

Desarrollo de un software web para la generación de planes de gestión de riesgos de software

Valentina P. Castro-Rivera⁽¹⁾, Raúl A. Herrera-Acuña⁽²⁾ y Marco A. Villalobos-Abarca⁽²⁾

(1) Ingeniera de Desarrollo, Kuvemar Desarrollo Tecnológico, Membrillar 2623, Arica-Chile.
(correo-e: valentina.pcr@gmail.com)

(2) Facultad de Ingeniería, Depto. de Ingeniería en Comp. e Inf., Univ. de Tarapacá, Casilla 7D, Arica-Chile
(correo-e: rherrera@uta.cl; mwillalo@uta.cl)

Recibido Nov. 11, 2019; Aceptado Ene. 14, 2020; Versión final Feb. 21, 2020, Publicado Jun. 2020

Resumen

Este artículo presenta el diseño y desarrollo de un sistema web para el apoyo en la generación de planes de gestión del riesgo para el desarrollo de software, orientado a organizaciones emergentes de desarrollo tecnológico, a emprendedores nuevos y establecidos, en el sector de micro y pequeñas empresas del área. El objetivo principal de este sistema es mejorar las posibilidades de éxito en el desarrollo y concreción de nuevos productos de software de dichas empresas, mediante una serie de herramientas de apoyo, como el uso de *coaching* experto, trabajo colaborativo y la administración de proyectos, basado en tecnología web. El sistema se implementó usando tecnologías web de punta y se desarrolló utilizando un método evolutivo incremental, similar a SCRUM. Nuestro sistema se probó exitosamente en una empresa local de desarrollo de software. Esto demuestra que nuestro sistema es una herramienta eficiente para la generación de planes de gestión del riesgo para el desarrollo de software.

Palabras clave: planes de gestión, riesgos de software, software de gestión

Development of a web software to generate management plans of software risks

This article describes the design and development of a web system for the creation of risk management plans. Our web software aims to supporting management plan creation for new tech-development organizations, for new and established entrepreneurs, and for micro and small start-ups. The main objective of our web software is to improve the probability of success in the release of new software products. Our web software provides a series of support tools such as expert coaching, collaborative work, and risk administration during the developmental process. Our web software was implemented using current web technologies and was developed using a method similar to SCRUM. It tested successfully in a real-life setting in a local software development company. This demonstrates that our system is an efficient tool for assisting in the creation of risk management plans in the real world.

Keywords: management plans, software risks, software for management

INTRODUCCIÓN

La gestión de riesgos es un proceso muy importante en el desarrollo de proyectos de software, Por lo general, el equipo a cargo no hace nada frente a los riesgos hasta que algo malo sucede. Pero existen muchos factores que pueden alterar los planes que afectarán la ejecución del proyecto, ya sea que ocurran modificaciones en los requisitos debido a nuevas necesidades del cliente, cambios en el equipo de trabajo que afectan a la calendarización del proyecto o la presión de la competencia que pueden ser una amenaza para el desarrollo normal del proyecto (Sommerville, 2005). Un estudio realizado por KLCI en 268 organizaciones desarrolladoras de software de todo el mundo, indicó que la gran mayoría de los participantes valoran la aplicación de las prácticas de la gestión de riesgos, en el cual un 97% de los encuestados respondieron que cuentan con procedimientos para identificar y evaluar el riesgo; y un 80% identificó prevenir y evitar problemas como un beneficio primario (Sarigiannidis, y Chatzoglou, 2011). Además, los resultados indicaron que un 3% de las organizaciones participantes no utilizan ningún método para gestionar el riesgo, un 18% lo hace de manera informal gestionando los riesgos a medida que aparecen, mientras que un 37% lo hace discutiendo con el equipo de trabajo. Por otro lado, un 28% de las organizaciones emplea procedimientos periódicos para identificar y cuantificar los riesgos, y solo un 14% utiliza un método formal.

Una de las razones más comunes para el uso de un enfoque informal, es que las organizaciones son jóvenes o inmaduras, es decir, recién están comenzando a desarrollar software, por lo que prefieren involucrar a todos los miembros del equipo para ayudar a identificar y evaluar los riesgos. Por otra parte, quienes utilizan un enfoque periódico argumentan que es un paso esencial para el buen desarrollo del proyecto, ya que mantiene al cliente informado y controla los costos (Sarigiannidis, y Chatzoglou, 2011).

Por otra parte, el estudio realizado por *Standish Group (Chaos Report)* el año 2015 (Hastie y Wojewoda, 2016) entrega una visión del estado de la industria de desarrollo de software. Se consideraron 50.000 proyectos de todo el mundo que van desde pequeñas mejoras a los sistemas masivos e implementaciones de re-ingeniería. Los resultados de dicho estudio muestran que el porcentaje de proyectos exitosos alcanza el 29%, por otro lado los fallidos ocupan un 19%, es decir, estos proyectos nunca se concretaron o fueron simplemente un fracaso. En cambio, 52% los proyectos puede que hayan tenido éxito o hayan fracasado pero durante el desarrollo del proyecto, se presentaron con más inconvenientes y problemas. Los resultados del año anterior de este estudio (Johnson, 2015), además, indicó los factores por los cuales el proyecto tendía a fracasar. Muchos de estos coinciden con los factores de riesgos que pueden ser un problema pero que no se miden antes de comenzar un proyecto. Estos pueden ser el cambio en los requisitos, poca participación de los usuarios, nuevas tecnologías, entre otros.

Así, de acuerdo a lo presentado en estos estudios, son muchos los factores que hacen que un proyecto falle, principalmente por falta de detección temprana de los riesgos que pueden afectar el desarrollo normal de un proyecto de software. De lo anterior queda en evidencia los beneficios de contar con una herramienta de software para apoyar la gestión de riesgos en el desarrollo de proyectos Informáticos. Así, esta herramienta gestionaría los riesgos de un proyecto desde las etapas tempranas y durante todo el ciclo de vida y, de esta forma, se podría administrar de manera proactiva cada acontecimiento que se pudiera presentar durante el análisis, diseño, codificación, prueba y mantenimiento del software. Además, los planes de riesgo se podrían ajustar a un estándar específico, como el IEEE 1540-2001 o superior.

