PROTOTIPO FUNCIONAL DE APLICACIÓN WEB PARA LA DETECCIÓN TEMPRANA DE LOS RIESGOS EN LAS ETAPAS DE PRE ANÁLISIS Y ANÁLISIS EN LOS PROYECTOS DE SOFTWARE.

DIANA MARCELA JAIMES JAIMES
MARÍA CAMILA RINCÓN AYALA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
GRUPO DE INVESTIGACIÓN GTI TECNOLOGÍAS DE INFORMACIÓN
LÍNEA DE INVESTIGACIÓN SISTEMA DE INFORMACIÓN EN INGENIERÍA DE SOFTWARE
BUCARAMANGA
2020

PROTOTIPO FUNCIONAL DE APLICACIÓN WEB PARA LA DETECCIÓN TEMPRANA DE LOS RIESGOS EN LAS ETAPAS DE PRE ANÁLISIS Y ANÁLISIS EN LOS PROYECTOS DE SOFTWARE.

DIANA MARCELA JAIMES JAIMES
MARÍA CAMILA RINCÓN AYALA

Proyecto de Trabajo de Grado de Ingeniería de Sistemas

Director
Angélica María Zapata García

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
GRUPO DE INVESTIGACIÓN GTI TECNOLOGÍAS DE INFORMACIÓN
LÍNEA DE INVESTIGACIÓN SISTEMA DE INFORMACIÓN EN INGENIERÍA DE SOFTWARE
BUCARAMANGA
2020

Tabla de contenido

INTRODUCCIÓN 8
1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN 9
2. PREGUNTA DE INVESTIGACIÓN 13
2.1 HIPÓTESIS 13
2.1.1 HIPÓTESIS PRINCIPAL 13
2.1.2 HIPÓTESIS ALTERNATIVAS 13
3. OBJETIVO GENERAL 14
4. OBJETIVOS ESPECÍFICOS 15
4.1 OBJETIVO ESPECÍFICO 1 15
4.2 OBJETIVO ESPECÍFICO 2 15
4.3 OBJETIVO ESPECÍFICO 3 15
4.4 OBJETIVO ESPECÍFICO 4 15
5. MARCO REFERENCIAL 16
5.1 MARCO CONCEPTUAL 16
5.1.1 Gestión de riesgos: 16
5.1.2 Ingeniería de requerimientos: 16
5.1.3 Metodologías de gestión de riesgo: 16
5.1.4 Herramientas de administración de riesgo 17
5.1.5 Pre análisis 17
5.1.6 Análisis 17
5.2 MARCO TEÓRICO 18
5.2.1 Gestión de riesgos: 18
5.2.2 Ingeniería de Requerimientos: 22
5.2.3 Metodologías: 23
5.2.4 Herramientas: 25
5.2.5 Etapas de pre análisis y análisis 26
5.3 ESTADO DEL ARTE 27
5.4 MARCO LEGAL Y POLÍTICO 34
6. DISEÑO METODOLÓGICO 36
6.1 FASES Y ACTIVIDADES 36

6.2 MATERIALES, TÉCNICAS E INSTRUMENTOS 38
6.2.1 Técnica: Análisis de documentos 38
6.2.2 Técnica: Pruebas Unitarias al Prototipo de Aplicación 38
6.2.3 Técnica: Encuesta 38
6.3 POBLACIÓN Y MUESTRA 39
7. CRONOGRAMA DE ACTIVIDADES 40
8. PRESUPUESTO 41
9. DESARROLLO DE LA TESIS 42
9.1 Cálculo de los Riesgos 47
10. PROTOTIPO DE DETECCIÓN DE RIESGOS 49
11. RESULTADOS OBTENIDOS 59
12. ANÁLISIS DE LOS RESULTADOS OBTENIDOS 61
12.1 Análisis resultados dados por el prototipo al usuario 61
12.2 Encuesta de percepción del cliente sobre el prototipo 63
12.2.1 Perfil de los proyectos seleccionados en la muestra 67
13. CONCLUSIONES 68
14. RECOMENDACIONES 70
15. TRABAJOS FUTUROS 71
16. ANEXOS 72
17. REFERENCIAS 95

LISTA DE TABLAS

Tabla 1. Riesgos de costo en proyectos de software.....20
Tabla 2. Riesgos de requerimientos en proyectos de software 20
Tabla 3. Riesgos de desarrollo en proyectos de software 21
Tabla 4. Riesgos de calidad en proyectos de software.....21
Tabla 5. Riesgos empresariales en proyectos de software.....22
Tabla 6. Herramientas para la gestión de riesgo en proyectos de software 25
Tabla 7. Objetivos de las etapas de Pre análisis y análisis.....27
Tabla 8. Análisis del Estado del arte.....28
Tabla 9. Normas para la gestión de riesgo en los proyectos de software.....34
Tabla 10. Clasificación de las empresas.....35
Tabla 11. Cronograma.....40
Tabla 12. Presupuesto.....41
Tabla 13. Tabla Comparativa de las metodologías más utilizadas para la gestión de riesgos 73
Tabla 14. Características de las herramientas tecnológicas para la gestión de riesgo.....82
Tabla 15. Tabla comparativa entre las herramientas tecnológicas y el prototipo funcional 83

LISTA DE FIGURAS

Figura 1. Árbol de problemas 11
Figura 2. Cuadro comparativo de la gestión de riesgo entre los años 90 y actualmente 19
Figura 3. Proceso de ingeniería de requerimientos 23
Figura 4. Funcionamiento de Practical Threat Analysis (PTA) 25
Figura 5. Gráfica circular con los documentos del estado del arte ordenados por año 28
Figura 6. Línea de tiempo de los documentos seleccionados 29
Figura 7. Procedimiento llevado a cabo para el desarrollo del trabajo de grado ... 42
Figura 8. Diagrama de casos de uso del prototipo funcional web 44
Figura 9. Diagrama de secuencia de Crear cuenta 45
Figura 10. Diagrama de secuencia de Iniciar Sesión 46
Figura 11. Diagrama de secuencia de Crear empresa 46
Figura 12. Diagrama de secuencia de Crear proyectos 47
Figura 13. Funcionamiento interno del prototipo de detección de riesgos 49
Figura 14. Diagrama de actividades del prototipo web 50
Figura 15. Logo PHP 51
Figura 16. Logo PostgreSQL 52
Figura 17. Logo Heroku 53
Figura 18. Pantalla Home 54
Figura 19. Pantalla Inicio Sesión 54
Figura 20. Pantalla Registro 55
Figura 21.Pantalla Agregar empresa parte 1 55
Figura 22. Pantalla Agregar empresa parte 2 56
Figura 23. Pantalla de la página principal 56
Figura 24. Pantalla Crear Proyecto 57
Figura 25. Pantalla Ver Proyectos 57
Figura 26. Pantalla Ver Riesgos del Proyecto 58
Figura 27. Frecuencia de los riesgos que se presentaron en los proyectos 62
Figura 28. Gráfica circular de la pregunta número 1 de la encuesta 64
Figura 29. Gráfica de barras de la pregunta número 2 de la encuesta 64
Figura 30. Gráfica de barras de la pregunta número 3 de la encuesta 65
Figura 31. Gráfica de barras de la pregunta número 4 de la encuesta 65

TABLA DE ANEXOS

Anexo 1. ESTUDIO COMPARATIVO DE LAS METODOLOGÍAS MÁS UTILIZADAS PARA LA GESTIÓN DE RIESGO 72
Anexo 2. HERRAMIENTAS TECNOLÓGICAS PARA LA GESTIÓN DE RIESGO 81
Anexo 3. INFORMACIÓN QUE CONTENDRÁ EL PROTOTIPO PARA DETECTAR LOS RIESGOS 86

INTRODUCCIÓN

En el desarrollo de los proyectos de software, es de vital importancia tener en cuenta los diferentes riesgos que se presentan, desde el instante en que surge la idea de construir un nuevo software y en las diferentes etapas de la metodología elegida para el desarrollo del proyecto. Así que, es fundamental llevar a cabo una apropiada gestión de riesgo en todas las etapas del desarrollo, principalmente en las etapas de pre análisis y análisis, para así elevar las posibilidades de obtener un resultado exitoso del proyecto y/o evitar el fracaso de este. Entendiendo la gestión de riesgos como las prácticas y principios destinadas a identificar, analizar y manejar los diferentes riesgos que se presentan en los proyectos de software (Bannerman, 2008).

Al realizar un estudio minucioso de revisión con base en información de diversas fuentes, se encontraron los riesgos reincidentes en los proyectos de software; con los cuales se busca generar conciencia en los gerentes de los proyectos frente al riesgo, y así mismo, ayudar en la reducción del número de proyectos que no son entregados o no cumplen a cabalidad con los requerimientos del cliente. Esto, se logra mediante la identificación temprana de los riesgos en las primeras etapas para así, contar con un mejor seguimiento de estos en lo que resta del desarrollo del proyecto; basándonos en criterios como las características del cliente, equipo de trabajo y los requerimientos funcionales de cada proyecto de software, y de este modo, mostrar los riesgos que se podrían presentar en el desarrollo del software junto con su respectiva gestión o tratamiento de mitigación.

Palabras Claves: *Gestión de riesgos, Ingeniería de Requerimientos, Metodologías de gestión de riesgo y Herramientas de gestión de riesgo.*

1. PLANTEAMIENTO DEL PROBLEMA Y JUSTIFICACIÓN

La industria del software ha tenido un crecimiento notorio en los últimos años, lo cual, se puede apreciar en los diferentes sectores de la sociedad, como la salud, la educación, seguridad, entre otros. En la educación se ha innovado la manera de dar las calificaciones, y de enseñar a los alumnos, además, el número de computadores ha aumentado en estas instituciones; en los centros médicos la tecnología es más notoria para la detección y seguimiento de enfermedades; en la gran mayoría de hogares del mundo hoy día se cuenta con uno o más dispositivos inteligentes. Todo lo dicho anteriormente, demuestra que la gran mayoría de las personas ha experimentado el uso de software, pero muy pocas conocen el trasfondo que conlleva la planeación, desarrollo e implementación de este.

En la creación del software, la mayoría de los implicados, como lo son el gerente del proyecto, los desarrolladores, los diseñadores, los responsables de pruebas y de calidad e incluso el cliente, se interesan principalmente en que el producto final sea terminado lo más rápido posible y no prestan atención a realizar una planeación más a fondo, en la cual se omiten factores importantes para que el producto de software final sea exitoso, como lo es la gestión de riesgos.

Por tal razón, una de las grandes problemáticas a tratar durante el ciclo de vida del software, es que, tanto a nivel mundial como en Colombia no se tiene una adecuada gestión del riesgo, según el tipo de proyecto que se está desarrollando. Esto, conlleva a efectos como, la entrega de proyectos fuera de tiempo, presupuesto y requerimientos establecidos o en la peor situación, proyectos completamente fracasados.

Esto se ve reflejado a nivel mundial, según el reporte Chaos del 2018, donde solo el 28% de los proyectos de software, fueron entregados satisfactoriamente, es decir, a tiempo y dentro del presupuesto, además el 52% de estos, fueron entregados fuera de lo previamente establecido y el 20% fallaron totalmente, es decir, no fueron entregados (Report, 2018).

Asimismo, en Latinoamérica mediante un informe realizado por el Project Management Institute (PMI), se muestra que, el 71% del éxito en el cumplimiento de los objetivos estipulados, se da, gracias a la alta prioridad en la gestión de proyectos. Así mismo, se logra obtener un producto final dentro del presupuesto, a tiempo y con el cumplimiento de los objetivos previstos, esto, con un 64%, 61% y 71%, respectivamente (Profession, 2016).

Todas estas estadísticas son causadas, principalmente, por no contar con una cultura con bases sólidas en la gestión de riesgo, esto se evidencia en una encuesta realizada a 53 empresas en España, donde el 42% de ellas consideró innecesaria esta práctica por motivos de carga en trabajo y tiempo y un 50% afirmó que no se cuenta con una formación en buscar problemas de manera anticipada; esto nos deja con un mínimo porcentaje de empresas que sí llevan a cabo la gestión de riesgos (Sanz & Silva, 2014).

En Colombia, según un estudio realizado por la Federación Colombiana de la Industria de Software y TI (Fedesoft), en el 2014 se contaba con 4016 empresas activas de las cuales el 80% se ubican en la región centro (Bogotá, Medellín, Manizales y Pereira) y tan solo el 4% en la región Oriente (Santander y Norte de Santander). En la cual, el 57% son medianas empresas, el 27% son nuevas y tan solo el 14% son de gran tamaño (Fedesoft, MinTic, & Sena, 2015). Sin embargo, Colombia no cuenta con una documentación acerca de los proyectos de software realizados, es por tal motivo, que no se tiene certeza si existe o no una adecuada gestión de riesgos. Esta situación es respaldada por la falta de normatividad, ya que, hay diferentes normas para la gestión de riesgos en la Seguridad y Salud Ocupacional [NTC OHSAS 18001], la Inocuidad Alimentaria [ISO 22000] y en otras áreas (ISOTools, 2019). Pero, por el contrario, para los proyectos de software se cuenta con pocas normas, entre estas se encuentran: IEEE 1540-2001's Risk Management Standard (IEEE, 2001), el estándar ISO/IEC 16085:2006 (ISO, 2006) y la ISO 31000 (ISOTools, 2019), esta última, es en función de cualquier tipo de organización, por lo que también es utilizada en los proyectos de software.

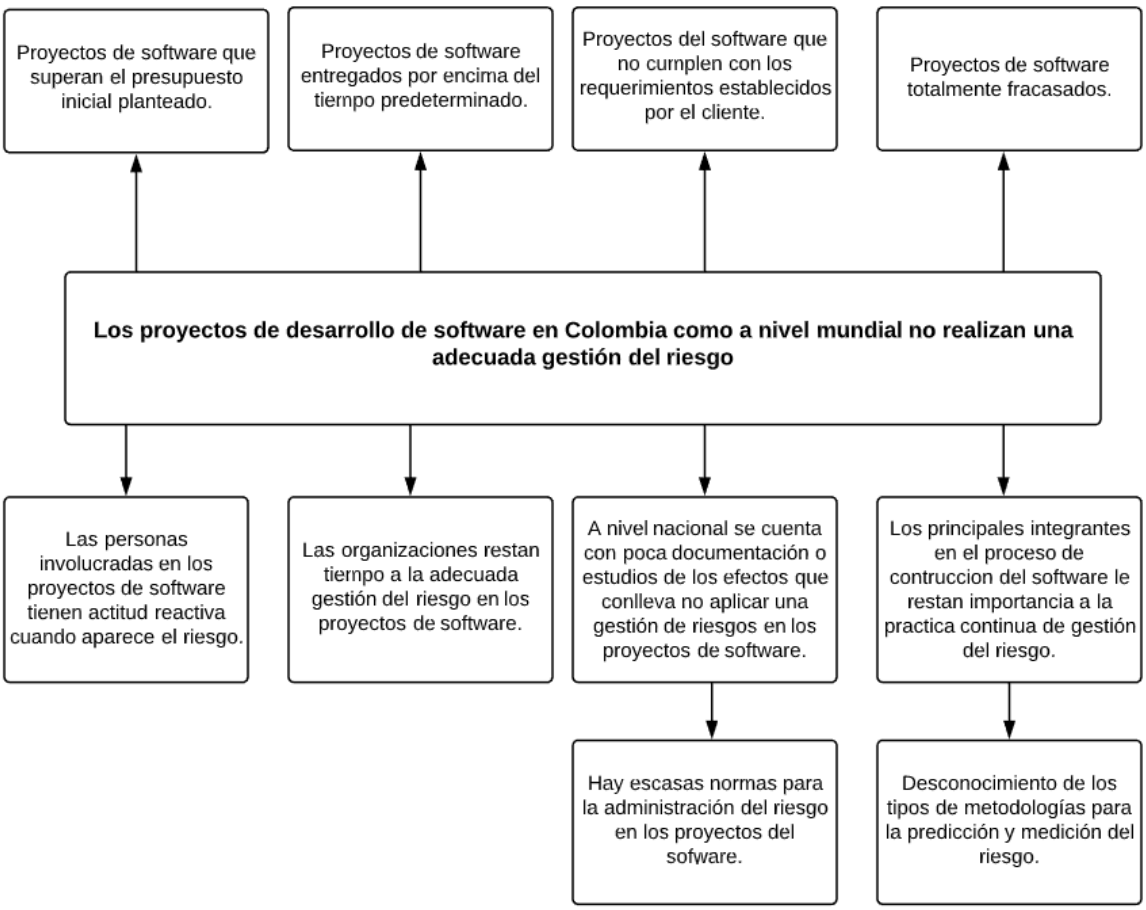


Figura 1. Árbol de problemas

En resumen, el principal problema a abordar es la falta de conciencia por parte de los directores de proyectos que omiten realizar una adecuada gestión del riesgo, lo cual se presenta en el ámbito nacional e internacional y se evidencia en la investigación realizada. En Colombia radica en las pocas normas, estándares o medidas que guían a hacer una adecuada gestión de riesgos a lo largo del proyecto, siendo una de las consecuencias, que las organizaciones y el personal involucrado en el desarrollo de proyectos de software, tengan poco conocimiento de las metodologías para abordar esta problemática, e igualmente el hecho de que le restan importancia a esta práctica. Es por tal razón, que se llevará a cabo esta tesis, con el fin de proponer un prototipo que permita la detección de los riesgos durante las etapas de pre análisis y análisis, mediante las características propias de cada

proyecto, como las del cliente, del equipo de trabajo y de los requerimientos funcionales.

2. PREGUNTA DE INVESTIGACIÓN

¿Qué mecanismos se podrían plantear para realizar una detección temprana de los riesgos que se presentan durante el proceso de construcción de un producto de software en las etapas de pre análisis y análisis, según el contexto empresarial en donde éste se está llevando a cabo, para así, aumentar la probabilidad de éxito del proyecto?

2.1 HIPÓTESIS

2.1.1 HIPÓTESIS PRINCIPAL

Un prototipo de aplicación web puede ayudar a la detección temprana de los riesgos, según las características propias de cada proyecto software (tiempo para entrega del proyecto, costos totales, tipo de empresa, requerimientos del producto a desarrollar, personas involucradas en la creación del software y el proceso de desarrollo que se lleve).

2.1.2 HIPÓTESIS ALTERNATIVAS

- Se dificulta determinar los procedimientos y pautas específicas para la gestión del riesgo en proyectos de software.
- Al aplicar las pautas apropiadas de gestión de riesgos en los proyectos de software se podrá tener un mejor desarrollo en las diferentes etapas del proyecto.

3. OBJETIVO GENERAL

Implementar un prototipo funcional de aplicación web que permita la identificación de los riesgos en las etapas de pre análisis y análisis de un proyecto de software, teniendo en cuenta criterios como las características del cliente, del equipo de trabajo y de los requerimientos funcionales, para que de esta forma se detecten los riesgos de manera temprana.

4. OBJETIVOS ESPECÍFICOS

4.1 OBJETIVO ESPECÍFICO 1

Analizar las metodologías de gestión de riesgos más utilizadas en los proyectos de software como pautas para la elaboración del prototipo funcional de aplicación web.

4.2 OBJETIVO ESPECÍFICO 2

Revisar las herramientas tecnológicas actualmente utilizadas como apoyo a la gestión de riesgos en los proyectos de software para la identificación de factores diferenciales en el prototipo funcional de aplicación web.

4.3 OBJETIVO ESPECÍFICO 3

Implementar un prototipo funcional de aplicación web para la identificación temprana de los riesgos en los proyectos de software.

4.4 OBJETIVO ESPECÍFICO 4

Validar la aplicabilidad del prototipo funcional de aplicación web orientado a la detección temprana de los riesgos en un proyecto de software.

5. MARCO REFERENCIAL

En esta sección son presentados el marco conceptual, el marco teórico, el estado del arte y el marco legal y político del proyecto de investigación.

5.1 MARCO CONCEPTUAL

Los conceptos teóricos utilizados como bases fundamentales en este proyecto son: *Gestión de riesgos, Ingeniería de Requerimientos, Metodologías de gestión de riesgo, Herramientas de administración de riesgo, Pre análisis y Análisis.*

5.1.1 Gestión de riesgos:

La gestión de riesgos, se puede definir como un conjunto de principios y prácticas destinadas a identificar, analizar y manejar los factores de riesgo. Esta gestión, se inicia identificando los riesgos, luego se estima la exposición al riesgo para cada factor, utilizando esta fórmula:

$$R = P\ x\ I$$

donde R es la exposición al riesgo atribuible a un factor de riesgo particular, P es la probabilidad de que se produzca el evento no deseado e I es el impacto o la magnitud de la pérdida si ocurre el evento. Después de conocer este factor se da prioridad a los que presentan una mayor amenaza. Y, finalmente, los factores de riesgo se controlan progresivamente y se mantiene un estado de estos riegos actualizando periódicamente (Bannerman, 2008).

