:** Usuario del sistema, encargado de ingresar o consultar lecciones aprendidas de los proyectos.

- **Tipo lección:** La clasificación de cada lección para facilitar su identificación ya sea vinculada a los productos y servicios prestados o vinculada a mejora de procesos internos del área.
- **Lección**: Se entiende por lección a la información útil generada o encontrada por un usuario en el desempeño de sus labores que pueda ser de utilidad para otros integrantes de la organización. En la tabla 1 se dan unos ejemplos de registro de lección.

Tabla 1. Ejemplos de registro de lecciones aprendidas
Fuente: Elaboración propia

id_lección	id_proyecto	id_usuario	tipo_leccion	etapa_proyecto	descripcion	fecha_creacion	causa	Impacto	tipo_accion	accion
1	1	1	Mejores prácticas	planeación	Es necesario alinear a la alta gerencia sobre los entregables y los puntos que definen el alcance del proyecto desde el día 0 del proyecto.	02/05/2016	Stakeholders del proyecto no estuvieron presentes durante las definiciones primordiales del alcance del proyecto.	Alcance planeado no coincida con expectativas del cliente	correctiva	En la preparación de reuniones importantes para las definiciones iniciales del proyecto comprometer la asistencia de stakeholders indispensables de todas las partes involucradas.
2	1	1	Mejores prácticas	planeación	"Agilizar el proceso de definición de entregables pensando más en lo que el cliente necesita desde escenarios tempranos del proyecto.	03/05/2016	los entregables planteados inicialmente no respondieron totalmente a las exigencias del cliente, para definirlas correctamente sería de gran ayuda un previo análisis de las necesidades por parte del equipo de consultoría	Tareas definidas no abarcaban la totalidad de la funcionalidad solicitada.	correctiva	Apoyo del equipo técnico en la planeación
3	2	1	Mejores prácticas	Ejecución	Siempre re agendar la reunión de seguimiento y no dejar pasar más de 2 semanas sin seguimiento.	12/09/2018	Seguimiento del proyecto no se hacía en las fechas establecidas y pasaban semanas sin hacerlo	No se evidenciaban ni socializaban retrasos ni dificultades a tiempo debido a falta de reuniones de planeación.	correctiva	Concientizar sobre la importancia de los seguimientos a clientes y miembros del equipo y mantener re agendamientos lo más pronto posible con todos los involucrados.
4	3	1	Mejores prácticas	Ejecución	Apoyar al cliente	16/11/2018	Pusieron a cargo de liderazgo del	Los temas tratados en las reuniones	Mejora	Prestar apoyo metodológico a la

					(técnicos, directores, soporte) siempre que existan vacíos en el entendimiento de la temática tratada.		proyecto de lado de cliente una persona poco experimentada en tecnología.	de seguimiento y la metodología aplicada no eran comprendidos por el director de proyectos del lado del cliente lo que hacía que muchos pendientes no fueran resueltos y entorpecía el desarrollo de las reuniones por falta de manejo de la metodología.		persona que lo necesite (del cliente o del equipo) con la finalidad de llevar a buen término todas las actividades realizadas.

- **Etapa del proyecto:** Clasificación de las diferentes etapas de un proyecto donde se puede originar una lección aprendida: análisis, diseño, desarrollo, aseguramiento de calidad, paso a producción.
- **Acción:** puede ser correctiva, preventiva o una recomendación, el cómo se abordó un evento específico que generó la lección aprendida. Por ejemplo, *el proyecto se encontraba en riesgo de retraso porque el equipo no tenía conocimientos sólidos de cómo implementar la solución,* la contratación de un experto fue la acción tomada para evitar que el riesgo se materializara.

3.3.2.4 Infraestructura

Arquitectura conceptual propuesta de la solución

En la tabla 2 se describen los 3 niveles propuestos para la arquitectura conceptual.