La existencia del riesgo siempre conlleva dos características: la incertidumbre si ocurrirá o no algún imprevisto, o qué tipo de consecuencia traerá si se hace real (Wong et al., 2010). En un proyecto de software los riesgos se consideran una amenaza para quienes conforman el equipo de trabajo, para el proyecto y el software que está en ejecución. Es por esto que al momento de analizar los riesgos se debe definir a que categorías pertenecen. A continuación, se presentan tres categorías (Pressman, 2002): (i) Riesgos del proyecto: Amenazan la planificación del proyecto, retrasándola y aumentando los costos. Ocurren problemas de falta de personal y recursos, problemas con los clientes y los requisitos; (ii) Riesgos técnicos o del producto: Amenazan la calidad y en el desarrollo del software. Ocurren problemas en el diseño, implementación, pruebas y mantenimiento. La falta de tecnología necesaria también es un factor; y (iii) Riesgos del negocio: Amenazan la viabilidad del software. Ocurren problemas asociados a la real necesidad del producto, es decir, construir algo excelente y que nadie lo requiera.

La importancia de la gestión de riesgos radica principalmente en la minimización de la incertidumbre que se presenta en cada proyecto de software, ya que muchas veces los requerimientos son confusos y cambian a medida que el proyecto avanza, existen errores al estimar los tiempos y los recursos necesarios. La clave es actuar de forma proactiva y anticiparse a eventuales riesgos que pongan en peligro el proyecto. Si llegasen a ocurrir, se debe crear planes de contingencia para evitarlos o disminuir el efecto que pueda producir en el proyecto.

GESTION DE RIESGOS

En la Figura 1, se muestra la relación existente entre el problema, solución y aspectos tecnológicos involucrados en el desarrollo. En el primer nivel, se encuentra la problemática de la falta de detección temprana de los riesgos que pueden afectar el desarrollo normal de un proyecto de software. El segundo nivel, hace referencia a la solución, la cual consiste en una plataforma web que permita elaborar y administrar planes de gestión de riesgos de software. La plataforma web fue concebida en tres niveles o capas, teniendo la presentación como capa superior que estará encargada de la interacción entre sistema y usuario a través de una interfaz gráfica intuitiva, es decir, entendible y fácil de usar. La capa de presentación se comunica únicamente con la capa intermedia, donde se establecen todas las reglas y funcionalidades que deben cumplirse. Esta capa, se comunica con la de presentación para recibir las solicitudes y presentar los resultados. La última capa, permite el acceso a datos para solicitar al gestor de bases de datos almacenar o recuperar información.

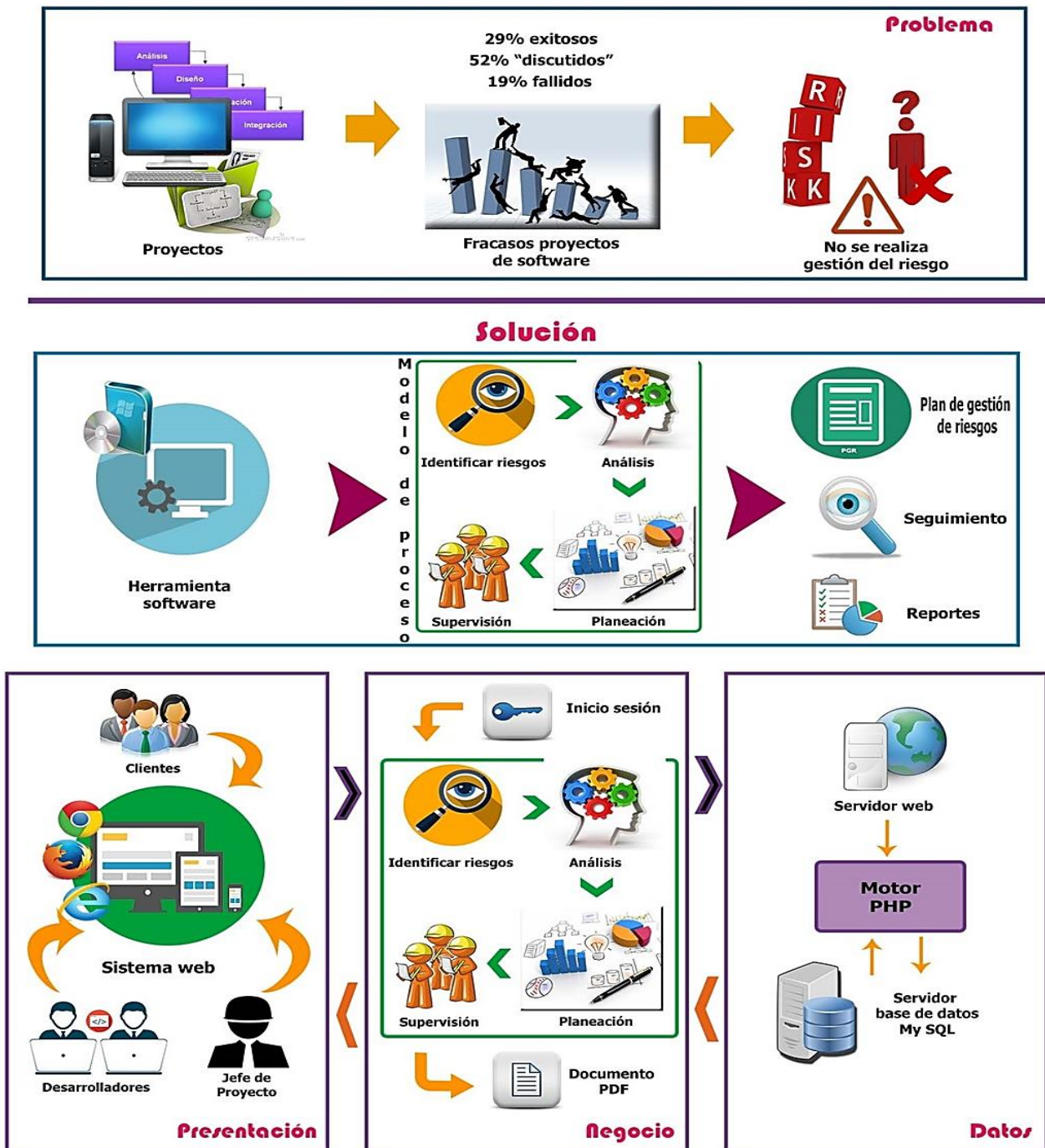


Fig. 1: Diagrama del problema, solución y aspectos tecnológicos

La gestión de riesgos es un proceso iterativo que se debe aplicar durante todo el desarrollo del proyecto, cada vez que aparece nueva información que implique a un riesgo, se deberá realizar un nuevo análisis y cambiar las prioridades de ser necesario. En la Figura 2, se muestra un proceso de gestión de riesgos.