5.1.2 Ingeniería de requerimientos:

Se encarga de brindar especificaciones de los requerimientos de manera correcta, que describen con detalle y en forma consistente las necesidades del cliente, esto realizando su respectivo análisis y verificación (Arias Chaves, 2005).

5.1.3 Metodologías de gestión de riesgo:

Es una compilación de procedimientos, técnicas, herramientas y documentos auxiliares que sirven de apoyo a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Estas están conformadas por fases, las cuales se dividen en unas subfases, que sirven de guía para los desarrolladores de sistemas a escoger las técnicas más apropiadas en cada instante del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo (Brizuela & Blanco, 2013).

5.1.4 Herramientas de administración de riesgo

Es un conjunto de herramientas tecnológicas que brinda un soporte automatizado o semiautomatizado para el proceso y los métodos (el análisis de requerimientos, modelación del diseño, construcción del programa, pruebas y apoyo) en los proyectos de ingeniería de software (Pressman, 2010).

5.1.5 Pre análisis

Antes de que empiece un proyecto de desarrollo, es indispensable realizar una serie de tareas previas, que ayudarán en la finalización con éxito de un proyecto. Las tareas iniciales que se realizan en esta fase inicial del proyecto incluyen actividades tales como, la determinación del ámbito del proyecto, la realización de un estudio de viabilidad, el análisis de los riesgos asociados al proyecto, una estimación del coste del proyecto, su planificación temporal y la asignación de recursos a las distintas etapas del proyecto (Berzal, 2006).

5.1.6 Análisis

Esta etapa del ciclo de vida del software corresponde al proceso mediante el cual, se intenta encontrar, qué es lo que verdaderamente se necesita, para así, llegar a una comprensión adecuada de los requerimientos del sistema (Berzal, 2006).

5.2 MARCO TEÓRICO

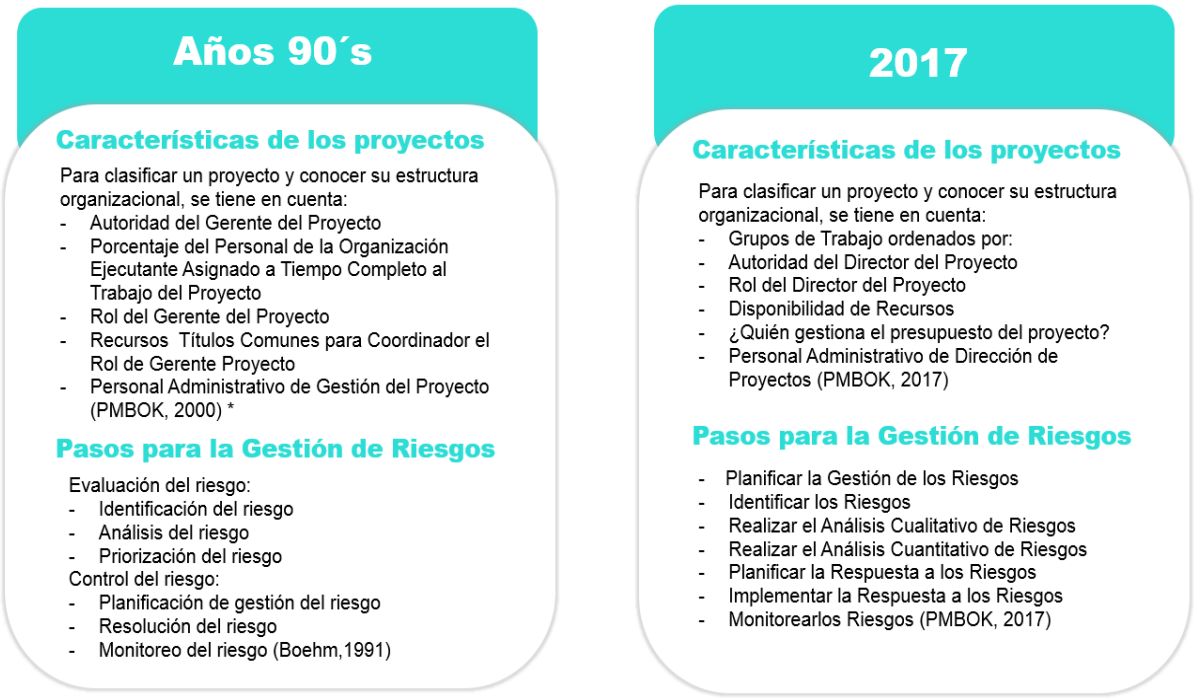
5.2.1 Gestión de riesgos:

El diccionario Webster define "*riesgo"* como "*La posibilidad de pérdida o lesión"*. En la gestión de riesgos se puede observar dos variables, las cuales son: la probabilidad de un resultado insatisfactorio y la pérdida para las partes afectadas si el resultado es insatisfactorio (Boehm,1991).

En los proyectos de software se ven implicadas varios integrantes, como, el cliente, el usuario y el desarrollador, cada uno cuenta con opiniones de satisfacción las cuales son diferentes pero muy importantes, así que está claro que un "resultado insatisfactorio" se puede evidenciar de distintas maneras (Boehm,1991).

La gestión de riesgos a lo largo de los años ha contado con cambios, los cuales son necesarios debido a la variación en el contexto empresarial, el contexto social y las características del proyecto mismo. Por lo tanto, en la Figura 2 se muestra la diferencia que ha tenido la gestión de riesgo a lo largo de este tiempo.

Figura 2. Cuadro comparativo de la gestión de riesgo entre los años 90 y actualmente



La gestión de riesgos es una parte fundamental en la creación de software. Los autores de la World Academy of Science Engineering and Technology o Waset (Hoodat & Rashidi, 2009), mencionan los diferentes riesgos que se pueden presentar en un proyecto de software, los cuales se muestran clasificados en las Tablas (1,2,3,4,5).

Tabla 1. Riesgos de costo en proyectos de software

Tipo de Riesgo	Riesgo	Descripción del Riesgo	Etapa del Ciclo de Vida
Riesgos de costo de software	Falta de acuerdo entre el cliente y el desarrollador	Cuando los requerimientos no son claros desde un inicio y además varían mucho durante el ciclo de vida del software. Lo cual provoca demoras en la entrega	Pre análisis
	Desacuerdo entre miembros	Esto se da, cuando la comunicación entre el personal del área de desarrollo no es la óptima y así mismo su sinergia no es la más eficaz para el cumplimiento de objetivos en común.	
	Escasez de personal	Este riesgo, es dado cuando la estimación del personal es incorrecta, para completar el proyecto en el tiempo definido o si en el ciclo de vida del proyecto existe mucha rotación del personal.	
	Falta de suficientes habilidades técnicas en el equipo de trabajo	Cuando no se capacita al personal de una manera adecuada o, por el contrario, se eligió un personal que no contaba con las competencias requeridas para el proyecto.	
	Conocimiento inadecuado de las herramientas requeridas	Cuando no existe una capacitación a los desarrolladores acerca de las herramientas a utilizar.	

Tabla 2. Riesgos de requerimientos en proyectos de software

Tipo de Riesgo	Riesgo	Descripción del Riesgo	Etapa del Ciclo de Vida
Riesgos de requerimientos de software	Ambigüedad de requerimientos	Esto se da principalmente, cuando el cliente no tiene claro lo que quiere.	Análisis
	Falta de informe de requerimientos	Este riesgo es dado, por el gerente de proyectos, cuando éste no lleva un registro de los requisitos que quiere el cliente. Lo cual es algo negativo en el momento de desarrollo del software.	
	Requerimientos imposibles	Este riesgo, es dado cuando el cliente tiene requerimientos con muchas funcionalidades o tecnología inviable y ser entregado en poco tiempo.	
	Requerimientos inválidos	Este riesgo, es dado cuando el cliente presenta requisitos no acordes con el producto que se está llevando a cabo, por el producto no cumplirá las funciones necesarias	

Tabla 3. Riesgos de desarrollo en proyectos de software

Tipo de Riesgo	Riesgo	Descripción del Riesgo	Etapa del Ciclo de Vida
Riesgos de desarrollo de software	Gran tamaño de arquitectura	Esto se genera cuando se cuenta con un gran número de tecnologías de la información mal relacionadas, lo cual ocasiona retrasos en la entrega y un mayor costo.	Diseño
	Cambio de personal, gestión, tecnológico y/o ambiente	Cuando alguno de estos cambios se presenta, esto genera que el equipo no cuente con sinergia y por lo tanto aumente el costo y tiempo de entrega del proyecto	
	Falta de una estimación adecuada en los proyectos	Esto se presenta cuando no tienen una adecuada gestión del coste, esto significa que no identificaron todas las actividades que generan un costo dentro del proyecto.	Implementación
	Errores técnicos del personal	Estos se presentan durante el proceso de codificación del proyecto, cuando el personal no tiene una buena comunicación y, por ende, los trabajos individuales ocasión errores en el proyecto.	
	Falta de monitoreo por el gerente de proyectos	Este riesgo se presenta por falta de control o seguimiento del proyecto, ya que, esto es lo que permite verificar si el proyecto va marchando según lo planificado.	

Tabla 4. Riesgos de calidad en proyectos de software

Tipo de Riesgo	Riesgo	Descripción del Riesgo	Etapa del Ciclo de Vida
Riesgos de calidad de software	Mala definición de los requerimientos	Esto se da porque en la primera etapa de la metodología, el cliente aprueba los requisitos presentados por el gerente de proyecto, sin prestar mayor atención.	Pruebas
	Falta de reevaluación en la calidad del producto	Se presenta cuando le restan importancia a realizar un seguimiento continuo del proyecto que se está llevando a cabo y así saber si está cumpliendo con los objetivos.	
	Los productos que no son adecuados con una estrategia total	Cuando el producto de software es entregado sin cumplir totalmente los requerimientos definidos al comenzar el desarrollo de este.	

Tabla 5. Riesgos empresariales en proyectos de software

Tipo de Riesgo	Riesgo	Descripción del Riesgo	Etapa del Ciclo de Vida
Riesgos empresariales de software	Fracaso en el presupuesto total	Cuando los requerimientos se encuentran en un cambio constante en todo el ciclo de vida del software.	Mantenimiento
	Fracaso en el en la entrega del producto	Este se presenta cuando no se tiene en cuenta el riesgo del tiempo en la entrega del producto.	
	Fracaso en la gestión debido al cambio en el personal	Cuando no tienen una adecuada gestión de recursos humanos para mitigar este riesgo en el índice de rotación de personal	

Finalmente, como todos conocemos el software es un producto intangible, abstracto y cambiante, y estas características propias del software, hace que los requerimientos varíen constantemente. Esta es una de tantas razones, por las que un proyecto de software siempre cuenta con riesgos en todas las diferentes etapas de su desarrollo, pero en algunas con una mayor proporción que en otras; por ende, se debe llevar a cabo una gestión de riesgos para mitigarlos, y así, lograr que el proyecto cumpla lo establecido por el cliente**.** Por lo cual, no interesando qué metodología de ciclo de vida se esté llevando a cabo para el desarrollo del proyecto, siempre es vital efectuar esta práctica de forma proactiva.

5.2.2 Ingeniería de Requerimientos:

La identificación y el análisis de requerimientos es esencial para que el proyecto cumpla con el funcionamiento que el cliente solicitó y además para que se reduzcan los costos y el riesgo durante el desarrollo. El principal objetivo de la recolección de los requerimientos es entender lo que el cliente y usuario espera que realice el producto final. Entendiéndose por requerimientos las funciones y propiedades esenciales que identifican al sistema (Fuentes, 2011).

Para la identificación de los requerimientos de un sistema es necesario llevar a cabo un proceso, el cual se muestra en la Figura 3 (Fuentes, 2011).

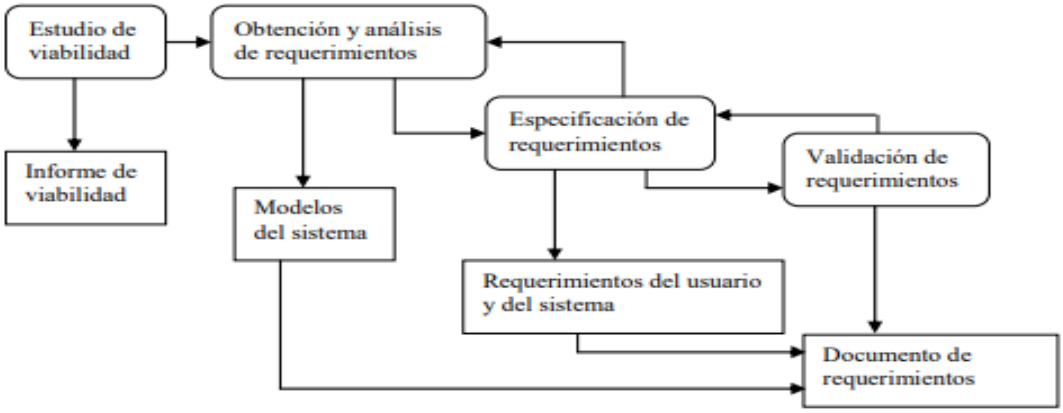


Figura 3. Proceso de ingeniería de requerimientos

Fuente: (Fuentes, 2011)

Los requerimientos de software se dividen en 2 categorías: requerimientos funcionales y requerimientos no funcionales.

- Requerimientos funcionales:

Son declaraciones de los servicios que debe proporcionar al sistema, la forma en que responde a entradas particulares y como actúa en situaciones específicas. Estos requerimientos también pueden dar a conocer explícitamente lo que el sistema no debe hacer (Fuentes, 2011).

- Requerimientos no funcionales:

Definen las restricciones con las que cuenta el sistema como la fiabilidad, el tiempo de respuesta y la capacidad de almacenamiento. Estas surgen de las necesidades del usuario, a causa de las restricciones de presupuesto, políticas de la organización, la necesidad de interoperabilidad con otros sistemas de software o hardware o factores externos (Galipienso, Martínez, Lizán, & Jover, 2005).

5.2.3 Metodologías:

El desarrollo de software es rico en oportunidades estratégicas, pero está sujeto a múltiples fuentes y altos niveles de incertidumbre, como los costos de desarrollo son irrecuperables, es importante manejar la incertidumbre.

- Software Engineering Institute (SEI)'s Software Risk Management (SRM): Es una metodología de riesgos de software la cual ayuda a la prevención, mitigación e identificación de los riesgos. Esta metodología se basa en la comunicación y es proceso iterativo durante todo el desarrollo del proyecto (Dhlamini, Nhamu, & Kachepa, 2009).
- Capability Maturity Model Integration (CMMI) / Integración del modelo de madurez de capacidades (CMMI): Es una metodología la cual cuenta con el propósito de identificar posibles problemas antes de que ocurran, para que las actividades de manejo de riesgos puedan ser planeadas e invocadas según sea necesario a lo largo de la vida del proyecto. Cuenta con 5 niveles de riesgo, que van de 1 a 5, entre más nivel tenga significa que el nivel de impacto es menor (Dhlamini, Nhamu, & Kachepa, 2009).
- Project Management Body of Knowledge (PMBOK): Es un guía de gestión de proyectos y un reconocimiento internacional estándar, que proporciona los fundamentos de la gestión de proyectos. Según (Dhlamini, Nhamu, & Kachepa, 2009) para esta guía, la gestión de riesgos comprende una serie de procesos que son: Planificación de gestión de riesgos, Identificación de riesgo, Análisis cualitativo de riesgos, Análisis cuantitativo de riesgos, Planificación de respuesta al riesgo, Monitoreo y control de riesgos.
- Practical Threat Analysis (PTA): Es una metodología calculadora de modelado de amenazas y una herramienta de evaluación de riesgos de amenazas que ayuda a evaluar los riesgos operativos y de seguridad en los sistemas y a construir una política de mitigación de riesgos adecuada. La PTA recalcula automáticamente el riesgo de amenazas y contrarresta las prioridades de implementación y proporciona a los tomadores de decisiones un plan de mitigación actualizado que refleja los cambios en las realidades

de las amenazas. En la Figura 4 se puede ver modelo de amenaza de la PTA (RANKOVIĆ & IVANOVIĆ, 2018).

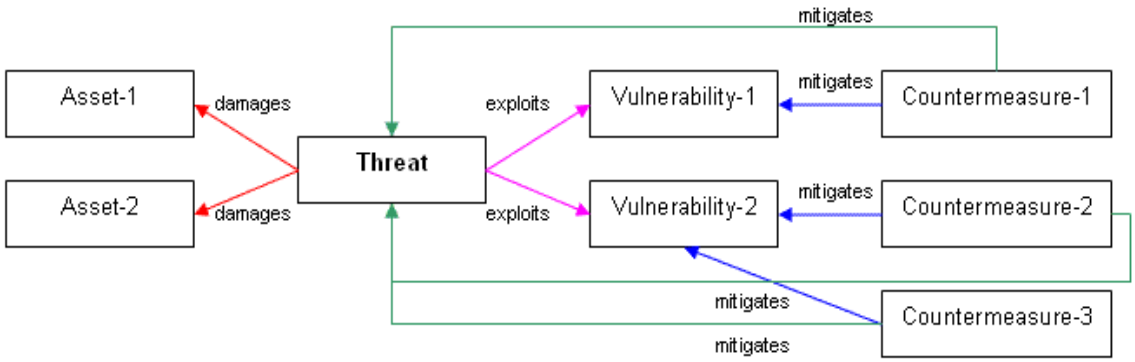


Figura 4. Funcionamiento de Practical Threat Analysis (PTA)

Fuente: (RANKOVIĆ & IVANOVIĆ, 2018).

5.2.4 Herramientas:

La mayoría de las herramientas de gestión de riesgo, generalmente, cuentan con un repositorio electrónico de riesgos para registrar y actualizar riesgos. Este repositorio, se realiza mediante lo compilación de registros de riesgos los cuales se analizan en el ciclo de vida del proyecto de software (Dhlamini, Nhamu, & Kachepa, 2009).

A continuación, en la Tabla 6, se muestra cinco herramientas de gestión de riesgo para los proyectos de software, en la cual se expondrá las características generales de cada herramienta y el alcance en cada etapa del desarrollo del software de la metodología RUP, las cuales son Inicio, Elaboración, Construcción e Implementación.

Tabla 6. Herramientas para la gestión de riesgo en proyectos de software

HERRAMIENTA	CARACTERÍSTICAS GENERALES	ALCANCE EN CADA ETAPA
RISK IT (ISACA, 2019)	➢ Orientación de principio a fin sobre cómo gestionar los riesgos relacionados con TI. ➢ Una mejor comprensión de las responsabilidades y roles.	✔ Inicio ✔ Elaboración ✔ Construcción ✔ Transición

	➢ Permite a la empresa tomar decisiones apropiadas con conciencia del riesgo.	
Risk Radar Enterprise (RRE) (Dhlamini, Nhamu, & Kachepa, 2009)	➢ Permite una gestión y comunicación efectivas. ➢ Admite el ciclo de vida completo del historial de riesgos desde la identificación hasta su culminación. ➢ Brinda a los gerentes y a sus equipos la visibilidad en tiempo real y la comunicación abierta que necesitan a lo largo de sus actividades.	✔ Inicio ✔ Elaboración ✔ Construcción ✔ Transición
RiskyProject Professional (Intaver Institute, 2019)	➢ Crear y modificar la programación de proyectos ➢ Análisis y gestión de riesgos tanto cuantitativos como cualitativos. ➢ Integra el análisis de riesgos de costos y determina el impacto de los riesgos e incertidumbres en el cronograma.	✔ Inicio ✔ Elaboración ✔ Construcción ✔ Transición
Full Monte 2016 (Barbecana, 2018)	➢ Informes personalizables, incluidos gráficos de barras para fechas, duraciones y costos. ➢ Admite subproyectos externos con enlaces entre proyectos. ➢ Análisis de sensibilidad y valores optimistas y pesimistas.	✔ Inicio ✔ Elaboración ✔ Construcción ✔ Transición
Project Risk Analysis (Katmar Software, 2018)	➢ Gráficos fáciles de leer que le dan al estimador una apreciación general de los riesgos de manera rápida y efectiva. ➢ Evalúa los riesgos de capital en los proyectos. ➢ Calcula la contingencia requerida utilizando el método estadístico probado conocido como simulación Monte Carlo.	✔ Inicio ✔ Elaboración ✔ Construcción ✔ Transición

5.2.5 Etapas de pre análisis y análisis

A continuación, se presentan los objetivos de las dos primeras etapas del desarrollo del software (Pre análisis y análisis) (Universidad de Antioquia, 2019).