Tabla 2. Arquitectura conceptual propuesta de la solución
Fuente: Elaboración propia

Nivel	**Descripción**
NIvel de Interfaces	Front-end dedicado a recopilar información de lecciones aprendidas, herramientas actualmente utilizadas en gestión de las diferentes etapas del proyecto (Team Foundation, Mantis, MavenLink).
Nivel de Aplicaciones	Módulo back-end encargado de consultar y procesar la información proveniente de las diferentes interfaces, también se encarga de almacenar dicha información en el motor de bases de datos dedicado.
Nivel de Fuentes de Información	Además de las interfaces, para obtener información puede ser necesario consultar diferentes fuentes de información que no cuenten con una interfaz gráfica (procesos batch, servicios web expuestos por terceros, archivos planos), por lo que a nivel de aplicación. Estas fuentes varían dependiendo de la organización donde se aplique el modelo.

En la figura 11 se ilustra cómo se relacionan los componentes de la infraestructura propuesta.

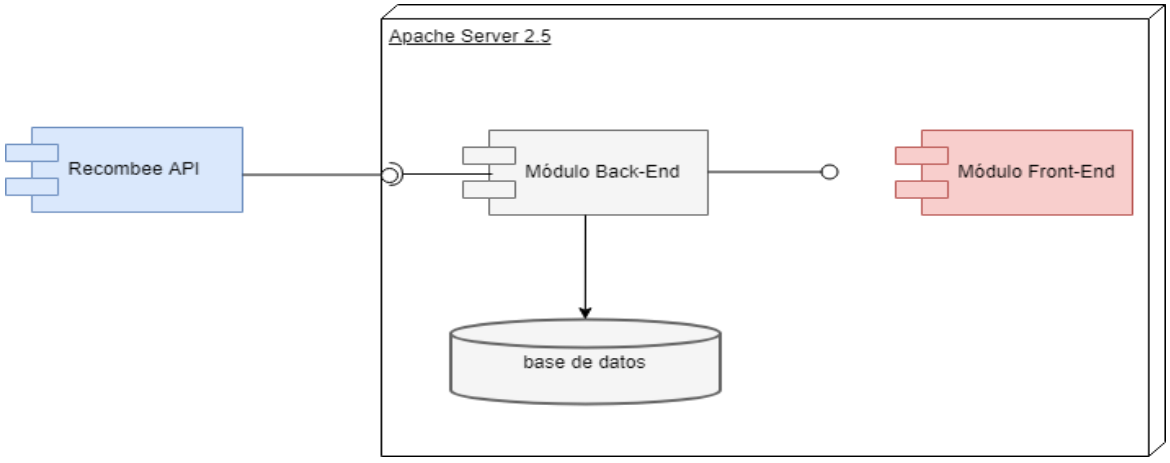


Fig. 11 Infraestructura Física
Fuente: Elaboración propia

Con los componentes identificados se pueden definir las tecnologías y herramientas de software, en la figura 12 se pueden ver aquellas que se utilizarán para diseñar y desarrollar la aplicación:

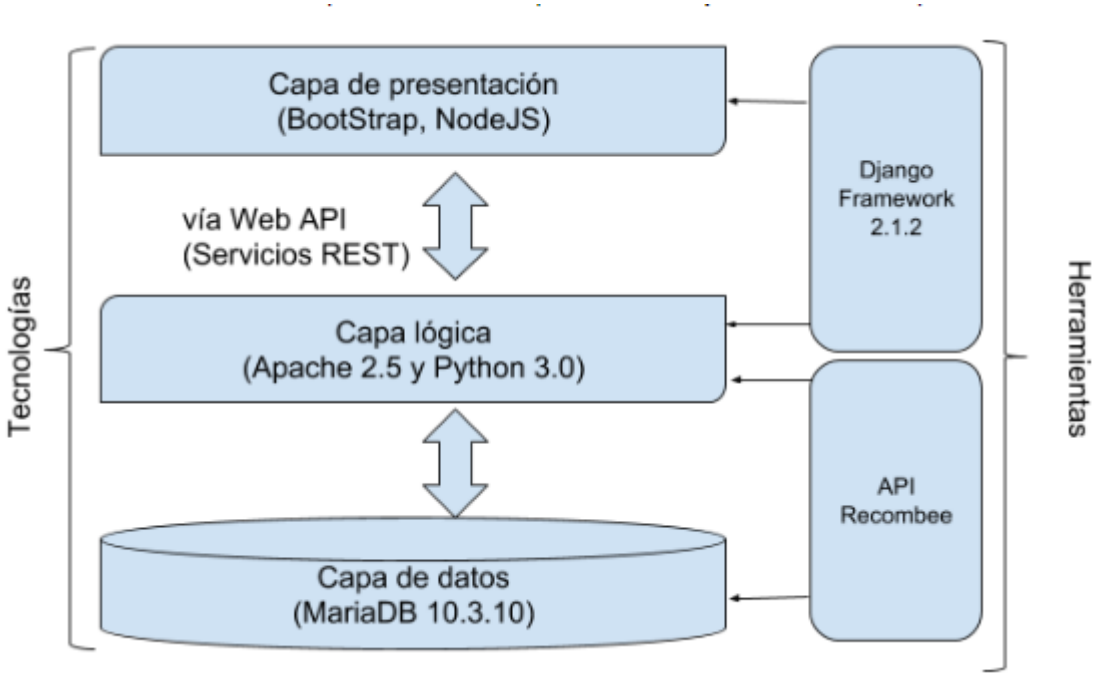


Fig. 12 Infraestructura Lógica del Modelo
Fuente: Elaboración propia

Frontend: Capa de presentación donde el usuario diligencia lecciones aprendidas, consulta y le son provistas las lecciones aprendidas recomendadas.

- **HTML 5:** Lenguaje de marcas de hipertexto, se usa para generar la estructura básica de cualquier página, este es entendido por cualquier navegador en sus versiones más recientes.
- **CSS3:** Hojas de estilo en cascada, se usan para dar estilo y mejorar la apariencia visual de los sitios web.
- **Javascript:** Se usa para ejecutar código en el lado del cliente.

Backend: Capa donde se aplica la lógica de negocio, se consulta y persisten datos en la base de datos y se ejecuta el llamado a la API de recomendaciones como servicios Recombee.

- **Apache 2.5:** El servidor web HTTP en el cual se alojará la aplicación hecha en Python.
- **Python 3.0:** El lenguaje de programación en el cual se desarrollará toda la lógica del modelo.

Datos: Capa de persistencia de la información.

- **MariaDB 10.3.10:** Última versión del motor de bases de datos con el cual se gestionará la información mediante lenguaje SQL.

Django Framework: De acuerdo con la documentación de mozilla developers se define como **"**Un framework web de alto nivel que permite el desarrollo de sitios web rapido, seguro y mantenible. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago." (MDN Web docs, 2018)

Recombee: Es una API implementada por científicos de datos, usada como sistema de recomendación y en el caso del modelo presentado se usará en la capa de Machine Learning. La razón por la cual se escoge esta API es principalmente su facilidad de uso y como los datos no son sensibles es viable tercerizar este servicio.

De acuerdo a la documentación de los servicios prestados por Recombee podemos decir que, al ser un sistema personalizable de recomendaciones apto para diferentes tipos de negocio. "Recombee usa una variante de la recomendación basada en reglas, lo que permite "minar" las reglas sobre la marcha" Kordík (2016) es decir el modelo es actualizado constantemente cuando se realiza algún cambio. El algoritmo base que usa Recombee es conocido como "Recomendación de reglas ponderadas", de acuerdo a la documentación de la herramienta P. Kordík (2018), el algoritmo genera un conjunto de reglas a-priori de la matriz de interacciones, que generalmente es inmensa y es usada

por el algoritmo para comparar los datos y obtener las recomendaciones que más se acercan a la información con la que se cuenta, en el caso particular de este proyecto, a las lecciones aprendidas. Las reglas con un soporte suficiente son usadas para generar los elementos de la recomendación.