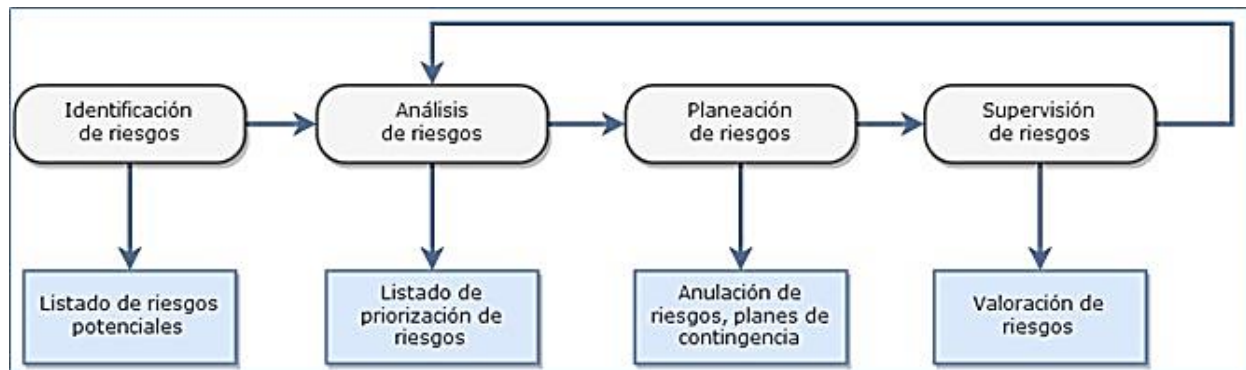


Fig. 2: Diagrama de un proceso de gestión de riesgos del software (Sommerville, 2005).

a. Identificación de riesgos: En esta etapa, se describen los posibles riesgos que puede presentar el proyecto, no se valoran ni se priorizan. Por lo general, la identificación se realiza haciendo una lluvia de ideas o en base a la experiencia; *b. Análisis de riesgos.* En esta etapa, se construye una tabla que incluye los riesgos identificados, la probabilidad de ocurrencia y el impacto que se generará en el proyecto. Una vez completada la tabla, se ordena los riesgos desde la más alta probabilidad y el más alto impacto hacia abajo. Luego se traza una línea de corte que divide la tabla en dos e implica que se debe prestar atención a los riesgos que se encuentran por sobre la línea, mientras que los riesgos que quedan bajo la línea, deben ser reevaluados por el equipo. El número exacto de riesgos que estén sobre la línea va a depender de la cantidad de riesgos que existan, pero se recomienda analizar los 10 más altos. Los riesgos que tienen entre un muy alto impacto a moderado y gran probabilidad de ocurrencia, deben ser muy bien estudiados, al igual que los que tienen un bajo impacto pero una gran probabilidad de ocurrir. En cambio, los que tienen gran impacto y pocas probabilidades de ocurrir no deberían tomar tanto tiempo en ser analizados; *c. Planificación de riesgos.* En esta etapa se deben planificar estrategias para gestionar los riesgos si llegasen a ocurrir. Las estrategias dependerán de la experiencia del grupo encargado de la gestión de riesgos, ya que no existe un proceso exacto que indique como hacerlo; *d. Supervisión de riesgos.* La etapa de supervisión de riesgos es un proceso continuo, y en cada progreso de la gestión, los riesgos deben ser considerados. Es por esto que se debe medir si la probabilidad del riesgo aumenta o disminuye.

El plan RSGR se utiliza como una estrategia para organizar los pasos de gestión del riesgo (Barafort et al., 2017). Cada riesgo que se encuentre por sobre la línea de corte será considerado como parte del plan y éstos serán los que pasen a la etapa de planificación (punto c). Los planes de riesgos no son más que documentos donde se muestra la información del riesgo, como nombre, tipo, descripción, valoración de probabilidad, valoración de impacto, refinamiento, supervisión, plan de acción y plan de contingencias. Si bien existen herramientas que apoyan parte del proceso de gestión de planes de riesgos, no todas aplican tecnologías modernas para su implementación, y tampoco cuentan con un usuario profesional a cargo de validar planes de riesgos (*coach*). Algunos ejemplos de trabajo relacionados se presentan a continuación. Posteriormente, se muestra una comparativa entre ellas y el sistema que se presenta en este artículo.

a. Proyecto de elaboración de la metodología de gestión de riesgos en proyectos de desarrollo de software para la empresa consultora CV3. En el trabajo desarrollado por Eddy Castillo (Castillo, 2009), se elaboró una metodología para gestionar los riesgos para la empresa CV Tres Consultores y Asociados. Se desarrollaron los procesos necesarios para la gestión de los riesgos, al igual que plantillas e instructivos necesarios para la aplicar la metodología y se desarrolló un plan de capacitación para preparar a los administradores de proyectos. No se desarrolla una herramienta software para gestionar los riesgos; *b. Herramienta para la gestión de riesgos en proyectos de software.* En el trabajo desarrollado por Rodolfo Bertone (Bertone et al., 2010), se elaboró una herramienta de software para gestionar los riesgos en proyectos de desarrollo de sistemas informáticos, la herramienta permite realizar la identificación, análisis, planificación, seguimiento y control de riesgos durante todo el ciclo de vida del proyecto; *c. Guía para apoyar la priorización de riesgos en la gestión de proyectos de tecnologías de la información.* En el trabajo desarrollado por Luisa Mosuquera (Mosuquera et al., 2013), se elaboró una guía que describe detalladamente los pasos para realizar la priorización de riesgos, la que tiene como fin disminuir la subjetividad con la que hasta ahora se realiza este proceso de gestión. Además, se desarrolla un prototipo software donde se aplica cada paso propuesto; *d. Plan de Riesgos para la implementación, desarrollo y mantenimiento de componentes de Web 2.0 en Bibliotecas.* En el trabajo desarrollado por Horacio Daniel Kuna

(Kuna et al., 2008), se elaboró una metodología para realizar la gestión de riesgos, efectuando tareas de identificación, administración y mitigación. El proceso se aplica en el Centro de Documentación Entidad Binacional Yacyretá, Argentina; e. *Un análisis crítico de las técnicas de gestión de riesgos de software en sistemas a gran escala*. En el trabajo desarrollado por Maruf Pasha (Pasha et al., 2018), se describe una lista de herramientas necesarias para gestionar el riesgo en sistemas de software a gran y pequeña escala. Además, presenta una lista de factores de riesgos recopilada de diferentes literaturas existentes, y se categorizan según las diferentes fases del proceso de desarrollo de software. Finalmente, propone una metodología para mitigar riesgos en sistemas a gran escala; f. *Planificación cuantitativa y gestión de riesgos y del desarrollo ágil de software*. En el trabajo desarrollado por Kamran Ghane (Ghane, 2017) se describe un modelo de gestión de riesgos enfocado en la utilización de metodologías ágiles para el desarrollo de software. Propone un sistema que permite calcular el riesgo en función de atributos del proyecto, como tiempo, costo, alcance y calidad.