Tabla 7. Objetivos de las etapas de Pre análisis y análisis

Pre-análisis	Análisis
✓ Aclarar y comprender la solicitud del proyecto ✓ Determinar alcance del proyecto ✓ Lograr un conocimiento general y estructural de los requerimientos de información ✓ Planear alternativas de desarrollo ✓ Evaluar costos y beneficios ✓ Determinar factibilidad operativa, técnica y económica ✓ Planear las actividades generales ✓ Requerimientos globales	✓ Conocer el sistema de información actual ✓ Conocer todos los requerimientos y necesidades de información adicionales ✓ Revaluar puntos de la etapa de planeación ✓ Formular modelos lógicos y estructurados que representen el nuevo sistema ✓ Servir de enlace entre la parte administrativa y la parte técnica ✓ Fomentar la participación de gerente y empleados en todo el proceso ✓ Requerimientos específicos del sistema

5.3 ESTADO DEL ARTE

En esta sección se muestra una recopilación de 18 documentos encontrados en bases de datos académicas, los cuales se analizaron y se obtuvieron los diferentes métodos para identificar y mitigar los riesgos en los proyectos de software. Para la selección de estos documentos se utilizaron las siguiente palabras claves: *Risk Management, Software Projects* y *Risk Management Methodologies.* El rango de búsqueda de estos documentos está definido entre el año 2012 y 2020.

En la Figura 5 se puede observar los documentos seleccionados por año de publicación.

Documentos encontrados de análisis de riesgos ordenados por año

Año	Documentos
2012	3
2013	1
2014	2
2015	7
2016	2
2018	1
2019	1
2020	1

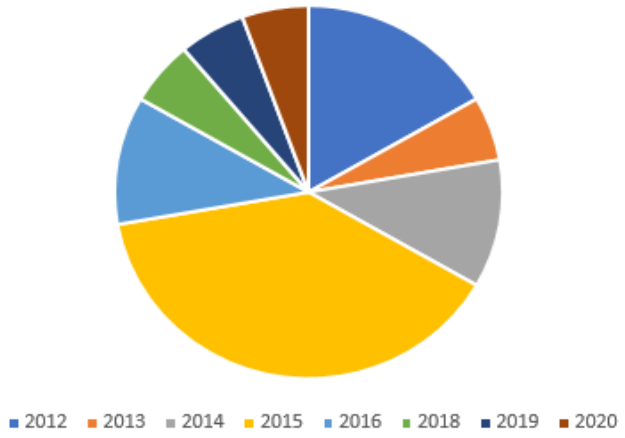


Figura 5. Gráfica circular con los documentos del estado del arte ordenados por año

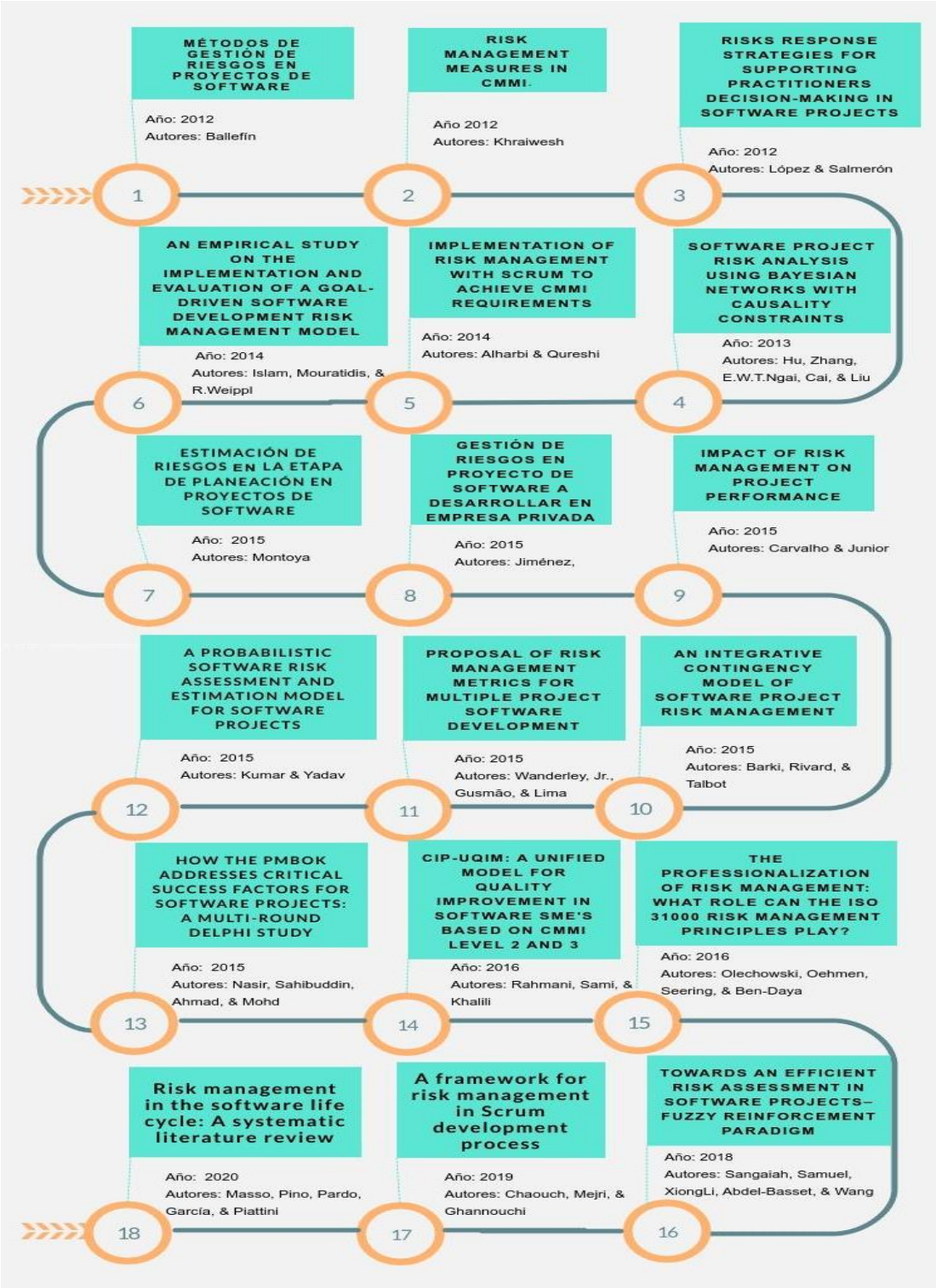
Los documentos recuperados son un 62,5% Artículos de revista y 37,5% Documentos de Investigación. Seguidamente, en la Tabla 8 se puede observar las bases de datos en donde se consultaron los documentos que serán presentados en esta sección.

Tabla 8. Análisis del Estado del arte

Análisis del estado del arte	
Bases de Datos Consultadas	• Science Direct • Google Scholar • Taylor & Francis
Cantidad de Referencias Recuperadas	• Science Direct: 10 documentos recuperados • Google Scholar: 6 documentos recuperados • Taylor & Francis: 2 documentos recuperados
Fecha de Búsqueda	Agosto 5, 2019
Criterios de Búsqueda	Rango de Fecha de Búsqueda: 2012 y 2020
	Palabras claves para la investigación: • Risk Management • Software Projects • Risk Management Methodologies

Los documentos recuperados fueron ordenados en orden cronológico de publicación, en una línea de tiempo, para así, contar con una visión global de ellos. (Ver Figura 6).

Figura 6. Línea de tiempo de los documentos seleccionados



A continuación, se presenta una síntesis de los documentos encontrados en la literatura, relacionados con esta investigación:

En este estudio se encontró bibliografía en la cual se evidenciaba diferentes métodos de gestionar el riesgo en los proyectos de software, documentos como *Software project risk analysis using Bayesian networks with causality constraints* y *A Probabilistic Software Risk Assessment and Estimation Model for Software Projects*, presentaron el teorema de Bayes (BBN) y redes bayesianas (BN), en el primer documento mencionado sirvió de base las BN para el diseño de un marco para el análisis de causalidad de los riesgos de los proyectos de software (Hu, Zhang, E.W.T.Ngai, Cai, & Liu, 2013), mientras que el segundo documento, hizo uso de las Bayesian Belief Network (BBN) para la realización de un modelo que permite estimar la probabilidad general de riesgo del software en manera cualitativa (Bajo, Medio, Alto) y la probabilidad de ocurrencia de cada riesgo (Kumar & Yadav, 2015).

Otro método utilizado para ejecutar la gestión de los riesgos en los proyectos de software, son las métricas, *Proposal of Risk Management Metrics for Multiple Project Software Development* propuso dos métricas Pure Risk Point (PRP) y Exponential Risk Point (ERP) basándose en la métrica “Punto de riesgo”. PRP es ideal para calcular el número de riesgos presentes en el proyecto y ERP clasifica los riesgos según su aparición o impacto en niveles (Muy bajo, Bajo, Promedio, Alto y Muy alto), lo cual ayudará en la toma de decisiones durante el ciclo de vida del proyecto (Wanderley, Jr., Gusmão, & Lima, 2015). Por el contrario, *Risks Response Strategies for Supporting Practitioners Decision-Making in Software Projects* propone el uso de la técnica de análisis de importancia-rendimiento (IPA), para así, seleccionar la estrategia de respuesta más adecuada para los riesgos en los proyectos de IS/IT (Information Systems/Information Technologies) (López & Salmerón, 2012). *An empirical study on the implementation and evaluation of a goal-driven software development risk management model* utilizó el Modelo de gestión de riesgos de desarrollo de software (GSRM) para mostrar su efectividad en la

integración de las actividades de gestión de riesgo en la ingeniería de requisitos. Al aplicar este modelo se obtuvieron resultados positivos, ya que, este método es práctico y razonable (Islam, Mouratidis, & Weippl, 2014). En el documento *Métodos de Gestión de Riesgos en Proyectos de Software* se realizó una comparación entre los diferentes métodos de gestión de riesgo, como, Boehm, RiskIt, PMI-PRM, SAFE y RIMAN, donde se llegó a la conclusión, de que el más adecuado es RiskIt, ya que, no es necesario tener conocimientos previos en gestión de riesgo y además las actividades a realizar se pueden iterar durante el proyecto (Ballefín, 2012).

En el documento *Towards an efficient risk assessment in software projects–Fuzzy reinforcement paradigm* se propone un marco de evaluación para el riesgo de software el cual tiene un enfoque híbrido compuesto por, Método De Prueba Y Evaluación De Toma De Decisiones Difusas (DEMATEL), Toma De Decisiones Multicriterio Difusa (FMCDM) y TODIM (un acrónimo en portugués para la toma de decisiones interactiva y de atributos múltiples). Esta propuesta es efectiva para identificar y analizar los riesgos más notorios en los proyectos de software para así, ayudar a la toma de decisiones durante el ciclo de vida del producto (Sangaiah, Samuel, XiongLi, Abdel-Basset, & Wang, 2018).

An Integrative Contingency Model of Software Project Risk Management buscó probar la siguiente hipótesis: El rendimiento del proyecto de software está influenciado por un ajuste, el cual es definido como la medida en que el perfil de gestión de riesgos de un proyecto coincide con su exposición al riesgo. Finalmente, se concluye que los proyectos de alta exposición al riesgo parecen requerir un perfil de gestión de riesgos diferente al de los proyectos de baja exposición al riesgo (Barki, Rivard, & Talbot, 2015). Otra hipótesis realizada para la gestión de riesgos fue la expuesta en el documento, *Impact of risk management on project performance: the importance of soft skills* en donde se presenta la correlación de los aspectos duros (Formación académica, Conocimientos, etc.) y blandos (Habilidades y competencias personales) de la gestión de riesgo con el éxito de los proyectos en cualquier área, incluyendo la de software. Con este estudio llegaron a la conclusión

de que los aspectos blandos de la gestión de riesgos cuentan con un mayor efecto en el éxito de los proyectos (Carvalho & Junior, 2015).

Según *CIP-UQIM: A unified model for quality improvement in software SME's based on CMMI level 2 and 3* el conjunto de modelos más utilizados para la mejora de la calidad de toda la organización son CMMI-DEV, ISO 9001 y PMBOK (Rahmani, Sami, & Khalili, 2016).

En los documentos *Implementation of Risk Management with SCRUM to Achieve CMMI Requirements* y *Risk Management Measures in CMMI* se evidenció el uso de las CMMI para la gestión de riesgos de software. En el primer documento, se planteó una propuesta llamada “Registro de riesgo” para implementar la gestión de riesgos en la metodología SCRUM en base al estándar CMMI, para controlar la entrada y salida de los riesgos (Alharbi & Qureshi, 2014). En el segundo documento, fueron definidas una serie de medidas generales basadas en las siete prácticas especificadas por la CMMI para la gestión de los riesgos en los proyectos de software, de esta manera se proporciona una mejor visión de la gestión de riesgos del proyecto, además, poder controlar y evaluar los procesos del producto de software (Khraiwesh, 2012).

How the PMBOK Addresses Critical Success Factors for Software Projects: A Multi-round Delphi Study muestra una lista de 26 factores críticos del éxito en los proyectos de software entre los cuales se encuentra la gestión del riesgo. Estos factores fueron analizados con la guía PMBOK, lo cual dio como resultado que esta guía abarca varios aspectos de la gestión de proyectos incluyendo la gestión de riesgos, pero las diferencias entre cada proyecto hacen que esta guía requiera personalización para su uso (Nasir, Sahibuddin, Ahmad, & Mohd, 2015). En el documento *Gestión de riesgos en proyecto de software a desarrollar en empresa privada* se identificó los riesgos más importantes en cada una de las cinco fases de desarrollo, al igual se realizó un análisis cualitativo de cada riesgo, por medio del cual encontraron la probabilidad de ocurrencia y el impacto de cada uno. De esta manera proponen un plan de respuesta a los riesgos más críticos, todo esto basándose en la guía PMBOK (Jiménez, 2015).

En el documento *A framework for risk management in Scrum development process*, los autores realizaron un marco para la gestión de riesgos aplicable a los proyectos que estén empleando la metodología ágil SCRUM, este marco fue basado en las fases establecidas por la guía PMBOK, dichas fases fueron adaptadas, en base a una encuesta realizada a 65 profesionales que habían trabajado con SCRUM en proyectos de mediano y gran tamaño. Finalmente, cabe mencionar que este marco no ha sido aplicable en un proyecto (Chaouch, Mejri, & Ghannouchi, 2019).

Estimación de Riesgos en la Etapa de Planeación en Proyectos de Software es un documento en donde se presenta un modelo de gestión de riesgo en base de los lineamientos internacionales como CMMI, Normas ISO 9000 y PMBOK, el cual está compuesto por cuatro fases (Planificación de la gestión de riesgos, Identificación de riesgos, Análisis de riesgos y Evaluación de riesgos), las cuales permiten al gerente de proyectos evaluar la probabilidad de éxito del desarrollo de software, desde una etapa muy temprana en el ciclo de vida (Montoya, 2015).

The professionalization of risk management: What role can the ISO 31000 risk management principles play? Es un documento donde se presenta una investigación empírica y una discusión de los once principios de la Norma de Gestión de Riesgos ISO 31000: 2009 investigando su interrelación y probando su efecto en los resultados de los proyectos de ingeniería, a través, de una encuesta a gran escala a profesionales de esta área. Con el estudio evidenciaron que el cumplimiento de los principios de gestión de riesgos a un alto nivel es un factor importante para alcanzar mejor los objetivos de costos, cronogramas, técnicos y de clientes, además de lograr una ejecución de proyecto más estable (Olechowski, Oehmen, Seering, & Ben-Daya, 2016). Por último, *Risk management in the software life cycle: A systematic literature review,* es un documento en el cual se presenta una revisión sistemática de la literatura en el ámbito del riesgo de software, con el fin de identificar, analizar y caracterizar el estado del arte; detectando así los riesgos que se pueden presentar durante el ciclo de vida de este (Masso, Pino, Pardo, García, & Piattini, 2020).

5.4 MARCO LEGAL Y POLÍTICO

Las normas para la gestión de riesgo en los proyectos de software son escasas. En este apartado profundizaremos en tres de ellas, las cuales se pueden ver en la Tabla 9.

Tabla 9. Normas para la gestión de riesgo en los proyectos de software

Estándar ISO/IEC 16085:2006	Estándar IEEE 1540-2001	Norma ISO 31000
Define un proceso para la gestión de riesgo durante el ciclo de vida del software, además se puede aplicar simultáneamente a sistemas y software. Cabe resaltar que este estándar es pago (ISO, 2006)	Expone un proceso continuo de gestión de riesgo, durante todas las etapas del ciclo de vida y admite la adquisición, el suministro, el desarrollo, la operación y el mantenimiento de productos y servicios de software. Se espera que este estándar sea adaptado a organizaciones y proyectos para satisfacer las necesidades específicas, además, respalda implícitamente el uso de herramientas y técnicas para que el proceso de gestión de riesgos sea continuo durante todo el ciclo de vida (IEEE, 2001).	Es una norma internacional la cual muestra a las organizaciones los principios y directrices para gestionar los riesgos. El objetivo de esta consiste en que cualquier organización no importando el tipo o tamaño de la empresa, pueda gestionar de manera efectiva los riesgos; además se recomienda que cada organización desarrolle un marco de trabajo para integrar la gestión del riesgo en cada actividad. (ISOTools, 2019).

Cabe resaltar, que se encuentran más normas y estándares para la gestión de riesgos en los proyectos de software, como, ISO/IEC 15288, la cual establece procesos para describir el ciclo de vida y define procesos y terminología que se puede aplicar a lo largo del proyecto para gestionar cada etapa (ISO, 2015) y la norma ISO/IEC 12207, donde se establece un marco para los procesos del ciclo de vida, con terminología definida, y además, cuenta con procesos, actividades y tareas para la adquisición de un producto de software (ISO, 2008), entre otras normas y estándares.

Finalmente, es importante destacar que todas estas normas mencionadas anteriormente rigen la gestión de riesgos en las micros, pequeñas y medianas empresas. La Ley 905 de 2004 muestra que los criterios utilizados para determinar

el tamaño de una empresa son: el valor de los activos y el número de los empleados (Ver Tabla 10).

Microempresas	Pequeñas Empresas	Medianas Empresas
Planta de personal hasta de 10 empleados	Número de empleados desde 11 hasta 50	Desde 51 hasta 200 empleados
Activos inferiores a 500 salarios mínimos legales vigentes, exclusiva la vivienda.	Activos entre 501 y 5000 salario mínimo legal mensual vigente.	Activos desde 5001 hasta 30000 salarios Mínimo Mensual Legal Vigente.

Tabla 10. Clasificación de las empresas

Fuente: (Grupo BanColombia, 2018)

6. DISEÑO METODOLÓGICO

En esta sección es presentado el diseño o marco metodológico el cual hace referencia al tipo, diseño, población, muestra, técnicas e instrumentos utilizados para la realización de la investigación (Franco & Camacho, 2008). Por lo cual, en este apartado se puede encontrar dicha información.

6.1 FASES Y ACTIVIDADES

Para el desarrollo del proyecto de trabajo de Grado, se definieron cuatro etapas principales, relacionadas con los objetivos específicos planteados en este proyecto. Seguidamente, son presentadas las actividades que son comunes en cada una de estas etapas:

- Búsqueda de información en las diferentes plataformas académicas sobre las metodologías, herramientas y riesgos de los proyectos de software.
- Documentación de los resultados obtenidos según la búsqueda anterior.

Finalmente, se presentarán las actividades particulares en cada una de ellas:

Etapa 1:

Descripción: Analizar las metodologías de gestión de riesgos más utilizadas en los proyectos de software como pautas para la elaboración del prototipo funcional de aplicación web.

Actividades relacionadas:

- Identificación de las metodologías utilizadas para la gestión del riesgo.
- Síntesis de revisión de la literatura acerca de las metodologías.

Etapa 2:

Descripción: Revisar las herramientas tecnológicas utilizadas actualmente como apoyo a la gestión de riesgos en los proyectos de software para la identificación de factores diferenciales en el prototipo funcional de aplicación web.

Actividades relacionadas:

- Identificación de las herramientas tecnológicas actualmente utilizadas para la administración del riesgo de software.
- Identificación de las etapas donde actúan cada una de las herramientas tecnológicas actualmente utilizadas para la gestión de riesgos.

Etapa 3:

Descripción: Implementar un prototipo funcional de aplicación web para la identificación temprana de los riesgos en los proyectos de software.

Actividades relacionadas:

- Identificación los riesgos más frecuentes en las etapas de pre análisis y análisis en los proyectos de software.
- Implementación del prototipo para la detección temprana de riesgos en las etapas de pre análisis y análisis
    - Análisis de los requerimientos para la construcción del prototipo.
    - Diseño técnico y funcional del prototipo.
    - Implementar el prototipo según las actividades de análisis y diseño realizadas previamente.
    - Pruebas Unitarias al prototipo según los casos de pruebas definidos.

Etapa 4:

Descripción: Validar la aplicabilidad del prototipo funcional de aplicación web orientado a la detección temprana de los riesgos en un proyecto de software.