De acuerdo al alto soporte científico del algoritmo desarrollado por Recombee, relevancia en el mercado actual y facilidad de uso, se considera una opción viable y tangible para su uso como apoyo en la implementación del modelo propuesto

Capítulo 4 Evaluación y discusión del modelo de Gestión del Conocimiento

De acuerdo los resultados de la investigación y al modelo propuesto consecuencia de la misma, se puede decir que la gestión del conocimiento es de suma importancia para cualquier organización y por lo tanto cada departamento debería realizar gestión de conocimiento, de esta forma es posible hacer más eficientes los procesos internos, de lograrse, podría generar una optimización de costos y de tiempo. Por lo tanto, las lecciones aprendidas deben dejar de ser un bien pasivo en las organizaciones y deben empezar a ser registradas y utilizadas de forma proactiva, ya que en ellas se evidencia la información obtenida de la experiencia que puede apoyar a proyectos nuevos y evitar la repetición de errores ya superados.

Las organizaciones deben registrar sus lecciones aprendidas de una manera estandarizada para que esta pueda ser consultada bajo los mismos criterios con los que fue registrada, sin importar la persona o el proyecto que esté realizando la labor.

Para generar valor con el modelo propuesto se decidió usar una tecnología que actualmente está siendo explotada en plataformas como Netflix o Spotify, con esto se hace referencia a Machine Learning, esto hace posible que el aprendizaje se transmita eficientemente mediante un sistema de recomendaciones. Así, el conocimiento llega directamente a quien lo necesite de acuerdo con las características de su proyecto, de esta forma se evita la tarea de realizar búsqueda y depuración de resultados, de otra forma se tendría un repositorio de información. Se tuvo en cuenta el proceso de transferencia de conocimiento planteado en distintos modelos de gestión de conocimiento, a partir de ellos se generó un modelo propio.

4.1 Conclusiones

De acuerdo a los modelos de conocimiento y de gestión de conocimiento consultados se pudo generar un modelo propio, extrayendo los conceptos más relevantes de cada uno, se toma como referencia principal el modelo de espiral de TIC propuesto por Pérez y Dressler en el cual se le da una especial importancia al factor humano y al uso de las últimas tecnologías de la información para facilitar la transferencia del conocimiento, en ese modelo se presentan las diferentes tecnologías y la etapa de transferencia del conocimiento a la que se pueden aplicar, esto permitió complementar el modelo con la capa de Machine Learning para hacer una transferencia más eficiente.

Después de analizar los mecanismos de captura y almacenamiento de lecciones aprendidas en dos empresas diferentes, se evidencia que apuntan más hacia un repositorio de lecciones aprendidas cuyo acceso está restringido por lo que el conocimiento no se transfiere, por lo menos no de la manera que los modelos consultados en el desarrollo del documento proponen. El conocimiento debería ser para el que lo necesite y debería estar disponible en cualquier momento. Así mismo, dicha información era consultada ocasionalmente y la dirección de proyectos de ambas organizaciones no explotaba como debía esta información, esto último se evidenció en la ejecución de proyectos posteriores.

De acuerdo a los formatos usados para el registro de lecciones aprendidas y al modelo de espiral de TIC de Pérez y Dressler, en el concepto de aplicación de TICS con énfasis en el factor humano, se identificaron los actores que participan del levantamiento de lecciones aprendidas, así como los posibles usuarios de esta información, en vista que tanto en el modelo como en los formatos el capital humano es prácticamente el punto central, se definen 3 tipos de usuario los cuales hacen parte integral del área de dirección de proyectos. Se definieron todos los factores que generan información relevante para las lecciones aprendidas y que a su vez son usadas para gestionar el conocimiento, de acuerdo al mismo modelo se evidencia cómo las diferentes tecnologías pueden ser de ayuda para facilitar el proceso de transferencia del conocimiento.

Con los actores y factores definidos se pudo diseñar un modelo entidad relación básico y un modelo relacional que posteriormente se usarán para implementar una base de datos, su finalidad será el registro y gestión de las lecciones aprendidas que serán el insumo para el mecanismo de divulgación del aprendizaje.

El mecanismo escogido para la divulgación es un algoritmo de recomendación usando la tecnología de Machine Learning, con esto se garantiza que el conocimiento va a ser distribuido eficientemente, con la ventaja que los interesados no tendrán que buscar la información, esta llegará directamente a ellos. La utilización de Machine Learning en

este escenario de lecciones aprendidas permite que el sistema de almacenamiento de lecciones aprendidas vaya más allá, aprendiendo de las características de proyectos anteriores y generando un listado de posibles recomendaciones con base en esta información.