En la Tabla 1, se muestra una comparación entre los trabajos relacionados y el sistema desarrollado. Se destaca que si bien apoyan parte del proceso de gestión de planes de riesgos de software, no todas aplican tecnologías modernas para su implementación, no están disponibles para su utilización y tampoco describen la arquitectura de diseño.

Tabla 1: Estudios y herramientas existentes VS herramienta propuesta

Trabajo relacionado	Posee herramienta	Usa tecnologías modernas	Cubre todo el proceso	Sigue un estándar	Es de libre disposición	Presenta una Arquitectura diseño	Dependencia del modelo desarrollo
a	no	No aplica	Si	no	No aplica	No aplica	no
b	si	No	Si	no	no	No hay información	no
c	si	No	No	no	no	No hay información	no
d	no	No aplica	Si	si	No aplica	No aplica	no
e	no	No aplica	si	si	No aplica	No aplica	no
f	no	No aplica	si	si	No aplica	No aplica	si
Sistema desarrollado	si	si	si	si	si	Modelo Vista Controlador (MVD)	no

MATERIALES Y MÉTODOS

El enfoque utilizado en este trabajo fue la investigación aplicada, en virtud de lo cual, se presenta una solución tecnológica a un problema concreto, como en los trabajos de Mamani (Mamani et al., 2017), Cabarcas (Cabarcas et al., 2015), Gómez (Gómez et al., 2016), Coca (Coca et al., 2016), Alarcón-Aldana (Alarcón-Aldana et al., 2016), Quintero (Quintero et al., 2017) y Vidal (Vidal et al., 2017). Para esto, se utilizó un método evolutivo incremental, similar a SCRUM (Naz et al., 2016), el cual se describe gráficamente en la Figura 3. Este método está dirigido por casos de uso e incluye las etapas de especificación de requisitos, análisis y diseño, implementación y pruebas del software; permitiendo asumir que los requerimientos pueden cambiar en cualquier momento del ciclo de vida del software, además que se va desarrollando por subconjunto de requisitos especificados, para que en posteriores versiones el sistema se incremente con nuevas funcionalidades que satisfagan las necesidades requeridas.



Fig. 3: Diagrama del método evolutivo incremental del software

Para la implementación del software, se utilizaron distintas tecnologías, tanto para la planificación de requerimientos, diseño del prototipo del sistema, diseño del modelo lógico relacional, un entorno de desarrollo y un *framework* de aplicaciones web en PHP (Lancor y Katha, 2013; Olanrewaju et al., 2015; Gómez, 2014; Haughee, 2013), basado en el patrón de diseño Modelo Vista Controlador (MVC) (Hussain et al., 2017; González y Romero, 2014), MySQL para las consultas a la base de datos (Ni et al, 2018), JavaScript, como también la librería jQuery (Castillo, 2017) y Bootstrap (Laazari et al., 2019), para un mayor grado de calidad en la interfaz de usuario, permitiendo que ésta sea intuitiva y dinámica. En la Tabla 2, se puede apreciar el listado de las tecnologías y lenguajes de programación utilizados para el desarrollo del sistema. Cabe mencionar que todos son de costo cero y uso libre, excepto las tecnologías Trello (Johnson, 2017) y Axure (Krahenbuhl, 2015).

Tabla 2: Herramientas usadas en la implementación del software

Nombre	Descripción
MySQL Workbench	Para el diseño del modelo lógico relacional. Base de datos.
Framework Yii	Framework para la implementación del proyecto, utilizando el patrón de diseño MVC.
Sublime Text, Axure RP 8	Entorno de desarrollo para el proyecto. Elaboración del prototipo del proyecto.
WampServer	Servidor local en un único computador.
Librería JQuery, Bootstrap	Para una mayor calidad en la interfaz de usuario.
PHP, JavaScript, MySQL	Lenguajes de programación utilizados para el desarrollo del sistema.
Trello	Plataforma web para organizar de mejor manera el proyecto.

RESULTADOS

Los resultados se describen en varias subsecciones, para mayor claridad en su presentación y discusión: proceso de negocio, arquitectura del sistema, casos de usos, diagrama relacional, implementación, y pruebas.

Proceso de negocio

En primer lugar se modeló el proceso de negocio. En la Figura 4, se puede apreciar el modelo realizado donde actúan los distintos usuarios. En el procedimiento para realizar la gestión de los planes de riesgos, es necesario contar con cada usuario registrado en el sistema y que esté asociado a un proyecto, ya que cada uno tiene un papel importante, ya sea seleccionando los riesgos, valorándolos, haciendo la planificación o supervisión de éstos.

Arquitectura del sistema

Para concretar un marco de referencia de la construcción del sistema, se define un conjunto de componentes y las interacciones que tienen estos mismos para llevar a cabo alguna función en específica. En la Figura 5, se puede apreciar la arquitectura del sistema, en donde inicia por parte de un usuario que realiza las peticiones a través de un navegador web. Estas peticiones son recibidas y procesadas por el *framework* mediante el patrón de diseño MVC, realizando consultas o insertando información en la base de datos MySQL. Una vez procesada las solicitudes que ocurren en el servidor, son presentadas al usuario mediante las vistas, las cuales son mostradas en el navegador web.

Casos de Usos

Se describen los principales casos de uso para el desarrollo del proyecto, representando las funciones provistas por el sistema a distintos tipos de actores.

a. Caso de uso Gestionar Riesgos. Muestra la interacción de los usuarios Analista riesgos, Integrante proyectos, Jefe proyectos y Gerente proyectos en la operación principal del sistema, que es la gestión de riesgos, donde se aplican cada una de las etapas de este proceso. El resultado de esta interacción permite a los usuarios Gerente proyectos y Jefe proyectos tener documentados los pasos a seguir para cuando se presentan los riesgos identificados.

b. Caso de uso Administrar Riesgos. Muestra la interacción de los usuarios Administrador y Analista riesgos con la administración de riesgos, donde se puede crear un nuevo riesgo, asignándole el tipo de riesgo que corresponde. Además, se puede modificar o eliminar uno existente y mostrar una lista de los riesgos que ya se encuentran ingresados en el sistema.