Actividades relacionadas:

- Identificación de las empresas de software en la ciudad de Bucaramanga donde se aplicará el prototipo web para la detección temprana de los riesgos.
- Aplicación del prototipo funcional web en los proyectos de software pertenecientes a las empresas identificadas anteriormente.
- Síntesis con los resultados obtenidos a partir de la aplicación del prototipo.

6.2 MATERIALES, TÉCNICAS E INSTRUMENTOS

6.2.1 Técnica: Análisis de documentos

Instrumento asociado: Análisis de documentos

Es una técnica que se basa en fichas bibliográficas para el análisis del material impreso. Es utilizado para la realización del marco teórico de estudio para una investigación de calidad, se aconseja usar simultáneamente dos o más técnicas de recolección de datos, con el fin de contrastar y complementar la información (BERNAL, 2010).

6.2.2 Técnica: Pruebas Unitarias al Prototipo de Aplicación

Instrumento Asociado: Pruebas Unitarias al Prototipo de Aplicación

Es utilizada para verificar el correcto funcionamiento del código desarrollado. Estas pruebas pueden repetirse tantas veces se desee o sea necesario, para así, lograr cubrir la totalidad del código de nuestra aplicación (Universidad de Alicante, 2019).

6.2.3 Técnica: Encuesta

Instrumento Asociado: Encuesta

La encuesta es una técnica en la que se dispone de un cuestionario o un conjunto de preguntas con el fin de obtener información de las personas. Además, esta es una de las técnicas de recolección de datos más utilizadas, aunque siempre existe el sesgo de personas encuestadas. (BERNAL,2010).

6.3 POBLACIÓN Y MUESTRA

El prototipo de detección de riesgos mencionado en este documento tiene como objetivo una población determinada, en este caso a los directores de proyectos de software, ya que, conocen la dinámica de gestión de proyectos, además son los que están implicados directamente en estas primeras etapas y tienen el conocimiento necesario de los riesgos que se pueden presentar en un proyecto de software.

Para evaluar el prototipo se tomó una muestra de 4 proyectos de software, de diferentes empresas cada uno. La información recolectada se obtuvo por parte de los directores de proyectos que estuvieron a cargo de ellos.

7. CRONOGRAMA DE ACTIVIDADES

A continuación, se presenta el cronograma de actividades para la ejecución del trabajo de grado (Ver Tabla 11).

Tabla 11. Cronograma

8. PRESUPUESTO

A continuación, se presenta el presupuesto estimado para la realización del trabajo de grado, en donde se tiene en cuenta personal, equipos y software, materiales, entre otros. (Ver Tabla 12)

Tabla 12. Presupuesto

Rubros	Detalle del ítem	Descripción	Total
Personal	María Camila Rincón Ayala (Estudiante de pregrado)	Honorarios	9.600.000
	Diana Marcela Jaimes Jaimes (Estudiante de pregrado)	Honorarios	9.600.000
	Angélica María Zapata García	Honorarios	1.000.000
Equipos y Software	Computador	Personal	2.500.000
Materiales, Suministros y Bibliografía	Papelería	Personal	50.000
	Impresiones	Personal	20.000
Otros	Movilidad	Personal	3.840.000
Total			**26.610.000**

Fuente de Financiamiento: La presente Tesis se financiará con recursos propios en un 100%.

Nota: El rubro de la remuneración de las estudiantes de pregrado, se estimó para los 4 primeros meses un pago mensual de $800.000 para cada una, en este periodo de tiempo se realizó investigación y en los 4 meses restantes los cuales fueron de programación y aplicación del prototipo, se planteó un pago de $1.600.000 mensuales para cada integrante del proyecto.

9. DESARROLLO DE LA TESIS

En esta sección se presentan los pasos que se llevaron a cabo para la realización del prototipo, los cuales se muestran en la Figura 7.

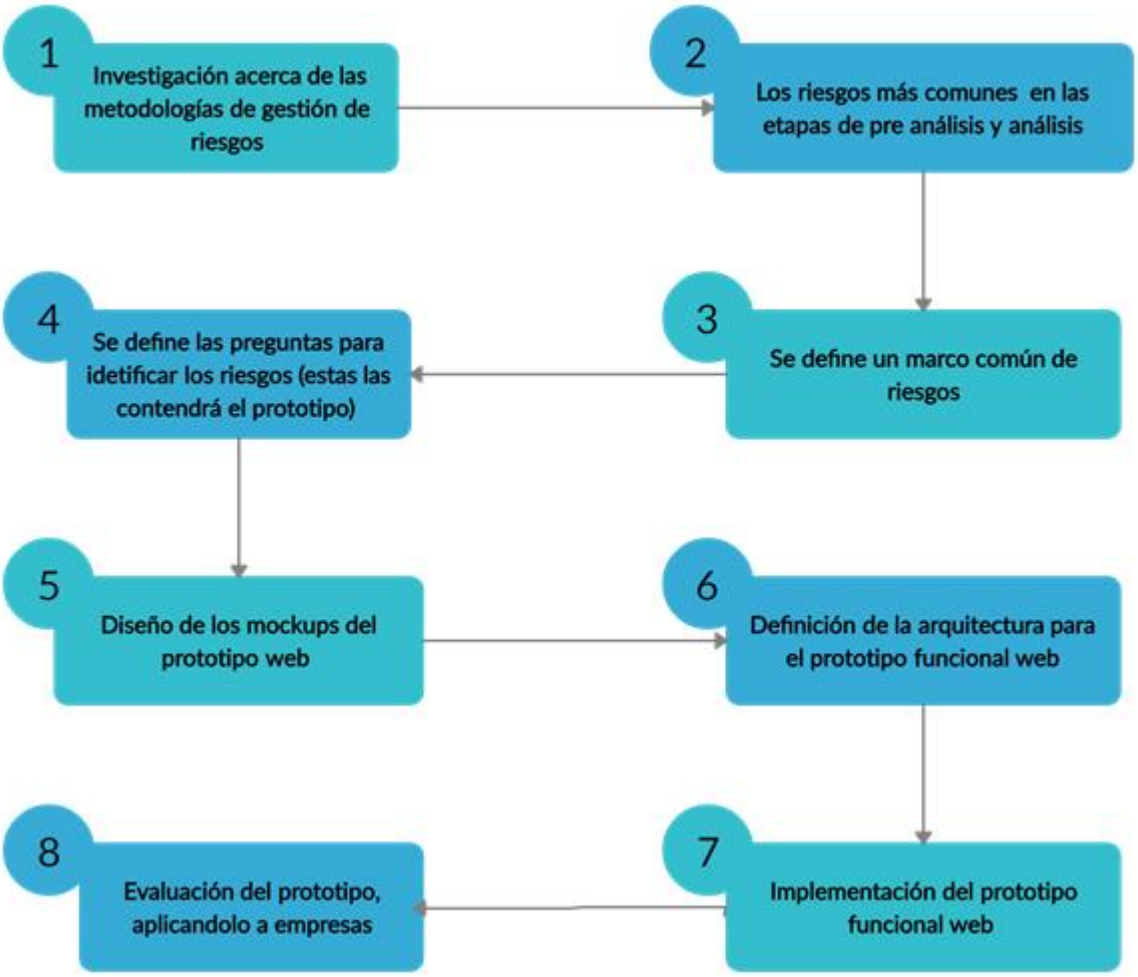


Figura 7. Procedimiento llevado a cabo para el desarrollo del trabajo de grado

A continuación, se ampliará la información acerca de cada uno de los pasos mostrados en la Figura 7.

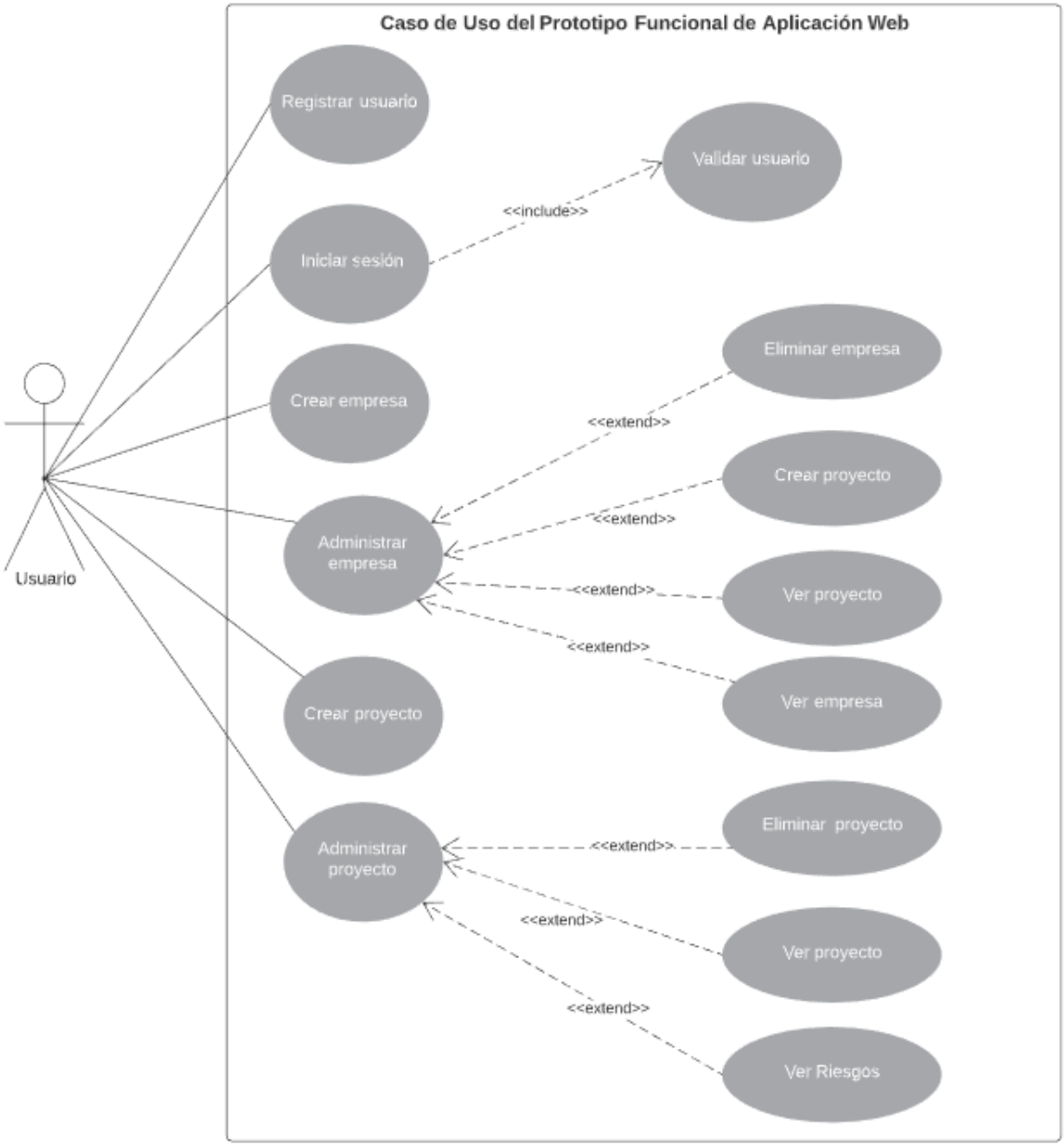
Primeramente, se empezó con una investigación acerca de las metodologías de gestión de riesgos más utilizadas actualmente, lo cual sirvió de guía para identificar el proceso que se lleva a cabo para detectar los riesgos en los proyectos de software, igualmente esta investigación permitió definir los riesgos más comunes

que se pueden presentar en las etapas de un proyecto, en este caso se enfocó en las etapas de pre análisis y análisis, para luego realizar un plan de respuesta a los riesgos detectados (Ver anexo 14.1).

Al tener la información anteriormente definida, se continuó al siguiente paso donde se definió un marco común de riesgos mediante una investigación realizada en diferentes fuentes electrónicas y libros (Ver Marco Teórico). En base a esta información se definió una serie de preguntas, mediante las cuales el prototipo identificará los riesgos de costo, requerimientos, desarrollo, calidad y aspectos empresariales del proyecto de software que se lleva a cabo (Ver anexo 14.3).

El tercer paso abordado en la elaboración del prototipo web fue el diseño del funcionamiento de este, por medio de un diagrama de casos de uso, diagrama de actividades y los diagramas de secuencias, seguidamente se elaboró el diseño de mockups del prototipo funcional de aplicación web.

Figura 8. Diagrama de casos de uso del prototipo funcional web



A continuación, en la Figura 9, 10, 11 y 12, se muestran 4 diagramas de secuencias del prototipo, los cuales son, Iniciar Sesión, Crear Cuenta, Crear Empresa y Crear Proyectos. Se profundizará un poco más en el último, ya que, aquí es donde se evidencia de una manera más amplia la forma en como el prototipo funcional de aplicación web estima los riesgos. Primeramente, se realizó una investigación

acerca de los riesgos que se pueden presentar en las etapas de pre análisis y análisis, así mismo, las preguntas que iba contener el formulario de crear proyecto. Cabe mencionar, que cada pregunta tiene asignado un riesgo de las primeras etapas del desarrollo del software.

Figura 9. Diagrama de secuencia de Crear cuenta

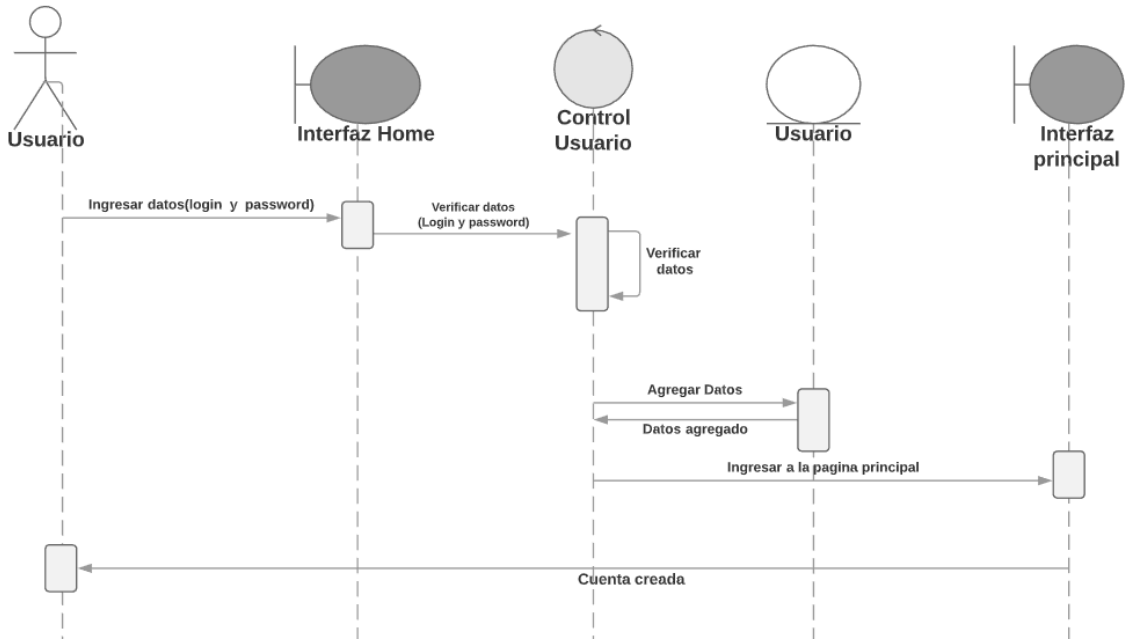


Figura 10. Diagrama de secuencia de Iniciar Sesión

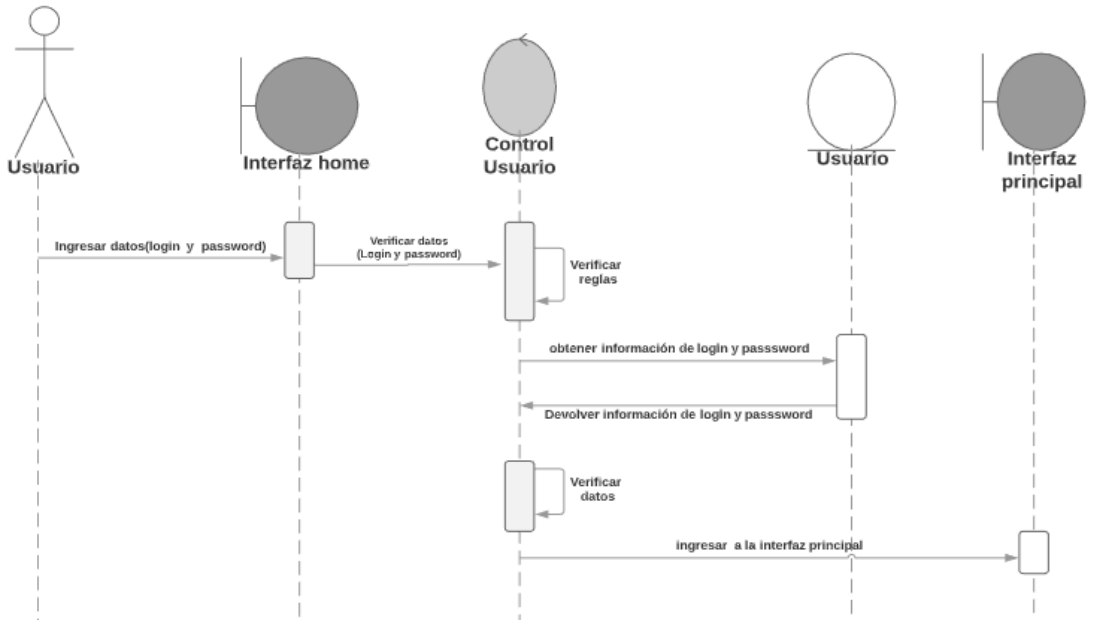


Figura 11. Diagrama de secuencia de Crear empresa

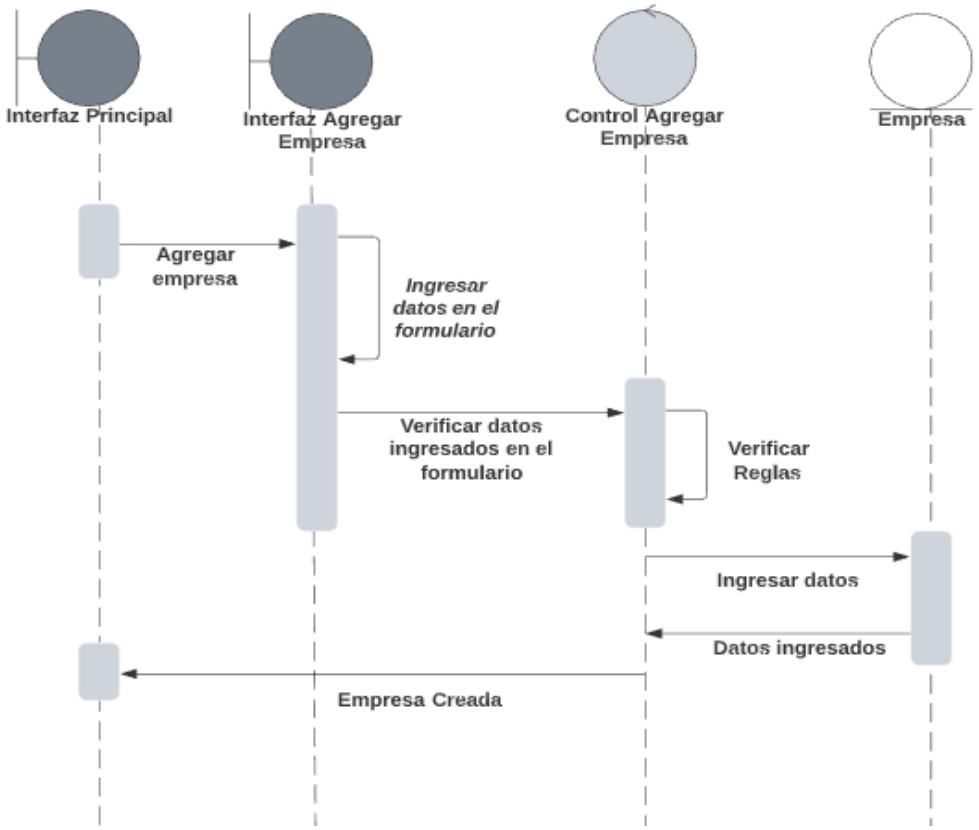
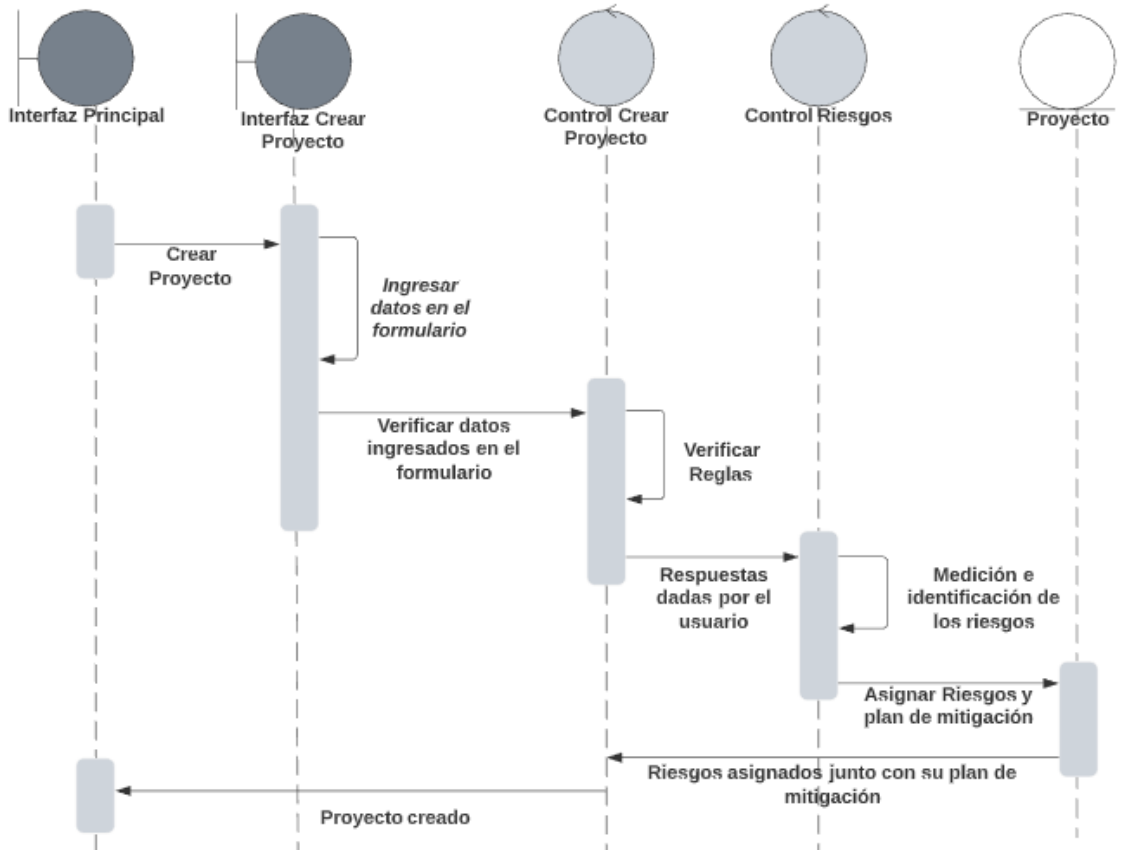