4.1.1 Trabajos de investigación futuros

Dentro del desarrollo del proyecto se encontraron varios temas de interés, primordialmente a causa del tiempo de ejecución del mismo no fue posible realizar un prototipo por lo que en primera instancia este será el objetivo primordial de los trabajos futuros, se considera que en este punto se cuenta con una base sólida para iniciar el proceso de implementación.

No obstante, se encontraron varios temas adicionales de interés que se listarán a continuación:

- Algoritmos de Machine Learning y su aplicación en herramientas de uso diario en la organización.
- Profundización de la relación de la inteligencia artificial con respecto a los repositorios de conocimiento de diferentes tipos de organizaciones.
- Ampliar el campo de aplicación de algoritmos inteligentes tratando de mejorar los existentes o creando uno nuevo.
- Evaluar la posibilidad de crear un sistema de uso de lecciones aprendidas, transversal en la organización que se alimente de todos los proyectos sin importar el área en que se produzcan.
- Evaluar un sistema de lecciones aprendidas abierto para todas las organizaciones públicas, que permita el uso y colaboración de diferentes industrias y empresas.

Referencias Bibliográficas

- Bergerson, B. (2003). Essentials of Knowledge Management. New Jersey: Wiley.
- Choo, C. W. (2006). The Knowing Organization. New York: Oxford University Press.
- Dalkir, K. (2005). Knowledge Management In Theory And Practice. Burlington: ELSEVIER.
- Demicheli, G. (s.f.). http://web.usbmed.edu.co/. Obtenido de http://web.usbmed.edu.co/usbmed/egresados/docs/memorias/ORGANIZACIONES%20INTELIGENTES.pdf
- Díaz, M. S. (s.f.). http://eprints.rclis.org. Obtenido de http://eprints.rclis.org/7964/1/aci060605.pdf
- Farfán Buitrago, D. Y., & Garzón Castrillón, M. A. (2006). La gestión del conocimiento. Bogotá: Editorial Universidad Del Rosario.
- http://gestrategica.org. (s.f.). Obtenido de http://gestrategica.org/guias/aprendizaje/como_a.html
- http://www.bdigital.unal.edu.co. (s.f.). Obtenido de http://www.bdigital.unal.edu.co/18937/1/14879-44820-1-PB.pdf
- http://www.uky.edu. (2002). Obtenido de http://www.uky.edu/~gmswan3/575/Holsapple_and_Joshi_2002.pdf
- https://www.mineducacion.gov.co. (s.f.). Obtenido de https://www.mineducacion.gov.co/1759/articles-324587_archivo_pdf_4_Gestion_Conocimiento_MEN.pdf
- Muzard, J. (2011). http://www.a-i-a.com. Obtenido de http://www.a-i-a.com/auladigital/ArticuloGC-JM-ESP.pdf
- Nonaka, I., & Takeuchi, H. (marzo de 1999). https://eva.fcs.edu.uy. Obtenido de https://eva.fcs.edu.uy/pluginfile.php/86017/mod_resource/content/1/Nonaka%20y%20Takeuchi_cap%203.pdf
- Pereira Alfaro, H. (2011). http://www.cegesti.org. Obtenido de http://www.cegesti.org/exitoempresarial/publicaciones/publicacion_135_310111_es.pdf