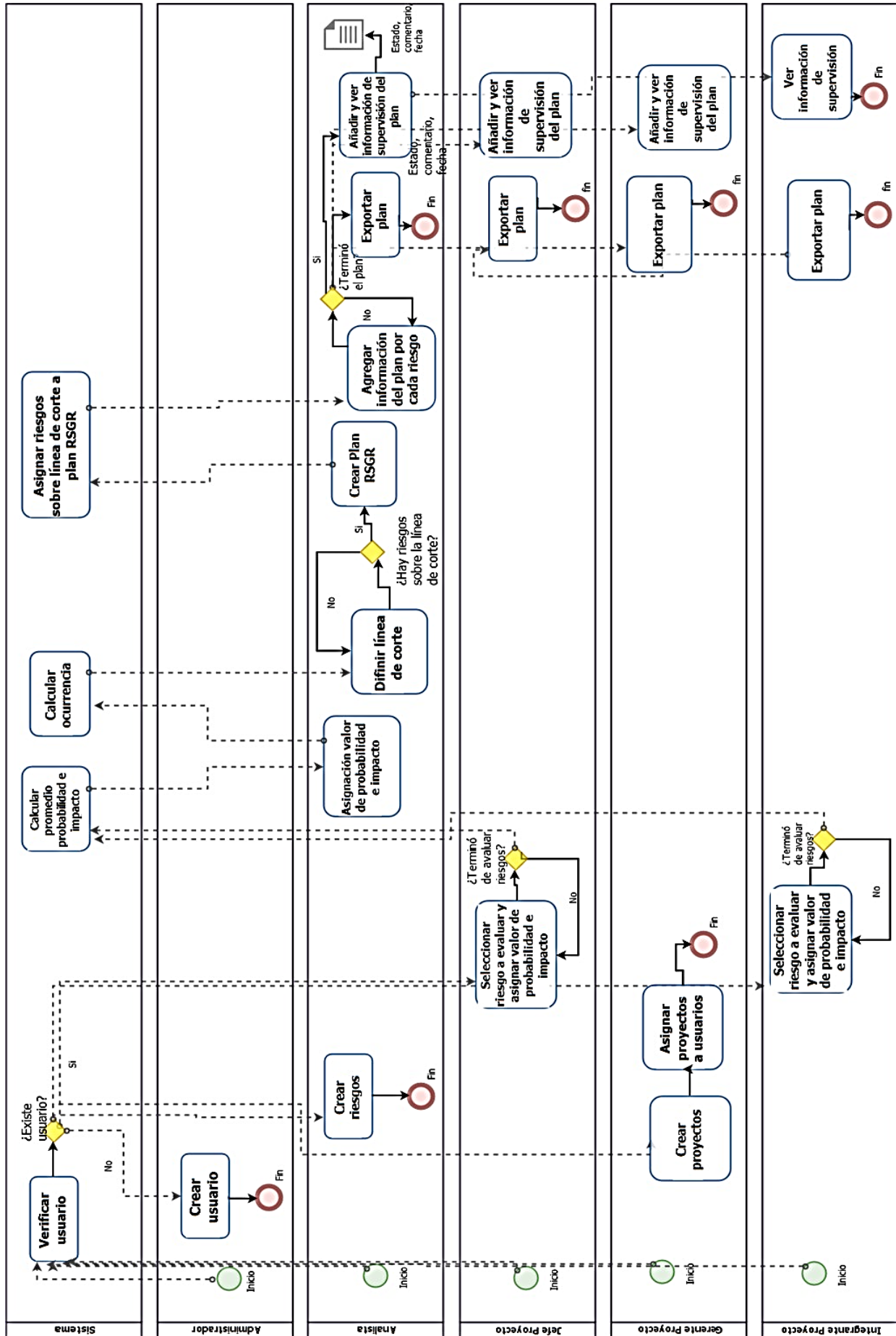


Fig. 4: Diagrama del modelo de proceso de negocio.

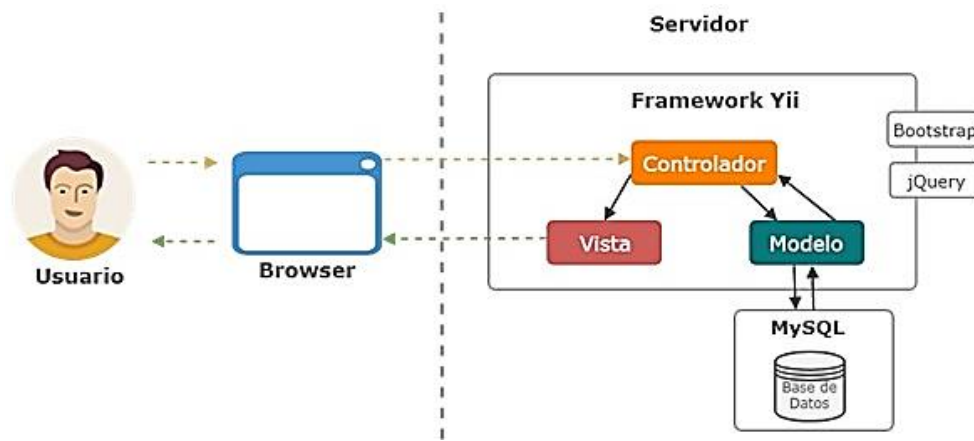


Fig. 5: Diagrama del Arquitectura del sistema

c. Caso de uso Administrar Proyectos. En este caso uso se aprecia la interacción de los usuarios Administrador y Gerente proyectos con la administración de proyectos, donde podrán crear un nuevo proyecto, modificar o eliminar uno existente y listar los proyectos que se encuentran registrados en el sistema.

d. Caso de uso Asignar Proyectos a Usuarios. Muestra la interacción de los usuarios Administrador y Gerente proyectos con la asignación de proyectos a los usuarios que están en el sistema, donde se puede crear una asignación, modificar o eliminar una existente y se puede mostrar una lista de todas las asignación de proyectos que se han realizado a los usuarios Jefe proyectos e Integrante proyectos.

e. Caso de uso Consultar Plan RSGR. En este caso uso se aprecia la interacción de todos los usuarios identificados en el sistema con la operación de consultar un plan RSGR que se encuentre registrado en el sistema hasta la fecha.

f. Caso de uso Administrar Usuarios. Este caso de uso, se muestra la interacción del Administrador con la administración de usuarios del sistema, donde podrá crear un nuevo usuario, modificar o eliminar uno existente y listar los usuarios que se encuentran registrados en el sistema

Diagrama Relacional

En la Figura 6, se puede apreciar el diagrama lógico relacional para el desarrollo del sistema en donde se muestran once entidades o tablas que fueron utilizadas para la persistencia de datos.