Figura 12. Diagrama de secuencia de Crear proyectos



9.1 Cálculo de los Riesgos

Para la detección de los riesgos del proyecto ingresado, el prototipo evalúa la información diligenciada por el usuario acerca del proyecto, y como se expuso anteriormente, cada pregunta cuenta con un riesgo el cual fue previamente asignado; por ende, al completar el formulario se procede a detectar si es posible que se presente un riesgo en el proyecto. Para esta identificación, el prototipo revisa las respuestas dadas por el usuario y examina si estas provocan que haya riesgos en el proyecto, igualmente, con los datos dados por éste se realiza la medición de los riesgos, es decir, se le asigna el impacto que tendrá cada uno de estos, para ello, se comparan todas las respuestas de las preguntas asociadas a determinado riesgo.

Con el funcionamiento del prototipo web ya definido, se procede a realizar el diseño de mockups en base al diagrama de actividades, esto se evidencia en el Capítulo 10.

El cuarto paso es el desarrollo del prototipo web, para ello primeramente se definió la arquitectura necesaria para el proyecto y seguidamente se inició con el desarrollo del mismo, en el Capítulo 10 se explica detalladamente este paso.

Finalmente, se realizó una evaluación del prototipo el cual se aplicó a diferentes proyectos del área metropolitana de Bucaramanga, esta información se especifica en el capítulo de Diseño metodológico en la sección Población y Muestra.

10. PROTOTIPO DE DETECCIÓN DE RIESGOS

En esta sección se mostrará los pasos que se siguieron para desarrollar el prototipo y el funcionamiento interno de este, el cual se observa en la Figura 13.

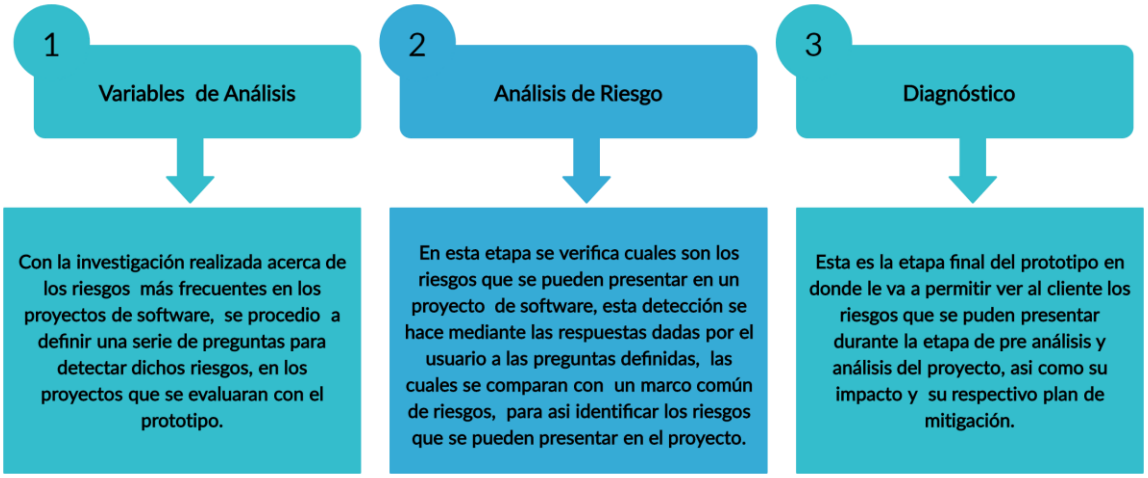


Figura 13. Funcionamiento interno del prototipo de detección de riesgos

A continuación, En la Figura 14 se evidencia más detalladamente las actividades o acciones que el prototipo de detección de riesgo permite realizar, al igual que el flujo de dichas acciones. Con esto, es posible conocer en una manera amplia como actúa el prototipo y así hacer más sencillo y organizado el momento de desarrollo.

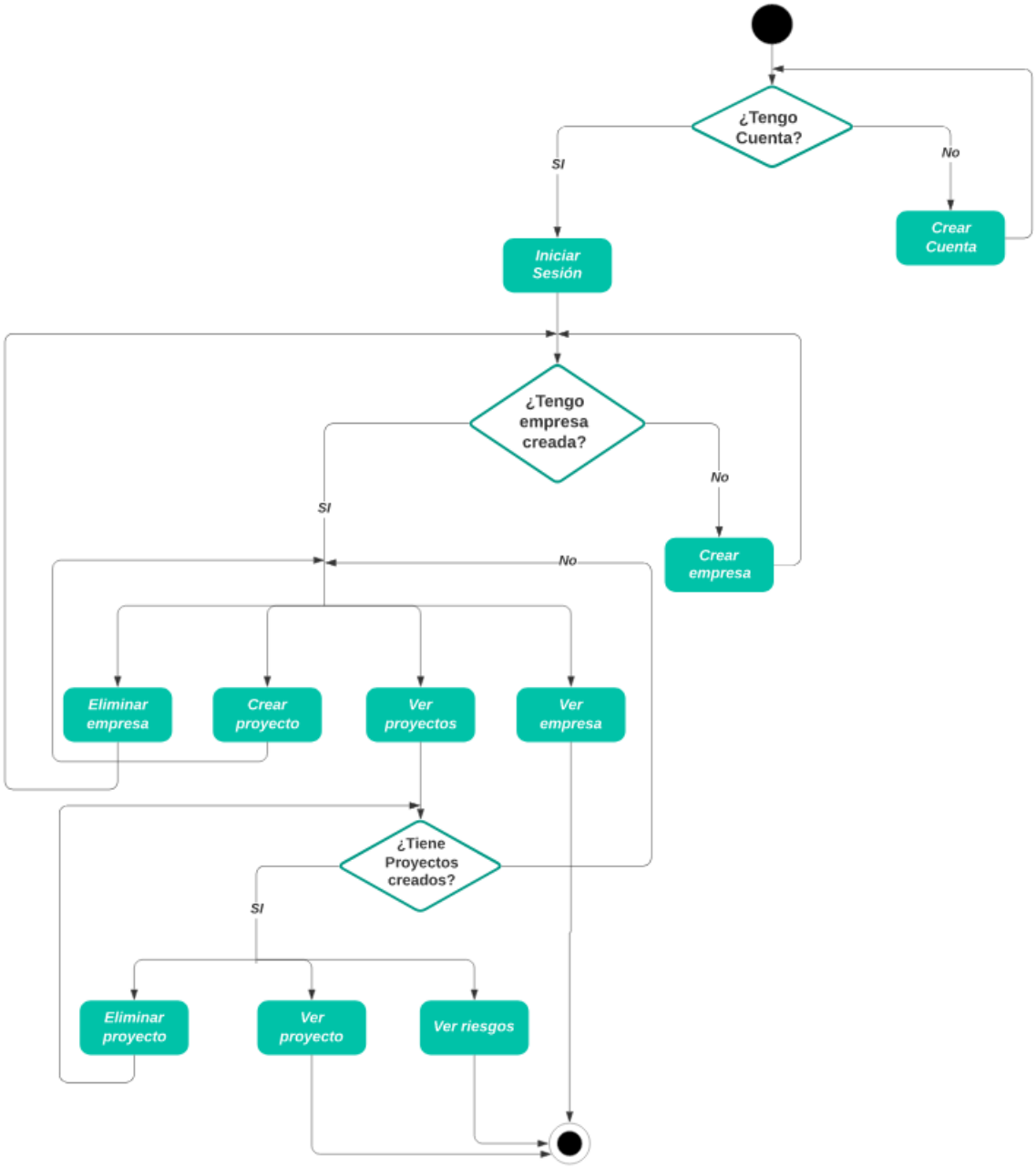


Figura 14. Diagrama de actividades del prototipo web

Seguidamente, se explicará cada paso que se realizó para el desarrollo del prototipo funcional web propuesto en este trabajo de grado.

- Diseño de mockups

El diseño de mockups se elaboró en físico para plasmar las ideas que nos ayudaron a obtener la interfaz más óptima para el usuario y el tipo de tema abordado, factores como colores, estilos, fuentes, botones, entre otros fueron definidos en esta etapa. Esto sirvió para tener seguridad respecto a los detalles del desarrollo de este y saber el nivel de complejidad que conlleva el realizarlo.

- Definición de la Arquitectura

En este paso, se definió las diferentes herramientas y tecnologías que se utilizara en el prototipo web, seguidamente se menciona cada una de ellas.

1. HTML5, CSS, JavaScript

   Estas tecnologías fueron utilizadas para la realización del apartado visual del prototipo funcional.

2. PHP

   Este lenguaje fue seleccionado por la gran cantidad de características ofrecidas entre las cuales destacan ser multiplataforma, ser orientada a objetos y permitir la conectividad a una cantidad variada de bases de datos. Esta tecnología se puede definir como: Un lenguaje de alto nivel que se interpreta del lado del servidor permitiendo así que las páginas se vean en cualquier ordenador (Puertas, 2014).

   Figura 15. Logo PHP

3. MYSQL

Este sistema de gestión de base de datos es ampliamente conocido, y se utilizó para el prototipo realizado localmente; debido a que se tiene conocimientos previos. Finalmente fue reemplazado por el gestor de PostgreSQL, ya que, era el que brindaba gratuitamente la plataforma Heroku.

4. PostgreSQL

   Es un sistema de gestión de base de datos relacional orientada a objetos, su código fuente está disponible de forma libre, además utiliza el modelo cliente/servidor y hace uso de multiprocesos, garantizando así la estabilidad del sistema (Ordoñez, Ríos, & Castillo, 2017).

   Debido a que el prototipo fue alojado en la nube y que la base de datos local MYSQL es paga en la plataforma de Heroku, se realizó una transición de la base de datos a una gratuita ofrecida por la plataforma, en este caso PostgreSQL.

   Cabe mencionar que surgieron múltiples problemas durante esta etapa de la transición de base de datos, debido a que fue la primera vez en la que se trabajó con PostgreSQL y que se alojaba una aplicación en la nube.

Figura 16. Logo PostgreSQL

5. Heroku

   Esta es una plataforma de servicio de computación en la Nube, de la cual se hizo uso para que el prototipo web sea más accesible a las empresas o proyectos que lo utilizarán. De esta manera la aplicación del prototipo de detección de riesgos se da de una manera más sencilla, rápida y segura.

Figura 17. Logo Heroku

- Prototipo

El prototipo final se realizó en dos fases, la primera fue el desarrollo del prototipo localmente, el cual se hizo mediante las tecnologías PHP, MYSQL y para el apartado visual HTML, CSS y JavaScript. La segunda fase fue el alojar el prototipo en la nube, esto se realizó a través de las mismas tecnologías exceptuando el gestor de base de datos, ya que, en esta etapa se trabajó con PostgreSQL.

A continuación, se muestra las pantallas del prototipo de detección de riesgos realizado para este proyecto de grado. El cual permite realizar las funciones de agregar empresa, crear proyectos y ver riesgos del proyecto.

Home: Esta es la página principal del prototipo donde se muestra una corta descripción del enfoque o características que tiene el prototipo. Igualmente, desde aquí se abre paso a las páginas de Iniciar Sesión o de Registro de Usuarios.

Figura 18. Pantalla Home

Inicio Sesión: Esta interfaz le permite al usuario ingresar a la aplicación, para esto necesita tener registrado un correo y una contraseña.

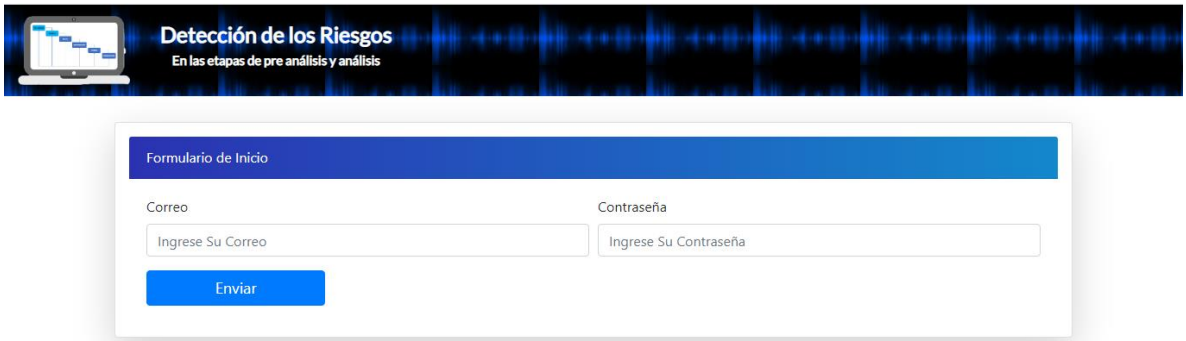


Figura 19. Pantalla Inicio Sesión

Registro: En caso de no tener un usuario creado, es necesario crearlo para así poder acceder a las funciones del prototipo y para esto es necesario diligenciar el formulario que se muestra en la Figura 20.

Figura 20. Pantalla Registro

Agregar Empresa: Lo primero que se debe hacer al ingresar al prototipo, es crear una empresa, esto se hace diligenciando el cuestionario que se muestra en la Figura 21 y 22, y finalmente guardar los datos al pulsar el botón *Guardar*.

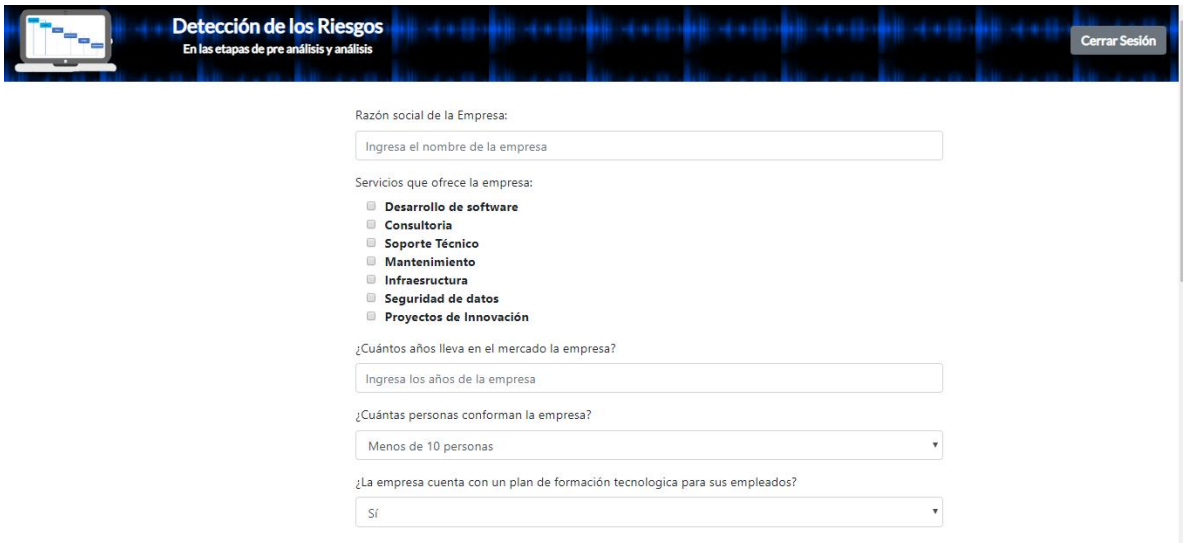


Figura 21.Pantalla Agregar empresa parte 1

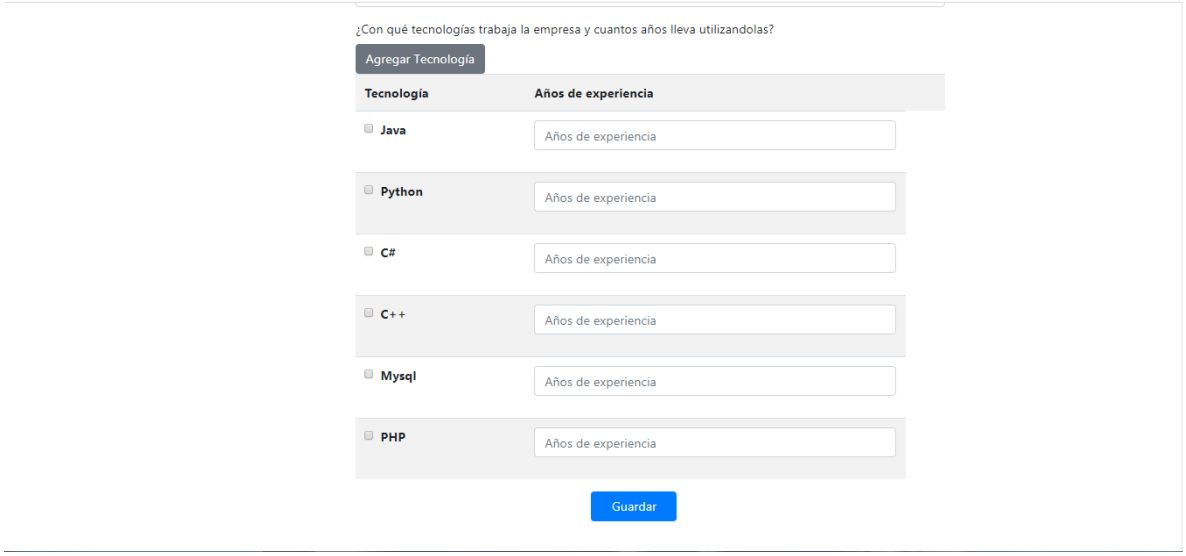


Figura 22. Pantalla Agregar empresa parte 2

Página Principal: Luego de crear empresa, se podrá ver esta interfaz la cual le permite al usuario las siguientes opciones: Ver la empresa que creó, Crear proyectos (El número de proyectos que desee), Ver proyectos y Eliminar empresa (Al eliminar la empresa se elimina también los proyectos de la empresa)

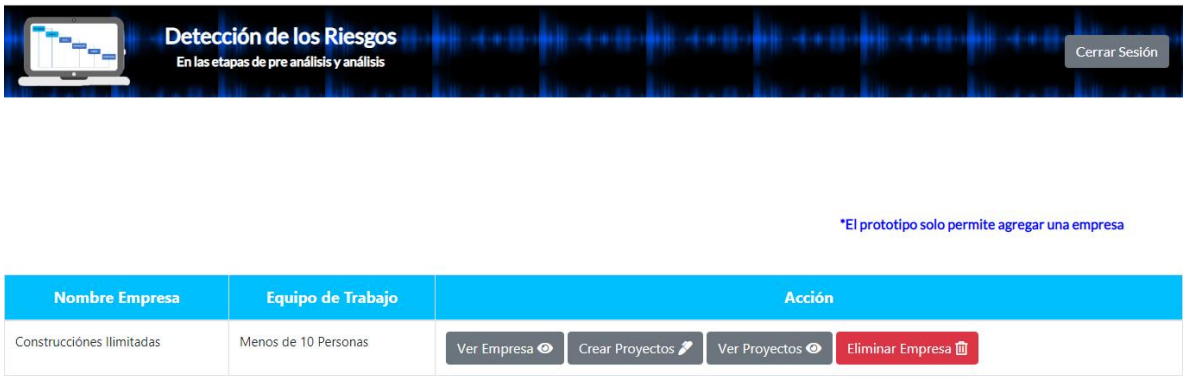


Figura 23. Pantalla de la página principal

Crear Proyectos: Esta interfaz será visualizada al dar clic en el botón Crear Proyectos que se muestra en la interfaz Principal, cabe resaltar que se puede crear tantos proyectos como necesite el usuario. Para crear estos proyectos se debe

contestar las preguntas que se ven en la Figura 24 y seguidamente pulsar el botón de *Guardar*, para que de esta forma el prototipo analice los datos y arroje los riesgos que tiene el proyecto.