- Perez, D., & Dressler, M. (2006). https://upcommons.upc.edu/. Obtenido de https://upcommons.upc.edu/bitstream/handle/2099/2945/Tecnologias%20de%20la%20informacion.pdfhttps://upcommons.upc.edu/bitstream/handle/2099/2945/Tecnologias%20de%20la%20informacion.pdf
- United Nations Development Programme. (2014). http://www.undp.org/. Obtenido de http://www.undp.org/content/dam/undp/library/capacity-development/English/UNDP%20Knowledge%20Strategy%20Report%202502-2%20LR%202,7MB.pdf
- Yew Wong, K., & Aspinwall, E. (2004). Knowledge Management Implementation Frameworks: A Review. Wiley InterScience, 93-104.
- (http://conceptodefinicion.de) (https://definicion.de, s.f.)
- Henninger, S. Automated Software Engineering (1997) 4: 319. Obtenido de https://doi-org.bdigital.udistrital.edu.co/10.1023/A:1008679010073
- Kozik, R., Choraś, M., Puchalski, D. et al. J Ambient Intell Human Comput (2018). https://doi-org.bdigital.udistrital.edu.co/10.1007/s12652-018-0784-5
- Satapathy, S.M. & Rath, S.K. Innovations Syst Softw Eng (2017) 13: 191. https://doi-org.bdigital.udistrital.edu.co/10.1007/s11334-017-0288-z
- Portugal Ivens, Alencar Paulo, Cowan Donald. The use of machine learning algorithms in recommender systems: A systematic review (2018). Obtenido de: https://ac-els-cdn-com.bdigital.udistrital.edu.co/S0957417417308333/1-s2.0-S0957417417308333-main.pdf?_tid=1fbec7ca-f771-475a-adb4-be6c0d13d176&acdnat=1538949041_bf71738a06149b8d41756b68a10d638f
- T. Dingsøyr, "Knowledge Management in Medium-Sized Software Consulting Companies," Empirical Software Engineering, vol. 7, p. 383–386, 2002.
- Perez, D., & Dressler, M. (2006). https://upcommons.upc.edu/. Obtenido de https://upcommons.upc.edu/bitstream/handle/2099/2945/Tecnologias%20de%20la%20informacion.pdfhttps://upcommons.upc.edu/bitstream/handle/2099/2945/Tecnologias%20de%20la%20informacion.pdf
- Higuita Correa, J. D. (n.d.). https://www.ceipa.edu.co. Retrieved from https://www.ceipa.edu.co/lupa/index.php/lupa/article/view/57/105
- Dueñas Sanchez, H. (s.f.). www.ceipa.edu.co. Obtenido de www.ceipa.edu.co/lupa/index.php/lupa/article/view/56/104

- http://cv.uoc.edu. (s.f.). Obtenido de http://cv.uoc.edu/UOC/a/moduls/90/90_331/web/main/m1/v1_1_4a.html
- https://www.ecured.cu. (s.f.). Obtenido de https://www.ecured.cu/Desarrollo_de_software
- http://conceptodefinicion.de. (s.f.). Obtenido de http://conceptodefinicion.de/modelo/
- https://definicion.de. (s.f.). Obtenido de https://definicion.de/base-de-datos/
- O. Sabri and F. Alfifi, "Integrating knowledge life cycle within software development process to produce a quality software product," 2017 International Conference on Engineering and Technology (ICET), Antalya, 2017, pp. 1-7.
- Banco Interamericano de desarrollo “Cómo identificar lecciones aprendidas”, https://www.repositoriopncvfs.pe/wp-content/uploads/2017/03/Como-identificar-Lecciones-Aprendidas.pdf
- Rowe, S. F. & Sikes, S. (2006). Lessons learned: sharing the knowledge. Paper presented at PMI® Global Congress 2006—EMEA, Madrid, Spain. Newtown Square, PA: Project Management Institute.
- Organización Panamericana de La Salud, Metodologías de la OPS/OMS paraintercambio de información y gestión del conocimiento en Salud - Lecciones Aprendidas, Organización Panamericana de La Salud.
- Banco de Desarrollo Interamericano, BID. “Cómo documentar lecciones aprendidas”, https://blogs.iadb.org/abierto-al-publico/2015/01/15/como-documentar-lecciones-aprendidas/.
- P. Kordík, "https://medium.com," 12 july 2016. [Online]. Available: https://medium.com/recombee-blog/recommender-systems-explained-d98e8221f468.
- P. Kordík, "https://medium.com," 3 june 2018. [Online]. Available: https://medium.com/recombee-blog/machine-learning-for-recommender-systems-part-1-algorithms-evaluation-and-cold-start-6f696683d0ed.