Implementación

Tanto el desarrollo, incluidas las pruebas, como la puesta *in-situ* (operación) fueron realizadas en las instalaciones de la empresa Kuvemar (<https://www.kuvemar.com/>). La implementación consideró cinco perfiles de usuario: administrador, gerente de proyecto, jefe de proyecto, analista (*coach*) e integrante. A continuación, se presentan las principales vistas del sistema de gestión de riesgos. Estas pertenecen a la sesión del usuario jefe de proyecto, analista de riesgos del módulo Gestionar Riesgos.

a. Vistas Jefe de Proyecto - Vista selección y evaluación de riesgos. En la Figura 7, se presenta la vista donde el jefe de proyecto puede realizar la selección de los riesgos que eventualmente podría sufrir el proyecto al cual fue asignado. En este formulario, deberá seleccionar el riesgo, ingresar un valor de probabilidad y un valor de impacto que el estime conveniente. Se aprecia el formulario correspondiente a la selección y valoración de los riesgos. Una vez hecho clic en el botón crear, el sistema desplegará el mismo formulario para que el usuario pueda seleccionar inmediatamente un nuevo riesgo. En la Figura 8, se presenta la vista donde muestra el listado de los riesgos que fueron seleccionados y valorados por el jefe de proyectos.

b. Vistas usuario Analista - Vista validación y evaluación de riesgos. En la Figura 9, se presenta la vista del usuario analista de riesgos. Al seleccionar el proyecto a evaluar, se despliega una lista con todos los riesgos evaluados y valorados por los usuarios que fueron asignados a este proyecto. Se muestra el listado de los riesgos junto a los promedios de probabilidad e impacto y, de esta forma, el analista de riesgos puede considerar la opinión de los participantes del proyecto respecto al efecto que puede producir un determinado riesgo en el desarrollo del proyecto.

En la Figura 10, se presenta la vista donde muestran los riesgos de un plan RSGR, que puede generar el analista. Se pueden apreciar los riesgos pertenecientes a un plan RSGR de un proyecto determinado, por cada riesgo se pueden realizar las operaciones que allí aparecen.

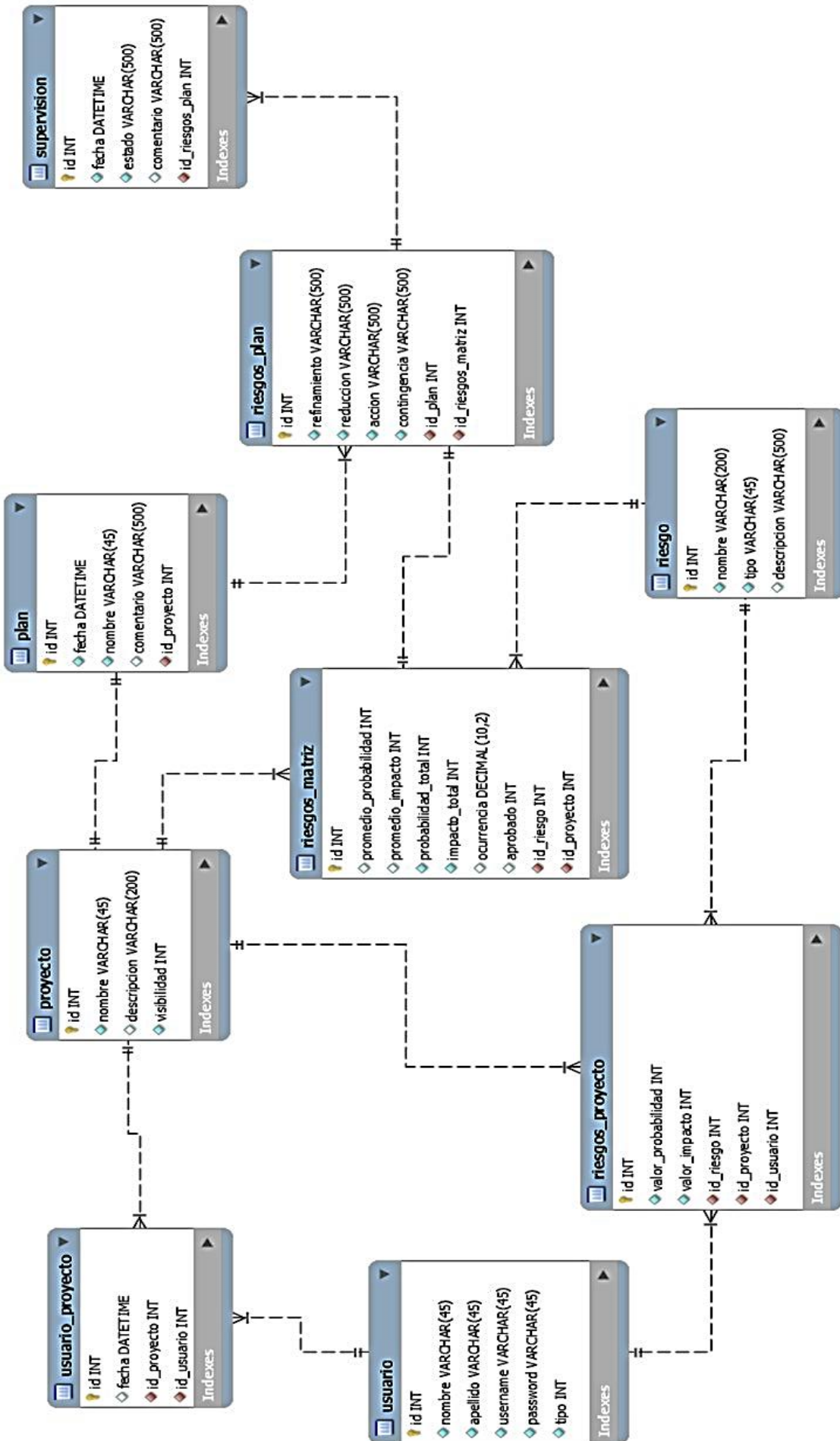


Fig. 6: Diagrama del modelo lógico relacional.

Gestión de Riesgos Inicio Usuarios Proyectos **Riesgos** Asignar Proyectos a Usuarios Planes RSGR Cerrar Sesión(admin)

Inicio / Proyectos / Tabla Riesgos

Tabla de Riesgos

Operaciones
[Lista Riesgos](#)
[Administrar Riesgos](#)

Riesgos	Promedio Probabilidad %	Promedio Impacto	Probabilidad %	Impacto
Requisitos poco claros	70	1	<input type="text"/>	<input type="text"/>
Tiempo de entrega	40	2	<input type="text"/>	<input type="text"/>
Diseño erróneo	40	1	<input type="text"/>	<input type="text"/>
Falta de un experto	30	2	<input type="text"/>	<input type="text"/>
Conflicto con los integrantes del grupo	20	3	<input type="text"/>	<input type="text"/>
Falta de seguimiento permanente de tareas y actividades	30	3	<input type="text"/>	<input type="text"/>

[Crear](#)

Fig. 7. Vista seleccionar riesgos por parte del usuario jefe de proyecto.

Riesgos Evaluados Proyecto: Coffee House

Operaciones
[Listar Proyectos](#)
[Añadir Riesgo](#)

Viendo 1-4 de 4 resultados.