(Aquí solo se muestran las primeras preguntas, cabe resaltar, que hay muchas más)

Figura 24. Pantalla Crear Proyecto

Ver Proyectos: Esta interfaz le permite ver todos los proyectos que tiene creados, cada proyecto creado tiene la opción de Ver Proyecto, Ver Riesgo y Eliminar Proyecto, como se muestra en la Figura 25.

Figura 25. Pantalla Ver Proyectos

Ver Riesgos del Proyecto: Al dar clic en el botón *Ver Riesgos* en la interfaz "Ver Proyectos" (Figura 25), se muestra esta pantalla la cual permite ver los riesgos del

proyecto e igualmente su impacto (bajo, medio y alto) y su respectivo plan de mitigación, como se muestra en la Figura 26.

Riesgos Bajos: 2 | Riesgos Medios: 2 | Riesgos Altos: 3

Riesgo	Impacto	Plan de Mitigación
Falta de suficientes habilidades técnicas en el equipo de trabajo	Alto	El equipo de trabajo y el director del proyecto deberían tomar capacitación con respecto a la tecnología con la que se va a trabajar, ya que, no es suficiente tener nociones de la tecnología sino que se necesita tener un conocimiento pleno de esta para que se pueda hacer entrega del software en el tiempo establecido.
Escasez de personal	Medio	Es necesario que el Gerente del Proyecto puede utilizar herramientas manuales o automatizadas para obtener un presupuesto estimado. Las herramientas de asignación pueden ser hojas de cálculo simples o una herramienta compleja de estimación del presupuesto. Además tiene que estimar el personal necesario y adecuado para llevar a cabo el proyecto teniendo en cuenta la experiencia y habilidades necesarias para cumplir con las actividades del cronograma del proyecto y de esta manera entregar el proyecto en el tiempo establecido.
Conocimiento inadecuado de las herramientas requeridas	Alto	La empresa debe brindar las capacitaciones que requiera para los proyectos que esté llevando a cabo, ya que, es algo crucial para mejorar los conocimientos y habilidades; esto generará que el personal sea más competentes aumentando la productividad y calidad del trabajo.
Falta de acuerdo entre el cliente y el desarrollador	Alto	Ya que el cliente no cuenta con una claridad plena de los requerimientos del proyecto y de la tecnología que se va a utilizar. Se considera necesario mantener una comunicación constante con este e Informar al cliente de los próximos pasos y de las tareas pendientes que se van a llevar a cabo, al igual si el Gerente de proyecto considera necesario, sería ideal realizar capacitaciones al cliente acerca de las tecnologías.
Ambigüedad de requerimientos	Bajo	La asistencia e intervención del cliente durante las reuniones realizadas es de vital importancia, ya que, el producto final debe satisfacer las expectativas o requerimientos que el cliente desea. Por lo cual, si el cliente no interviene activamente o no tiene claro lo que quiere, es importante que el Gerente de proyecto en el transcurso de las reuniones le realice múltiples preguntas al igual es necesario guiarlo en el tema que se esté llevando a cabo. Finalmente, es importante realizar un número representativo de reuniones durante el desarrollo del proyecto.

Figura 26. Pantalla Ver Riesgos del Proyecto

11. RESULTADOS OBTENIDOS

En esta sección son presentados los resultados del Proyecto de Trabajo de Grado.

- Documento con el análisis de las metodologías de gestión del riesgo más utilizadas en los proyectos de software.
    - En este documento se presenta una tabla comparativa de las metodologías más utilizadas para realizar la gestión de riesgos en los proyectos de software. (Ver anexo 1).

- Documento con la revisión de las herramientas tecnológicas actualmente utilizadas para la administración del riesgo en los proyectos de software.
    - En este entregable se evidencia una tabla comparativa de las herramientas tecnológicas actualmente utilizadas para la gestión de riesgos en los proyectos de software. (Ver anexo 2).

- Documento con los ítems para detectar los riesgos en las etapas de pre análisis y análisis.
    - Este documento contiene las preguntas formuladas en el prototipo realizado para la detección temprana de riesgos en las etapas de pre análisis y análisis, al igual que el plan de mitigación para cada riesgo. (Ver anexo 3).

- Prototipo funcional de aplicación web para la detección temprana de los riesgos en los proyectos de software.
    - En este entregable se expone el proceso y las tecnologías que se tuvieron en cuenta para el diseño y desarrollo del prototipo, al igual que el funcionamiento del mismo. (Ver Capítulo 10)

- Documento con análisis de los resultados obtenidos al aplicar el prototipo funcional.
    - Este entregable contiene las conclusiones obtenidas luego de haber aplicado el prototipo a proyectos de software que se van a realizar en el área metropolitana de Bucaramanga. (Ver capítulo 12).

12. ANÁLISIS DE LOS RESULTADOS OBTENIDOS

En este capítulo se presenta un análisis acerca de los resultados dados por el prototipo web a los usuarios, igualmente, se muestra un análisis de una encuesta realizada a los usuarios, la cual se hizo con el fin de obtener la percepción que tienen estos acerca del prototipo.

12.1 Análisis resultados dados por el prototipo al usuario

El prototipo de detección de riesgos de este trabajo de grado, fue utilizado por desarrolladores y directores de proyectos de software. Cabe mencionar, que el prototipo evaluó un total de 7 proyectos, pero los proyectos diligenciados por los desarrolladores no se tuvieron en cuenta para el análisis de este, por tres aspectos, como la falta de intervención en las primeras etapas del proyecto, no cuentan con la información necesaria para evaluar los riesgos del proyecto y la nula experiencia en la gerencia de proyectos. Por esta razón se realizó el análisis de 3 proyectos usados por los directores de proyectos pertenecientes a las siguientes empresas: Fundación Oftalmológica de Santander (Foscal), Sistema de Expertos EU y Switch Up S.A.S.

A continuación, se presenta un análisis de los riesgos arrojados por el prototipo a los diferentes proyectos ingresados por los usuarios. Cabe mencionar una vez más, que estos riesgos depende de las características propias del proyecto.
En la Figura 27 se evidencia, la cantidad de veces que el error fue arrojado por el prototipo y qué categoría era (Bajo, Medio, Alto).

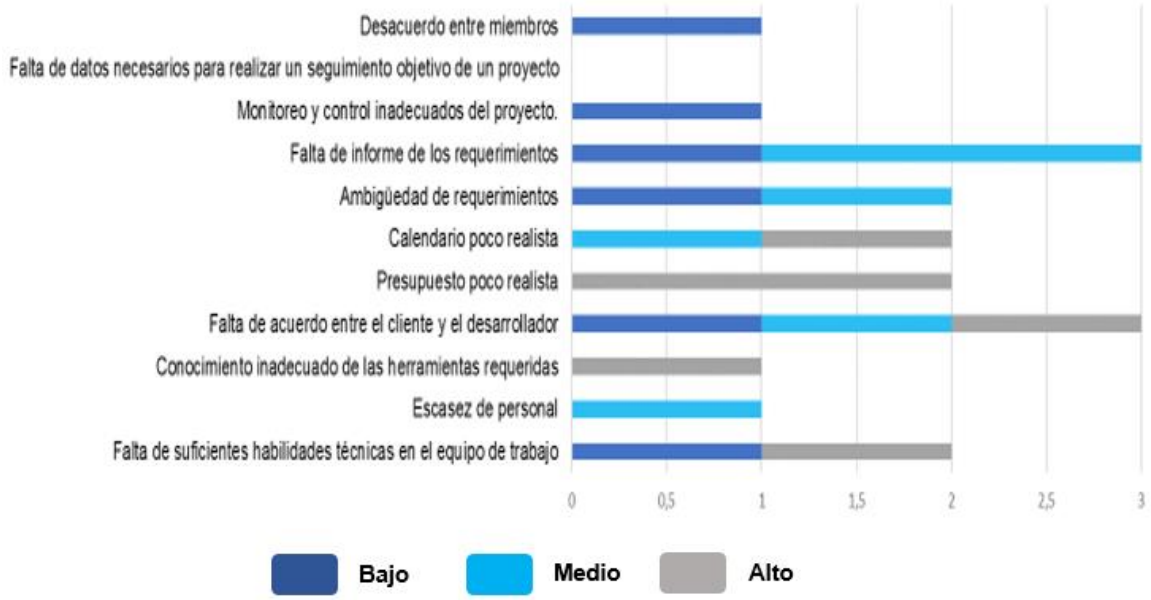


Figura 27. Frecuencia de los riesgos que se presentaron en los proyectos

Como se evidencia en la figura anterior, hubo un riesgo que no se presentó en ningún proyecto evaluado por el prototipo el cual fue, falta de datos necesarios para realizar un seguimiento objetivo de un proyecto; esto, debido a que, proveen la información actualizada de los requerimientos a todo el personal y cuentan un seguimiento, medición, análisis y evaluación del desempeño del proyecto.

Los riesgos que mayor número de veces se presentaron fueron:

- Falta de acuerdo entre el cliente y el desarrollador
- Falta de informe de los requerimientos

El primer riesgo mencionado se presentó en todos los proyectos debido a que el cliente no cuenta con el suficiente conocimiento de los requerimientos y de la tecnología que se va a utilizar en el proyecto de software. El segundo riesgo mencionado anteriormente, se presenta en los 3 proyectos en uno de ellos con un impacto bajo a causa de que no se realizaba documentación del cambio de los requerimientos y en los restantes con un impacto medio, esto a causa de que en

algunos casos el cliente no le da importancia al análisis y extracción de los requerimientos.

Finalmente, se muestra las razones por las que calendario y presupuesto poco realista se presentaron en dos de los proyectos, el primero riesgo mencionado se evidenció en casi todos los proyectos de los usuarios, en uno de ellos debido a que se le resta tiempo a la etapa de análisis, lo cual genera que el proyecto no se realice de forma correcta y en el proyecto restante se dio, ya que, no se distribuye bien el tiempo ni para la etapa de análisis ni para la etapa de desarrollo, por ende se produce un mayor riesgo al no tener claro el tiempo que se le debe aplicar a cada etapa. El último riesgo nombrado, se presentó en dos proyectos en un rango alto, debido a que el dinero necesario para el desarrollo del proyecto (Salarios del personal, Costos de hardware e infraestructura, Gastos de energía, agua, alquiler del local, ordenadores, etc.) era menor al presupuestado.

12.2 Encuesta de percepción del cliente sobre el prototipo

Esta encuesta fue realizada a una población de 3 personas, las cuales trabajan como directores de proyectos; las empresas encuestadas están ubicadas en el área metropolitana de Bucaramanga. A continuación, se muestra cada pregunta con el resultado obtenido.

1. ¿Considera que el prototipo es de fácil uso?

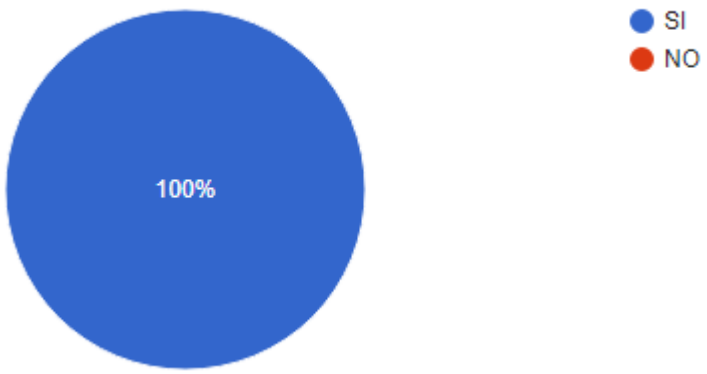


Figura 28. Gráfica circular de la pregunta número 1 de la encuesta

2. En una escala del 1 al 5, cómo calificaría el prototipo en el diseño de la interfaz (Donde 1 es nada satisfecho y 5 es totalmente satisfecho)

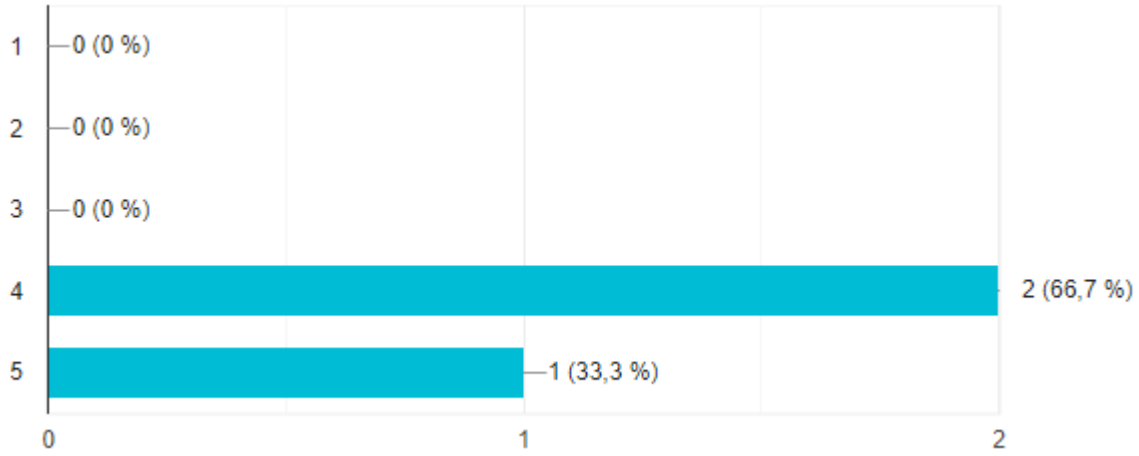


Figura 29. Gráfica de barras de la pregunta número 2 de la encuesta

3. En una escala del 1 al 5, cómo calificaría el funcionamiento del prototipo, es decir, la accesibilidad, navegación, etc. (Donde 1 es nada satisfecho y 5 totalmente satisfecho)

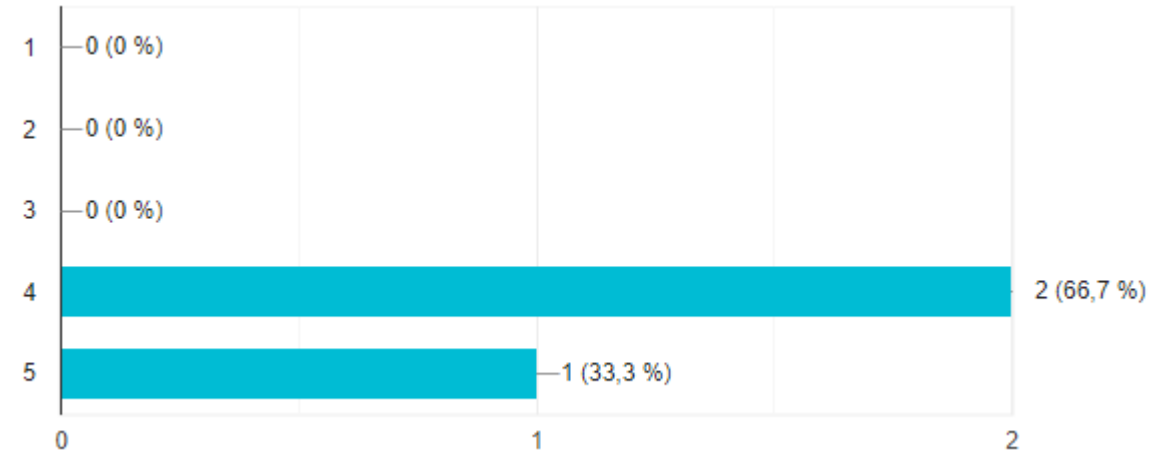


Figura 30. Gráfica de barras de la pregunta número 3 de la encuesta

4. ¿Considera que el prototipo le ayudó a mejorar la detección de los riesgos en su proyecto de software?

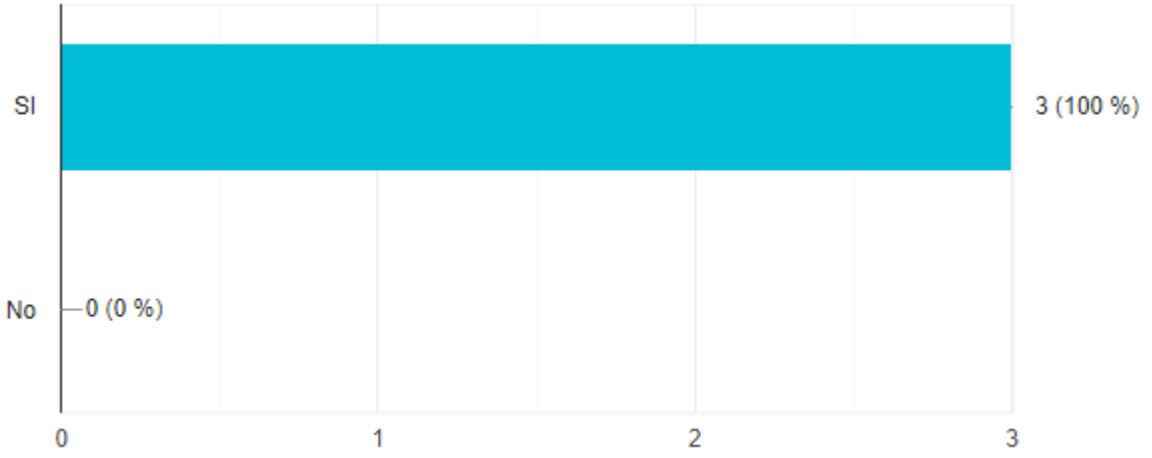


Figura 31. Gráfica de barras de la pregunta número 4 de la encuesta

5. ¿Qué aporte considera que le dio el prototipo a su proyecto?

- El proceso de contestar las preguntas ayuda a ver el proyecto desde diferentes perspectivas
- Claridad sobre las mediciones específicas que tenemos que realizar, debido a que por lo general los riesgos los consideramos de una manera más general y no tan detallada.

- Análisis temprano de riesgos

6. ¿Qué aspecto le gusto más del prototipo?

- A pesar de solicitar bastante información, lo hace de forma fácil y ordenada.
- Sin duda, la respuesta esperada del mismo. El prototipo responde según lo que se espera al momento de usarse, es conciso en su usabilidad.
- Análisis del proyecto para determinar riesgos.

7. ¿Como usuario que funcionamiento le agregaría al prototipo?

- Sería bueno que presentará la información resumida de alguna forma gráfica.
- La estética de los formularios distrae un poco, se pueden aprovechar mejor los espacios vacíos de la página y hacer que el proceso de llenado de formularios sea más fluido para que la experiencia sea integral.
- Si. Gráficas estadísticas.

8. Según la empresa a la que pertenece el proyecto, como considera la magnitud del proyecto: ¿Grande, Mediano o Pequeño?

- Mediano
- Mediano
- Grande

9. ¿Qué tipo de proyecto fue el ingresado por usted? Web / móvil

- Web
- Web
- Web y móvil

10 ¿El proyecto era: Una actualización de un proyecto existente / Nuevo proyecto?

- Nuevo
- Nuevo
- Nuevo

11. Si la anterior respuesta fue "Actualización" ¿Los proyectos que el prototipo identificó eran similares al que se le presentaron en las anteriores versiones de ese proyecto?