Riesgo	Valor Probabilidad (%)	Valor Impacto	Usuario
Tiempo de entrega	40	2	Mario Lopez
Requisitos poco claros	70	1	Mario Lopez
Diseño erróneo	40	1	Mario Lopez
Falta de un experto	30	2	Mario Lopez

Fig. 8. Vista riesgos seleccionados por parte del usuario jefe de proyecto.

Tabla de Riesgos

Operaciones
[Lista Riesgos](#)
[Administrar Riesgos](#)

Riesgos	Promedio Probabilidad %	Promedio Impacto	Probabilidad %	Impacto
Requisitos poco claros	70	1	<input type="text"/>	<input type="text"/>
Tiempo de entrega	40	2	<input type="text"/>	<input type="text"/>
Diseño erróneo	40	1	<input type="text"/>	<input type="text"/>
Falta de un experto	30	2	<input type="text"/>	<input type="text"/>
Conflicto con los integrantes del grupo	20	3	<input type="text"/>	<input type="text"/>
Falta de seguimiento permanente de tareas y actividades	30	3	<input type="text"/>	<input type="text"/>

[Crear](#)

Fig. 9. Vista validación y evaluación de riesgos

Riesgos del Plan	Operaciones
Requisitos poco claros	Ver Editar Supervisar
Tiempo de entrega	Ver Editar Supervisar

Fig. 10. Vista riesgos de un plan para generar el RSGR

Pruebas

El proceso de prueba conlleva la realización de un conjunto de tareas a lo largo del ciclo de vida del sistema. De acuerdo con el estándar IEEE 1012-2016 (Kim et al., 2016), el conjunto mínimo de pruebas que se realizaron fueron las siguientes: a. Prueba modular que consiste en la prueba de cada módulo aislado del resto del sistema. b. Prueba de integración que se realiza a medida que los diferentes módulos del sistema se integran en el mismo. Ya se ha realizado la prueba modular y se supone que todos los módulos son correctos. El objetivo fundamental de esta prueba es comprobar que las interfaces entre los distintos módulos son correctas; c. Prueba del sistema que se realiza cuando se han integrado todos los módulos y su objetivo es comprobar que el sistema satisface los requisitos del usuario, tanto los funcionales como los no funcionales; d. Prueba de aceptación que se realiza una vez que el sistema se ha implantado en su entorno real de funcionamiento y su objetivo es demostrar al usuario que el sistema satisface sus necesidades.

Las pruebas arrojaron como resultado que las distintas tecnologías aplicadas en los módulos para el desarrollo del sistema funcionaron correctamente, logrando una eficiente conexión entre las distintas capas del modelo, vistas y controladores para tener los resultados vistos anteriormente. Para lograr esto, las funciones desarrolladas tuvieron que recibir y enviar información, como también hacer consultas a la base de datos para realizar acciones de guardar u obtener esta información y, de esta forma, mostrarlas en las vistas que correspondan. Es preciso enfatizar que la función principal del sistema se basa en la generación de la estructura de un documento de un plan de gestión de riesgos en el desarrollo de un software, en donde se tuvo que implementar funciones e interfaces que permitan crear dicho documento de forma dinámica. Por esto, se desarrollaron vistas con múltiples formularios pertenecientes a distintos modelos, los cuales tuvieron un buen resultado, logrando cumplir con las funcionalidades del sistema.

DISCUSIÓN

Con el fin de ayudar a identificar de forma anticipada los problemas que puedan acontecer en el desarrollo de los proyectos de software, surgió la idea de desarrollar una herramienta software que ayude en la gestión de planes de riesgos en proyectos informáticos. A raíz de que el proceso de gestión de riesgos es inmaduro, se estudiaron diferentes modelos para la gestión. De esta forma, se pudo llegar a un proceso de negocio fácil de comprender y posteriormente llevar a concretar en la herramienta propuesta.

Para el desarrollo de este sistema, se utilizaron tecnologías modernas, lo que hace que el sistema propuesto se destaque entre los estudiados, ya que no todas las herramientas existentes son para la web y otras no están disponibles para ser utilizadas. Por otro lado, hay herramientas que cubren todo el proceso de gestión del riesgo, pero éstas se desarrollan de una forma muy acotada, siendo participe de los procesos sólo un miembro del equipo y no permitiendo una opinión del resto de los integrantes del proyecto y menos aún la de un analista experto (*coach*). Dados los resultados obtenidos en las pruebas realizadas al sistema, se puede concluir que la herramienta cumple con el propósito establecido, generando los planes de riesgos. La implementación de la herramienta fue diseñada para que cada usuario realice la gestión de riesgos de manera independiente, donde cada participante sólo puede revisar su propia evaluación de los riesgos. Sería interesante poder realizar un sistema, donde cada cambio que hagan los usuarios se vean reflejados en tiempo real y pueda realizarse retroalimentación de la misma manera.

CONCLUSIONES

A partir de los resultados obtenidos, se pueden extraer las siguientes conclusiones:

- 1) El sistema desarrollado es una propuesta tecnológica que busca disminuir la brecha existente entre los proyectos de desarrollo de software exitosos y los no exitosos que no tratan con la gestión de riesgos. Los servicios que se ofrecen y, en particular, el servicio para el Jefe proyecto, apuntan en este sentido. Todos los servicios en las pruebas de aceptación, demostraron ser eficaces en la misma empresa Kuvemar.
- 2) El sistema tiene gran potencial para ser utilizado por alumnos y profesores de la Universidad de Tarapacá, en sus carreras asociadas al desarrollo de algún tipo de software. Se cuenta con el apoyo desinteresado de la empresa Kuvemar, para dar soporte.
- 3) La herramienta desarrollada puede generar nuevas oportunidades de negocios, especialmente para emprendedores en el desarrollo de software, ya que les permitirá gestionar los riesgos asociados con su experiencia inicial valorándolo y definiendo acciones que puedan controlar o atenuar las consecuencias.
- 4) El sistema fue desarrollado aplicando tecnologías modernas y una metodología evolutiva incremental, representada mediante casos de uso que incluye las etapas de especificación de requisitos, análisis y diseño, implementación y pruebas del software a través del estándar IEEE 1012-2016. Esto permitirá mantener actualizado el sistema o hacerle mejoras a través del tiempo. Por ejemplo, se podría implementar un módulo que permita la comunicación a través de mensajes directos, con la finalidad de establecer una relación más cercana entre desarrolladores y analistas de riesgos. También, se podría implementar un módulo que permita a los desarrolladores trabajar en tiempo real.

AGRADECIMIENTOS

Este trabajo cuenta con el financiamiento del programa de Investigación de la Universidad de Tarapacá, Arica, Chile (2017-2018), en el contexto de los proyectos "Desarrollo de una propuesta formativa en habilidades blandas específicas para emprendimiento en TICs enfocada a alumnos de Ingeniería Informática en Chile" (código 8742-17) y "Desarrollo de un modelo para emprendimiento basado en la construcción de productos de interacción humano computador" (código 8727-18). Asimismo, a la empresa Kuvemar por su valiosa colaboración, interés y disposición de los equipos computacionales y personal humano, para el desarrollo, prueba y puesta en servicio del sistema construido.