- Ningún proyecto era una actualización

12.2.1 Perfil de los proyectos seleccionados en la muestra

De la información obtenida por parte de los usuarios se categorizó los proyectos teniendo en cuenta, el tamaño, el tipo y si es un proyecto nuevo o una actualización, del primer aspecto se evidencia que el tamaño de dos de ellos eran mediano y el restante grande, lo cual, al aplicar el prototipo, la empresa de tamaño grande reflejo que esta característica afectó, en que, se presentaran un número mayor de riesgos, debido a que a mayor magnitud mayor complejidad, igualmente varios factores relevantes no se tuvieron en cuenta en el proyecto, lo cual generó este resultado. El segundo aspecto identificado en los proyectos, fue su tipo, de lo cual se obtuvo que todos eran de tipo web, exceptuando uno, el cual era también móvil, la última característica mostró que todos los proyectos ingresados en el prototipo eran nuevos, lo cual no permitió tener una comparación de los riesgos que se presentaron anteriormente en dicho proyecto con los detectados por el prototipo; aunque no se cuente con esos datos los directores de proyectos si mencionaron que el prototipo cumplía su objetivo que era identificar los riesgos de forma temprana, lo cual es clave para mejorar la probabilidad de éxito del proyecto. Igualmente, por medio de las preguntas restantes de la encuesta, se logró evidenciar que el prototipo fue de fácil uso, además cuenta con un diseño y funcionamiento adecuado. Finalmente, cabe mencionar que los aspectos que los directores de proyectos dieron acerca de las funcionalidades que se le podría agregar al prototipo, se tendrán en cuenta en el capítulo de trabajos futuros de este documento.

13. CONCLUSIONES

El objetivo principal del presente trabajo de grado era la realización de un prototipo web el cual ayudará a detectar los riesgos en las etapas de pre análisis y análisis de forma temprana, para así aumentar la probabilidad de éxito de los proyectos de software.

En la propuesta se efectuó un análisis comparativo de las metodologías más utilizadas en la gestión de riesgo, como instrumento para conocer más concretamente las etapas y actividades que son comunes entre las mismas, esta información fue relevante para la comprensión del funcionamiento del prototipo.

Seguidamente, se realizó un estudio comparativo de ciertas herramientas tecnológicas utilizadas en la gestión de riesgos en los proyectos de software, entre los aspectos evaluados se encuentra, la información que se debe ingresar, la facilidad de uso, la información que falta en la herramienta, entre otros. Con esto, se detectó qué características debería tener el prototipo y que lo diferencia de las herramientas analizadas.

La solución que se plantea para disminuir el número de proyectos que no son entregados o no cumplen completamente con los requerimientos del cliente, es la implementación de un prototipo funcional de aplicación web, el cual permitirá de una forma rápida evaluar los posibles riesgos que se pueden presentar en los proyectos de software, basándose en aspectos como las características del cliente, del equipo de trabajo y de los requerimientos funcionales.

Para sustentar el uso del prototipo para la detección temprana de los riesgos en los proyectos de software, se realizó la aplicación de este, a proyectos que estuvieran en las primeras etapas de desarrollo, en donde se encontró que los riesgos que más se presentaron y tuvieron en común los proyectos, fueron la Falta de informe de los requerimientos y falta de acuerdo entre cliente y desarrollador, el mencionado primeramente se presentó por dos distintas razones, en algunos proyectos debido

a que, el director no llevaba la documentación del cambio de los requerimientos, por el contrario, en otros proyecto se dio porque el cliente no le daba importancia a la identificación de los requerimientos del proyecto. El último riesgo nombrado se identificó a causa de que, el cliente no tiene claridad sobre los requerimientos e igualmente no cuenta con conocimientos sobre la tecnología con la que se desarrollara el proyecto.

El prototipo fue utilizado por directores de proyectos de diferentes empresas, a los cuales se les efectuó una retroalimentación para conocer más concretamente la experiencia con respecto a la identificación de los riesgos y el diseño; frente a esto, se obtuvieron resultados positivos, debido a que los usuarios mencionaron que les aportó un análisis temprano de los riesgos, ayudándolos a tener una mejor claridad sobre las medidas que deben tomar. Esto hace ver que se cumplió la hipótesis principal planteada en este documento.

Finalmente, cabe mencionar que la realización de este trabajo de grado permitió que los autores del prototipo funcional de aplicación web adquirieran amplios conocimientos acerca de la importancia de llevar a cabo una gestión de riesgos en los proyectos de software. Igualmente, acerca de las arquitecturas utilizadas para el desarrollo del prototipo, entre las cuales, cabe resaltar las plataformas que ofrecen servicios para subir los proyectos a la nube, como la utilizada en el actual proyecto, la cual contribuyó a disminuir notablemente el costo en la infraestructura, y a su vez, permitió que la validación del prototipo por parte de los usuarios se realizará de una forma ágil.

14. RECOMENDACIONES

El presente capítulo tiene el propósito de dar las recomendaciones pertinentes que a juicio de los autores se deben tener en cuenta a la hora de abordar un proyecto similar al desarrollado en este documento, para así obtener mejores resultados en su ejecución.

1. Para aplicar el prototipo a un número mayor de proyectos, se recomienda ampliar la detección de los riesgos a las demás etapas involucradas en un proyecto de software, ya que, en este estudio se tuvieron dificultades para encontrar proyectos en estado inicial.
2. Para realizar un análisis más detallado de los riesgos se recomienda dar un puntaje al impacto identificado en cada uno de ellos, de igual manera, agregar una evaluación de probabilidad de ocurrencia, para así, obtener la priorización de los riesgos.
3. Se recomienda el uso del prototipo funcional de aplicación web para la detección temprana de los riesgos favoreciendo a las empresas que lo ejecuten en sus proyectos de software, porque contribuye para que se actúe a tiempo disminuyendo así el impacto de los riesgos que se presentan en las primeras etapas, salvaguardando el bienestar económico, prestigio de la empresa y en especial el producto final de un proyecto.
4. Se recomienda validar el prototipo por medio de una encuesta a los usuarios que lo utilizaron, ya que, es importante conocer si el diseño y en especial el funcionamiento cumplieron con el objetivo trazado, en este caso, la detección temprana de los riesgos.

15. TRABAJOS FUTUROS

En esta sección se presentan opciones de trabajos a futuros que permiten darle continuidad al presente proyecto.

1. Como trabajo futuro se propone que el prototipo realice una detección de los riesgos para todas las etapas que estén involucradas durante el desarrollo de proyectos de software, lo cual, ayudará a tener un mayor control de estos.
2. Otra propuesta para trabajo futuro es agregar un funcionamiento que muestre los riesgos de forma gráfica, para conseguir que los usuarios tengan una mejor visualización y entendimiento de los resultados que obtuvieron.
3. Otra propuesta para trabajo futuro es aplicar el prototipo funcional a una muestra más grande de proyectos de software, para así, obtener una percepción de mayor validez sobre el cumplimiento del objetivo planteado en este trabajo de grado y de las características del prototipo.
4. También como trabajo futuro se pretende implementar tecnologías como la Inteligencia Artificial (IA), para que la identificación de los riesgos sea más exacta y completa.
5. Finalmente, se plantea desplegar este prototipo como una aplicación móvil, facilitando a los usuarios el uso de este y de igual manera lograr un mayor alcance.

16. ANEXOS

Anexo 1. ESTUDIO COMPARATIVO DE LAS METODOLOGÍAS MÁS UTILIZADAS PARA LA GESTIÓN DE RIESGO

En el siguiente anexo se presenta un documento con el estudio comparativo de las tres metodologías más utilizadas para la gestión de riesgos en los proyectos de software. (Ver Tabla 13)

Tabla 13. Tabla Comparativa de las metodologías más utilizadas para la gestión de riesgos

Autor	Metodologías	Número de etapas	Nombre de las Etapas	Actividades	Tipo de empresa/Proyecto en la que puede ser utilizada
Project Management Institute (PMI)	**PMBOK**	6	Planificar la Gestión de los Riesgos	- Deliverable is the Risk Management Plan	Cualquier proyecto no importando el tipo de empresa a que pertenezca
			Identificar los Riesgos	- Determinar los riesgos que pueden afectar el proyecto - Categorizar el riesgo en: Técnico, Gestión de proyectos, Organizacional o Externo - Documentar las características del proyecto	
			Realizar el Análisis Cualitativo de Riesgos	- Definir probabilidad y consecuencias - Recopilación de datos - Impacto por objetivo - Pruebas de supuestos - Clasificación de precisión de datos	
			Realizar el Análisis Cuantitativo de Riesgos	- Analizar numéricamente el efecto de los riesgos identificados, sobre los objetivos generales del proyecto, teniendo en cuenta: Riesgo individual y de proyecto, Análisis de sensibilidad y árbol de decisiones, Distribuciones de probabilidad y Métodos de simulación	
			Planificar la Respuesta a los Riesgos	- Desarrollar opciones y acciones para reducir las amenazas del proyecto	
			Controlar los Riesgos	- Implementar planes de solución a los riesgos - Dar seguimiento a los riesgos identificados - Monitorear los riesgos - Identificar nuevos riesgos - Evaluar la efectividad del proceso de gestión de los riesgos	

Software Engineering Institute (SEI)	**CMMI**	3	Preparar la gestión de riesgos	- Determinar las fuentes y categorías de los riesgos - Definir los parámetros de los riesgos - Establecer una estrategia de gestión de riesgo	Empresas en nivel de madurez 2(Gestionado), es decir, que cuenta con buenas prácticas en el desarrollo de los proyectos o que ha logrado todos los objetivos tanto genéricos como específicos
			Identificar y analizar los riesgos	- Identificar los riesgos: - Evaluar, categorizar y priorizar los riesgos	
			Mitigar los riesgos	- Desarrollar los planes de mitigación de riesgo - Implementar los planes de mitigación de riesgo	
International Organization for Standardization (ISO)	**ISO 31000**	6	Identificación en Gestión de Riesgos	- Definir los objetivos del proyecto haciendo énfasis en: Asignación de responsabilidad, Línea de negocio, procesos y subprocesos, Aspectos metodológicos, Procesos críticos y Riesgo inherente	Empresas de todos los tipos y tamaños
			Análisis del Riesgo	- Establecer la probabilidad de impacto que puede causar un riesgo, aplicando los siguientes procesos: * Metodología acorde al grado de madurez. * Cualitativa o cuantitativa * Registro de eventos o incidentes. * Controles y su grado de efectividad.	
			Evaluación del Riesgo	- Realizar la evaluación de los riesgos, teniendo en cuenta: * Criterios de riesgo * Apetito al riesgo * Priorización de riesgos	
			Tratamiento en Risk Management:	- Determinar el tratamiento que se le va aplicar al riesgo, mediante: * Planes de acción * Seguimiento de cumplimiento de plan de acción * Razonabilidad del control y/o medidas de tratamiento * Asignación de presupuesto * Indicadores de efectividad	

			Comunicación y consulta en Gestión de Riesgos	- Realizar reportes internos o externos - Informar y consultar - Nivelar y evolucionar indicadores de riesgo - Efectuar seguimientos al perfil de riesgo - Establecer la periodicidad de riesgo - Mantener eficiencia	
			Revisión y monitoreo	Efectuar en base a los indicadores establecidos previamente, lo siguiente: - Cumplimiento de políticas y procedimientos - Efectividad del sistema - Seguimiento al perfil de riesgo - Periodicidad - Responsabilidades	

A continuación, se ampliará la información respecto a las metodologías comparadas anteriormente, para así contar con un conocimiento más extenso de estas.

PMBOK

La Guía de los Fundamentos para la Dirección de Proyectos brinda unas pautas para la dirección de proyectos individuales y define conceptos relacionados con la dirección de proyectos. Detalla asimismo el ciclo de vida de la dirección de proyectos y los procesos relacionados, además el ciclo de vida del proyecto. El PMBOK contiene el estándar, reconocido a nivel global y la guía para la profesión de la dirección de proyectos. La gestión de riesgos en la guía PMBOK sigue una serie de procesos, en base a (PMI, 2013) y (Dhlamini, Nhamu, & Kachepa, 2009), se menciona a continuación estos procesos con sus categorías.

- Planificar la Gestión de los Riesgos: Es el proceso en el cual se define como se deben hacer las actividades de gestión de riesgos de un proyecto.

- Identificar los Riesgos: Se establece y clasifican los riesgos que pueden afectar al proyecto y finalmente, se documentan sus características.

- Realizar el Análisis Cualitativo de Riesgos: Se establece un orden de importancia a los riesgos para análisis o acción posterior, evaluando y combinando la probabilidad de ocurrencia e impacto de dichos riesgos.

- Realizar el Análisis Cuantitativo de Riesgos: Examinar el efecto de los riesgos identificados sobre los objetivos generales del proyecto.

- Planificar la Respuesta a los Riesgos: Las opciones y acciones planteadas para reducir las amenazas de los riesgos deben ser clasificadas en:
    - Apropiado

    - Económico
    - Oportuno, realista
    - Acordado (financiado)

- Controlar los Riesgos: Durante el desarrollo del proyecto se deberá evaluar la efectividad del proceso de gestión de cada uno de los riesgos, en las siguientes categorías
    - En marcha, acción continua
    - Riesgos controlados
    - Nuevos riesgos identificados
    - Eficacia de la gestión de riesgos evaluada

CMMI

CMMI The Capability Maturity Model Integration o en español La Integración del modelo de madurez de capacidades, es un conjunto de prácticas globales que impulsa el crecimiento de las empresas mediante la creación y comparación de las capacidades claves. Además, las mejores prácticas son de ayuda durante todo el ciclo de vida del producto (CMMI, 2019).

El CMMI clasifica a las empresas en cinco niveles de madurez, los cuales son basados en el desempeño y esfuerzo de mejora. Dentro de cada nivel, se cuenta con un conjunto de actividades las cuales ayudan a mejorar el rendimiento de la empresa. Estos niveles son (CMMI, 2019):

1. Inicial
2. Gestionado
3. Definido
4. Gestión Cuantitativa
5. Optimizado

Dentro de la metodología de CMMI, en el nivel de madurez 2, se encuentra el proceso de gestión de riesgo (RSKM), en donde se identifica los problemas o

riesgos que pueden ocurrir y se realiza un tratamiento o planificación, para mitigar los impactos negativos que estos riesgos podrían traer a los objetivos del proyecto. En base a (Chrissis, Konrad, & Shrum, 2009) la gestión de riesgo del CMMI, se divide en 3 etapas y cada una cuenta con actividades específicas, las cuales se presentan a continuación:

- Preparar la gestión de riesgos: Se trata de las acciones o estrategia usadas para identificar, analizar y mitigar los riesgos. Para esto, se definen las siguientes actividades:
  - Determinar las fuentes y categorías de los riesgos: Las fuentes son los factores tanto internos como externos que causan los riesgos del proyecto. La identificación de estas fuentes trae consigo la identificación temprana de los riesgos. Al igual, se debe contar con categorías de riesgos.
  - Definir los parámetros de los riesgos: Los parámetros son utilizados para comparar de una mejor manera los riesgos, ya que, sin estos sería difícil priorizar y medir la gravedad de cada riesgo. Los parámetros son: Probabilidad del riesgo, Consecuencia del riesgo (es decir, impacto) y Umbrales para disparar las actividades de gestión.
  - Establecer una estrategia de gestión de riesgo: La estrategia hace parte de la visión de éxito que se tenga del proyecto. En esta se encuentra elementos como: El alcance, Los métodos y herramientas, Los parámetros, entre otros.
- Identificar y analizar los riesgos: Aquí, los riesgos son identificados y analizados para determinar su importancia.
  - Identificar los riesgos: La identificación de problemas, riesgos y amenazas son una base sólida para la gestión de riesgo. Los riesgos identificados deben describirse de una manera clara, para que posteriormente sean analizados y gestionados. Esta actividad, nos da

como resultado, los riesgos identificados, junto con el contexto, consecuencias y la ocurrencia.

  - Evaluar, categorizar y priorizar los riesgos: Se evalúa y categoriza cada riesgo con los parámetros y categorías definidas en la etapa anterior. Y, finalmente, se priorizan los riesgos basándose en los parámetros asignados, como, el impacto.

- Mitigar los riesgos: Los riesgos deben ser tratados y mitigados, en donde sea apropiado, para así reducir los efectos negativos que estos, traen al proyecto.
  - Desarrollar los planes de mitigación de riesgo: Para los riesgos con mayor impacto o importancia, es fundamental realizar un plan de mitigación donde se plasme, Los niveles y límites aceptables de un riesgo y el momento en que se lanza el plan de mitigación, Las personas o grupo que tratará el riesgo, El coste/beneficio del plan de mitigación y El plan de contingencia.
  - Implementar los planes de mitigación de riesgo: Se monitorea los riesgos periódicamente y se pone en marcha el plan de mitigación de riesgos en el momento apropiado. En esta actividad, se puede descubrir nuevos riesgos o planes de mitigación.

ISO

ISO 31000 Gestión de Riesgos: Su finalidad es proporcionar un enfoque que mejore la gestión del riesgo, de forma sistemática y ofrecer diversidad de posibilidades para que las organizaciones alcancen los objetivos trazados. El documento normativo establece todos los procesos y principios que se deben llevar a cabo para la gestión del riesgo, en la que recomienda a las empresas el desarrollo, la implantación y el mejoramiento continuo como un importante componente de los sistemas de gestión (ISO, 2016).

Las fases para llevar a cabo la gestión del riesgo según (EALDE, 2018) son las siguientes:

- Identificación en Gestión de Riesgos
- Análisis del Riesgo
- Evaluación en Gestión de Riesgos
- Tratamiento en Risk Management
- Comunicación y consulta en Gestión de Riesgos
- Revisión y monitoreo

Anexo 2. HERRAMIENTAS TECNOLÓGICAS PARA LA GESTIÓN DE RIESGO

En el siguiente anexo se presenta la información de las herramientas tecnológicas actualmente utilizadas para la gestión de riesgos (Ver Tabla 14) e igualmente, se muestra una tabla comparativa entre estas herramientas y el prototipo funcional de aplicación web del presente proyecto de grado (Ver Tabla 15).

Tabla 14. Características de las herramientas tecnológicas para la gestión de riesgo

Herramienta	Características						Análisis de la Herramienta *			
	Año de creación de la herramienta	Empresa	País	Tipo licencia	¿Tiene documentación?	Página web	Facilidad de Uso	Intuitiva	Información que debo darle	Información que falta y es relevante
Risk It	2009	ISACA	USA	Gratuita	Sí	(ISACA, 2020)	Media	No	Información detallada de la empresa, los objetivos, los involucrados e igualmente de los riesgos, como el impacto, el esfuerzo, la fiabilidad y la sensibilidad de cada uno	Ya que, es un framework al elaborar la revisión de este, es notorio que la información proporcionada y el proceso que se debe llevar a cabo para la gestión de riesgos es completo.
RiskyProject Professional (Versión 7.1)	2014	Intaver Institute	Canadá & USA	Paga	Sí	(Intaver Institute, 2020)	Baja	No	Información detallada del calendario, riesgos y costos para así crear la matriz de riesgos del proyecto	Falta que sea más automatizada ya que el usuario debe conocer de antemano mucha información que en momentos los pequeños proyectos no lo tienen
Full Monte 2016	2017	Barbecana Inc	No identificado	Paga	Sí	(Barbecana Inc, 2020)	Media	Sí	Calendario del proyecto, con información de fecha de inicio, fin, probabilidad optimista y pesimista, igualmente información de los costos del proyecto	Falta más funciones o enfoques que ayuden no solo a mirar el tiempo que se tiene establecido para cada actividad del proyecto, sino además es importante mostrar los demás riesgos que se pueden presentar durante las diferentes etapas, ya que, también es necesario llevarle un seguimiento a estos
Project Risk Analysis	No identificado	Katmar Software	No identificado	Gratis	No	(Katmar Software, 2018)	Media	No	Costos de cada uno de los aspectos que necesita el proyecto a desarrollar	Falta complementar más el enfoque de la aplicación que no solo sea dirigida a ver los riesgos del costo, sino que se amplíe más a los demás riesgos que pueda presentarse en un proyecto

Tabla 15. Tabla comparativa entre las herramientas tecnológicas y el prototipo funcional

Herramientas tecnológicas	Prototipo funcional de aplicación web
Las herramientas investigadas actúan en todas las etapas de desarrollo de un proyecto de software exceptuando una de ellas, la cual solo actúa en la etapa de pre análisis enfocándose en el riesgo de presupuesto	Actúa en las primeras etapas del proyecto de software, pre análisis y análisis
Para utilizar la mayoría de las herramientas, se necesita tener previos conocimientos acerca del funcionamiento de estas, lo cual, dificulta un poco el uso de ellas	En base a las respuestas dadas por los directores de proyectos que utilizaron el prototipo, se concluye que este es de fácil uso
Algunas de ellas no identifican los riesgos tempranamente debido a que, estas herramientas requieren mucha información la cual no es posible tenerla con precisión antes de iniciar el proyecto. Cabe mencionar que dos de estas herramientas hace detección temprana, una de ellas centrándose solo en el riesgo de presupuesto y la otra en el calendario y presupuesto del proyecto.	El prototipo realiza una detección temprana de los riesgos, ya que, se enfoca solo en los que se pueden presentar en las primeras etapas para así empezar con bases sólidas la realización del proyecto, lo cual genera que los riesgos disminuyan notablemente en las demás etapas.
La mitad de estas son gratuitas y las restantes pagas	Gratuita

El prototipo realizado para este proyecto cuenta con una detección temprana de los riesgos, este va dirigido a los proyectos que están iniciando, es decir, que se encuentran en las primeras etapas de desarrollo, las cuales son pre análisis y análisis. Realizar esta gestión es clave para prevenir que los riesgos evolucionen durante las demás etapas del desarrollo y afecten negativamente el resultado final del proyecto.