REFERENCIAS

- Alarcón-Aldana, A., Urrutia-Pinilla, J., y Callejas-Cuervo, M., *Aplicación Móvil para la Administración de Variables Físicas en Ciclismo al Aire Libre*, Información tecnológica, 27(4), 175-182 (2016).
- Barafort, B., Mesquida, A. L., y Mas, A., *Integrating risk management in IT settings from ISO standards and management systems perspectives*, Computer Standards & Interfaces, 54, 176-185 (2017).
- Bertone, R. A., Thomas, P. J., Taquias, D., y Pardo, S., *Herramienta para la Gestión de Riesgos en proyectos de software*, Proceedings del XVI Congreso Argentino de Ciencias de la Computación, 567-576 (2010).
- Cabarcas, A., Puello, P., y Martelo, R. J., *Sistema de Información Soportado en Recuperación XML para Pequeñas y Medianas Empresas (PYME) de Cartagena de Indias, Colombia*, Información tecnológica, 26(2), 135-144 (2015). X
- Castillo, A., *Curso de Programación Web: JavaScript, Ajax y jQuery*, IT Campus Academy, España (2017).
- Castillo, E., *Proyecto de elaboración de la metodología de gestión de riesgos en proyectos de desarrollo de software para la empresa consultora CV3*, Disertación Doctoral, Universidad para la Cooperación Internacional, San José, Costa Rica (2009).
- Coca, G. A., Castrillón, O. D., y Ruiz-Herrera, S., *Programación de un Sistema de Fabricación tipo "Job Shop" bajo un Enfoque de Sostenibilidad*, Información tecnológica, 27(6), 31-52 (2016).
- Ghane, K., *Quantitative planning and risk management of Agile Software Development*, 2017 IEEE Technology & Engineering Management Conference (TEMSCON), 109-112 (2017).
- Gómez, J. M. P., *UF2175-Diseño de bases de datos relacionales*. Ediciones Paraninfo, Madrid, España (2014).
- Gómez, U. E., Pérez, J. P., y Ramírez, J. L., *Sistema de Información Agrícola para la disminución de Brechas entre Oferta y Demanda-AGROCRAFT*, Información tecnológica, 27(3), 215-220 (2016).
- González, Y. D., y Romero, Y. F., *Patrón Modelo-Vista-Controlador*, Revista Telemática, 11(1), 47-57 (2012).
- Hastie, S., y Wojewoda, S., *Standish group 2015 chaos report-q&a with jennifer lynch*, Retrieved, 1(15) (2016).
- Haughee, E. *Instant Sublime Text Starter*, Packt Publishing Ltd, Birmingham, Reino Unido (2013).

- Hussain, S., Keung, J., y Khan, A. *Software design patterns classification and selection using text categorization approach*, Applied soft computing, 58, 225-244 (2017).
- Johnson, J., *CHAOS Report 2014*. The Standish Group (2015).
- Johnson, H. A. *Trello*, Journal of the Medical Library Association: JMLA, 105(2), 209 (2017)
- Kim, J. K., Woo, J. B., Kim, S. J., Kim, H. W., Ro, Y. S., Choi, J. J., y Park, J. H. *Application Study for Implementing IEEE 1012-2016*. Transactions of the Korean Nuclear Society Spring Meeting, Jeju, Korea (2017).
- Krahenbuhl, J. H., *Learning Axure RP interactive prototypes*, Packt Publishing Ltd, Birmingham, Reino Unido (2015).
- Kuna, H. D., Caballero, S., Jaroszczuk, S. E., y Miranda, M. J., *Plan de Riesgos para la implementación, desarrollo y mantenimiento de componentes de Web 2.0 en Bibliotecas, caso de estudio en una Biblioteca Especializada*, Proceedings de VI Jornada sobre la Biblioteca Digital Universitaria, 1-17 (2008).
- Laaziri, M., Benmoussa, K., Khouliji, S., Larbi, K. M., y El Yamami, A. *Analyzing bootstrap and foundation front-end frameworks: a comparative study*, International Journal of Electrical and Computer Engineering (IJECE), 9(1), 713-722 (2019).
- Lancor, L., y Katha, S., *Analyzing PHP frameworks for use in a project-based software engineering course*, Proceedings of the 44th ACM technical symposium on Computer science education, 519-524 (2013).
- Mamani-Alave, M., Villalobos-Abarca, M. y Herrera-Acuña, R., *Sistema web de bajo costo para monitorear y controlar un invernadero agrícola*, Ingeniare, 25(4), 599-618 (2017).
- Mosquera, L., Andrade, D. y Sierra, L., *Guía para apoyar la priorización de riesgos en la gestión de proyectos de tecnologías de la información*, Revista GTI, 12(33), 15-32 (2013).
- Naz, R., Khan, M. N. A. y Aamir, M., *Scrum-based methodology for product maintenance and support*, International Journal of Engineering and Manufacturing (IJEM), 6(1), 10 (2016).
- Olanrewaju, R. F., Islam, T., y Ali, N. A., *An empirical study of the evolution of PHP MVC framework*, Advanced Computer and Communication Engineering Technology, 399-410 (2015).
- Pasha, M., Qaiser, G., y Pasha, U., *A critical analysis of software risk management techniques in large scale systems*, IEEE Access, 6, 12412-12424 (2018).
- Pressman, R., *Ingeniería del Software: Un enfoque práctico*, 5a. Edición, McGraw-Hill, Madrid, España (2002).
- Quintero, W., Robles, C. A., y Vilorio, A. M., *Sistema de Información para Detección de Crecientes Súbitas en la Cuenca del Río Manzanares en Santa Marta, Colombia*, Información tecnológica, 28(6), 95-102 (2017).
- Sarigiannidis, L., y Chatzoglou, P. D., *Software development project risk management: A new conceptual framework*, Journal of Software Engineering and Applications, 4(5), 293-305 (2011).
- Sommerville, I., *Ingeniería del software*, Pearson educación, Londres, Reino Unido (2005).
- Vidal, C., López, L., Rojas, J., y Castro, M., *Desarrollo de Sistema Web de Reclutamiento y Selección y de Directivos por Competencias mediante PHP CodeIgniter 3.0*. Información tecnológica, 28(2), 203-212 (2017).
- Wong, S. Y., Chin, K. S., y Tang, D., *Strengthening risk evaluation in existing risk diagnosis method*, Industrial Engineering and Management Systems, 9(1), 41-53 (2010).

Página en blanco