Por último, cabe mencionar que existen herramientas para la gestión de proyectos las cuales sirven para ayudar en la disminución de los riesgos que se puedan presentar durante el desarrollo del proyecto, como lo pueden ser los riesgos en el Desacuerdo entre miembros, Monitoreo y control inadecuados del proyecto, entre

otros, los cuales pueden ser mitigados mediante diferentes herramientas, pero en este apartado solo mencionaremos algunas como Jira, TortoiseSVN, Git y Bitbucket.

Jira Software: Esta herramienta tecnológica es de gran ayuda a la hora de comunicarse con el equipo, asignar tareas y supervisa a los mismos, es por tal razón, que cuenta con tres opciones diferentes de plantillas de trabajo, las cuales son: Scrum esta es una buena opción para los equipos que usan metodología ágil que trabajan desde un backlog, planifican y estiman su trabajo en Sprints y entregan trabajo de forma periódica; también está la plantilla de Kanban que también utiliza la metodología ágil pero sin usar Sprints es decir, que supervisan el trabajo en un flujo continuo y por ultimo esta la Supervisión de errores, esta plantilla se utiliza cuando no se necesita el uso del tablero de trabajo sino que se necesita gestionar las tareas de desarrollo y los errores mediante una vista de lista, cabe resaltar que dependiendo de la que escoja la visualización del trabajo será diferente. Por último, la hoja de trabajo de Jira es un tablero el cual muestra el panel de tareas del proyecto, lo cual promueve la comunicación, el desarrollo iterativo y la organización del equipo (ATLASSIAN, 2020).

TortoiseSVN: Es un software que permite el control y revisión de versiones de un proyecto, se basa en Apache ™ Subversion (SVN)®, además es un software libre liberado bajo la licencia GNU GPL y puede usarse con las herramientas de desarrollo que desee y con cualquier tipo de archivo. Entre sus ventajas se encuentra, su fácil uso mediante el explorador de archivos de Windows, ya que, solo muestra los comandos que tienen sentido para el archivo o carpeta seleccionada; otra de sus ventajas es que nos permite visualizar un gráfico donde se puede ver el historial completo de un archivo o carpeta y así ver todas sus versiones, además cuenta con una herramienta personalizada de informe de fallas en cada versión de TortoiseSVN que nos ayuda a corregir los errores mucho más rápido (TortoiseSVN, 2020).

Git: Es una herramienta tecnológica que permite el control de versiones, es de código abierto y sus características permiten que se manejen proyectos de todos los tamaños de una manera eficiente y veloz. Una de las ventajas que tiene Git es la de ser rápido gracias a que la mayoría de sus operaciones se llevan a cabo localmente; otra de sus características es el aseguramiento de datos ya que, el modelo de datos que usa Git garantiza la integridad criptográfica de cada una de las partes del proyecto; igualmente, es un software que permite la creación de ramas para poder experimentar nuevas características en las que se estén trabajando y poder alternar entre ellas sin problemas, y cuando estas características se fusione con su línea principal se pueden eliminar (Git, 2020).

Bitbucket: Esta ofrece a los equipos de trabajo un lugar para planificar proyectos, colaborar en el código, probar e implementar; además es una herramienta del alojamiento de versiones de un proyecto la cual cuenta con tres opciones de alojamiento que son en la nube, servidor y centro de datos. Este software permite la integración de Jira para así mantener los proyectos organizados directamente desde las tarjetas o tareas que se tienen en Jira. Otra característica fundamental de esta herramienta es que nos permite almacenar el código y definir permisos y acceso a los compañeros de equipo (ATLASSIAN Bitbucket, 2020).

Anexo 3. INFORMACIÓN QUE CONTENDRÁ EL PROTOTIPO PARA DETECTAR LOS RIESGOS

En esta sección se presentan las preguntas incluidas en el prototipo web, para así, con estas hallar los riesgos que probablemente se presentarán en las etapas de pre análisis y análisis en los proyectos de software.

Riesgo 1: Falta de suficientes habilidades técnicas en el equipo de trabajo.

Descripción del riesgo: Cuando no se capacita al personal de una manera adecuada o, por el contrario, se eligió un personal que no contaba con las competencias requeridas para el proyecto.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Qué conocimientos tiene el equipo con la tecnología a utilizar en el proyecto? (escala del 1 a 5, donde 1 es novato y 5 experto)
- ¿Qué conocimiento tiene usted como director de proyecto con la tecnología a utilizar en el proyecto? (escala del 1 a 5, donde 1 es novato y 5 experto)

Plan de mitigación:

El equipo de trabajo y el director del proyecto deberían tomar capacitación con respecto a la tecnología con la que se va a trabajar, ya que, no es suficiente tener nociones de la tecnología, sino que se necesita tener un conocimiento pleno de esta para que se pueda hacer entrega del software en el tiempo establecido.

Riesgo 2: Escasez de personal.

Descripción del riesgo: Este riesgo, es dado cuando la estimación del personal es incorrecta, para completar el proyecto en el tiempo definido o si en el ciclo de vida del proyecto existe mucha rotación del personal.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Cuántas personas en total participarán en el proyecto?
- Según su experiencia en la dirección de proyectos ¿Considera que el personal asignado es suficiente para entregar el proyecto en el tiempo y presupuesto establecidos?

Plan de mitigación:

Es necesario que el Gerente del Proyecto puede utilizar herramientas manuales o automatizadas para obtener un presupuesto estimado. Las herramientas de asignación pueden ser hojas de cálculo simples o una herramienta compleja de estimación del presupuesto. Además, tiene que estimar el personal necesario y adecuado para llevar a cabo el proyecto teniendo en cuenta la experiencia y habilidades necesarias para cumplir con las actividades del cronograma del proyecto y de esta manera entregar el proyecto en el tiempo establecido.

Riesgo 3: Conocimiento inadecuado de las herramientas requeridas.

Descripción del riesgo: Cuando no existe una capacitación a los desarrolladores acerca de las herramientas a utilizar.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Si el personal no cuenta con conocimientos plenos de la tecnología y del lenguaje a trabajar en el proyecto, se harán capacitaciones para entender mucho más estas herramientas?

Plan de mitigación:

La empresa debe brindar las capacitaciones que requiera para los proyectos que esté llevando a cabo, ya que, es algo crucial para mejorar los conocimientos y habilidades; esto generará que el personal sea más competentes aumentando la productividad y calidad del trabajo.

Riesgo 4: Falta de acuerdo entre el cliente y el desarrollador.

Descripción: Cuando los requerimientos no son claros desde un inicio y además varían mucho durante el ciclo de vida del software. Lo cual provoca demoras en la entrega.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- Con respecto a los requerimientos del proyecto, ¿Cuál es el nivel de claridad que tiene el cliente frente a esto? (Bajo, Medio, Alto)
- ¿Cuánto conocimiento tiene el cliente acerca de las tecnologías a utilizar en el proyecto? (Nulo, Bajo, Medio, Alto)

Plan de mitigación:

Ya que el cliente no cuenta con una claridad plena de los requerimientos del proyecto y de la tecnología que se va a utilizar. Se considera necesario mantener una comunicación constante con este e Informar al cliente de los próximos pasos y de las tareas pendientes que se van a llevar a cabo, al igual si el Gerente de proyecto considera necesario, sería ideal realizar capacitaciones al cliente acerca de las tecnologías.

Riesgo 5: Presupuesto poco realista.

Descripción del riesgo: Para la realización del producto de software es necesaria una inversión de recursos económicos, tecnológicos y humanos. En momentos estos recursos son escasos, lo cual conlleva retraso en la entrega del producto final.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Dinero destinado para los costos adicionales para los recursos de hardware?
- ¿Cuánto presupuesto fue asignado para la realización del proyecto?
- ¿Cuánto dinero es destinado al pago de personal?
- ¿Qué dinero se tiene destinado para el pago de los gastos de estructura del proyecto, como coste de energía, agua, alquiler del local, ordenadores, etc.?

Plan de mitigación:

Cuando no se cuenta con suficientes recursos económicos para la realización del proyecto, esto conlleva el retraso de la entrega de este. Es por lo cual, que es necesario que el Gerente de proyectos, cuente con herramientas tecnológicas, métricas o hojas de cálculo simples para evaluar el coste del proyecto, ya que, por el momento los cálculos no son suficientes para el pago de personal y demás costos que conlleva el desarrollo del proyecto de software.

Riesgo 6: Calendario poco realista.

Descripción del riesgo: Para lograr el producto final se omiten las actividades que se deben tener en cuenta para la correcta realización del software. Por lo cual, se cuenta con un calendario con plazos muy cortos o, por el contrario, muy extensos para entregar el proyecto.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Cuánto tiempo se tiene programado para la entrega del proyecto según el cliente?

- ¿Cuánto tiempo se tiene estimado para las tareas de análisis del proyecto (Preanálisis, Análisis)?
- ¿Cuánto tiempo se tiene estimado para las tareas técnicas del proyecto (Desarrollo, Pruebas)?

Plan de mitigación:

Para determinar el tiempo de entrega del proyecto es muy importante establecer unos plazos razonables, realistas, medibles y pactados con el equipo de Proyecto y el cliente acerca de las actividades que son necesarias para completar el proyecto, además es necesario tener en cuenta el tiempo que se va a gastar en cada etapa del proyecto.

Riesgo 7: Ambigüedad de requerimientos

Descripción del riesgo: Esto se da principalmente, cuando el cliente no tiene claro lo que quiere.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿El cliente cuenta con disponibilidad de tiempo para las reuniones acerca del proyecto?
- ¿cómo interviene el cliente durante las reuniones del proyecto? (Activamente, Medianamente activo y Inactivo)
- Número de reuniones planificadas con el cliente
- ¿Cada cuánto tiempo se tendrá una reunión con el cliente?

Plan de mitigación:

La asistencia e intervención del cliente durante las reuniones realizadas es de vital importancia, ya que, el producto final debe satisfacer las expectativas o requerimientos que el cliente desea. Por lo cual, si el cliente no interviene activamente o no tiene claro lo que quiere, es importante que el Gerente de

proyecto en el transcurso de las reuniones le realice múltiples preguntas al igual es necesario guiarlo en el tema que se esté llevando a cabo. Finalmente, es importante realizar un número representativo de reuniones durante el desarrollo del proyecto.

Riesgo 8: Falta de informe de los requerimientos

Descripción del riesgo: Este riesgo es dado, por el gerente de proyectos, cuando éste no lleva un registro de los requisitos que quiere el cliente. Lo cual es algo negativo en el momento de desarrollo del software.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Se tiene documentación del cambio continuo de los requerimientos dados por el cliente?
- ¿Qué importancia le da el cliente a la tarea de análisis y extracción de los requerimientos? (Bajo, Medio, Alto)
- ¿Dónde se almacenan la documentación de los requerimientos?

Plan de mitigación:

Llevar un registro de los requerimientos del cliente y del cambio de estos en el transcurso de desarrollo del proyecto, es de vital importancia para contar con un orden a la hora de desarrollar el trabajo e igualmente evitar bucles en las correcciones de errores. Es por tal razón, que es esencial dar importancia a la extracción de los requerimientos del proyecto, asimismo es de gran ayuda almacenarlos en herramientas tecnológicas para un mayor orden y seguridad de la documentación.

Riesgo 9: Monitoreo y control inadecuados del proyecto.

Descripción del riesgo: Al llevar a cabo un proyecto de software es necesario aplicar de manera adecuada estas actividades porque permiten

verificar si el proyecto va marchando según lo planificado. El inadecuado control y monitoreo surge por falta de vigilar el correcto desarrollo de las actividades y tareas establecidas en el proyecto, así como el seguimiento y control de los recursos humanos y materiales que se disponen en el desarrollo del mismo.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Los encargados del proyecto realizan un continuo seguimiento de los requerimientos?
- ¿Se usan herramientas tecnológicas como Git o TortoiseSVN para el control de versiones del código?
- ¿Se usan herramientas tecnológicas de apoyo a la gestión de proyectos como Jira?

Plan de mitigación:

El no realizar un continuo seguimiento de los requerimientos genera un riesgo para el proyecto, por ende, es necesario realizar esto con una periodicidad regular y durante todas las etapas del proyecto, lo cual garantizará que cualquier irregularidad se detecte y corrija a tiempo, lo que permite reducir al mínimo los efectos perjudiciales para el proyecto. Igualmente, es necesario que se lleve a cabo una documentación de cada cambio que se vaya generando durante el proyecto, ya que, esto apoyará la identificación de la solicitud realizada por el cliente. Esta documentación no debe ser ambigua y debe ser validada por las dos partes, el cliente y la empresa de desarrollo de software. Además, al estar documentados y actualizados los requerimientos permite trabajar siempre sobre la última versión de este, mejorando así la entrega final del proyecto. Finalmente, es importante que la empresa emplee herramientas tecnológicas para el control de versiones de código como Git o TortoiseSVN y para la gestión de proyectos como Jira, ya que, son muy

conocidas por su eficiencia y son de las más utilizadas en las diferentes empresas de software.

Riesgo 10: Falta de datos necesarios para realizar un seguimiento objetivo de un proyecto.

Descripción del riesgo: Para realizar correctamente el control o seguimiento de la construcción del software es necesario contar con información previa, como el cronograma con sus actividades a realizar o los cambios en los requerimientos, sino se cuenta con estos datos será difícil realizar una retroalimentación exacta de lo que se lleva en el proyecto.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Los encargados del proyecto proveen el acceso y uso de la información para controlar los cambios que se deben realizar durante el desarrollo?
- ¿Los encargados del proyecto cuentan con procesos para realizar el debido seguimiento, medición, análisis y evaluación del desempeño del proyecto?

Plan de mitigación:

Para realizar correctamente el control o seguimiento del desarrollo del software, es importante que la información acerca de los cambios que se deben realizar en el desarrollo del proyecto sean accesibles para todos los integrantes del equipo de trabajo. E igualmente, es esencial que el Gerente del proyecto y/o líderes de grupo tengan un proceso determinado para evaluar el avance o desempeño en que se encuentra el proyecto en determinado momento.

Riesgo 11: Desacuerdo entre miembros

Descripción del riesgo: Esto se da, cuando la comunicación entre el personal del área de desarrollo no es la óptima y así mismo su sinergia no es la más eficaz para el cumplimiento de objetivos en común.

Para conocer si el proyecto que se llevará a cabo presentará este riesgo se han definido las siguientes preguntas:

- ¿Para este proyecto que recurso humano utilizará?

  Plan de mitigación:

  El desacuerdo entre miembros se produce cuando la sinergia del equipo no es la más eficaz, ya que, hay personas que solo desean trabajar con gente parecida a ellos lo cual genera fuertes barreras para trabajar en conjunto con gente distinta o nueva, de igual manera existen personas que si se adaptan a cualquier ambiente. Es por tal razón, que es importante contratar personas que estén dispuestas a trabajar en equipo y se adapten a cualquier situación.

17. REFERENCIAS

Alharbi, E. T., & Qureshi, M. R. (2014). Implementation of Risk Management with SCRUM to Achieve CMMI Requirements. *Computer Network and Information Security*, 20 - 25.

Arias Chaves, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *Revista de las Sedes Regionales*, 1-13.

ATLASSIAN. (2020). Jira software. Recuperado el 20 de 03 de 2020, de Comenzar a usar Jira: https://www.atlassian.com/es/software/jira/guides/getting-started/basics#step-2-pick-a-template

ATLASSIAN Bitbucket. (2020). *Bitbucket*. Recuperado el 20 de Marzo de 2020, de Una breve descripción de Bitbucket: https://bitbucket.org/product/guides/getting-started/overview

Avdoshin, S. M., & Pesotskaya, E. Y. (23 de 04 de 2012). IEEEXplore. Obtenido de https://ieeexplore-ieee-org.aure.unab.edu.co/document/6188471

Bannerman, P. L. (2008). Risk and risk management in software projects: A reassessment. Journal of Systems and Software, 2118-2133.

Ballefín, S. J. (2012). Métodos de gestión de riesgos en proyectos de software. Recuperado el 30 de 07 de 2019, de https://www.colibri.udelar.edu.uy/jspui/bitstream/20.500.12008/2967/1/tesis-jaureche.pdf

Balestrini, M. (2006). Cómo se elabora el proyecto de investigación. Caracas: BL Consultores Asociados.

Barbecana. (2018). *Full Monte*. Obtenido de https://www.barbecana.com/full-monte/

Barki, H., Rivard, S., & Talbot, J. (2015). An Integrative Contingency Model of Software Project Risk Management. Journal of Management Information Systems, 1-34.

BERNAL, C. A. (2010). *Metodología de la investigación administración, economía, humanidades y ciencias sociales (Tercera edición.* Colombia: PEARSON EDUCACIÓN.

Boehm, B.W. (1991). Software risk management: principles and practices. IEEE Software, 8, 32-41.

Brizuela, E. I., & Blanco, Y. C. (2013). Metodologías para desarrollar Almacén de Datos. Revista de Arquitectura e Ingeniería, 1-13.

Carvalho, M. M., & Junior, R. R. (2015). Impact of risk management on project performance: the importance of soft skills. *International Journal of Production Research*, 1-21.

Chaouch, S., Mejri, A., & Ghannouchi, S. A. (2019). A framework for risk management in Scrum development process. Procedia Computer Science , 1-6 .

Chrissis, M. B., Konrad, M., & Shrum, S. (2009). CMMI® Guía para la integración de procesos y la mejora de productos. Software Engineering Institute, 492 -506.

CMMI. (2019). CMMI Institute. Obtenido de https://cmmiinstitute.com/cmmi/intro

Dhlamini, J., Nhamu, I., & Kachepa, A. (2009). Intelligent Risk Management Tools for Software Development. CiteSeerx, 1-8.

EALDE. (08 de 03 de 2018). Gestión de Riesgos. Obtenido de https://www.ealde.es/sistema-gestion-de-riesgos-iso-31000/

Fedesoft, MinTic, & Sena. (2015). *Caracterización del sector teleinformática, software y TI en Colombia.*

Fernández Sanz, L., & Bernad Silva, P. (03 de 2014). Revista Facultad de Ingeniería Universidad de. Recuperado el 30 de 07 de 2019, de https://www.redalyc.org/pdf/430/43030033021.pdf

Fuentes, M. d. (2011). NOTAS DEL CURSO ANÁLISIS DE REQUERIMIENTOS. México D.F: UNIVERSIDAD AUTÓNOMA METROPOLITANA.

Franco, M. F., & Camacho, H. (2008). El proceso de investigación científica. Maracaibo : EDILUZ.

Galipienso, M. I., Martínez, A. B., Lizán, F. M., & Jover, J. P. (2005). Ingeniería del software. En requerimientos no funcionales (pág. 111). Madrid, España: Pearson Educación S.A.

Git. (2020). *Acerca de*. Recuperado el 20 de Marzo de 2020, de https://git-scm.com/about

Grupo BanColombia. (12 de 07 de 2018). Conoce todo sobre las pymes en Colombia. Obtenido de https://www.grupobancolombia.com/wps/portal/negocios-pymes/actualizate/legal-y-tributario/todo-sobre-las-pymes-en-colombia

Hoodat, H., & Rashidi, H. (2009). Classification and Analysis of Risks in Software Engineering. International Journal of Computer, Electrical, Automation, Control and Information Engineering, 7.

Hu, Y., Zhang, X., E.W.T.Ngai, Cai, R., & Liu, M. (2013). Software project risk analysis using Bayesian networks with causality constraints. Decision Support Systems, 439-449.

IEEE. (2001). Standard for Software Life Cycle Processes—Risk Management (Std 1540-2001). Recuperado de https://ieeexplore-ieee-org.aure.unab.edu.co/document/914365

Intraver Institute. (2019). *RiskyProject: Project Risk Analysis and Project Risk Management Software*. Obtenido de http://intaver.com/

ISACA. (2019). *Risk IT Framework for Management of IT Related Business Risks*. Obtenido de https://www.isaca.org/Knowledge-Center/Risk-IT-IT-Risk-Management/Pages/default.aspx

Islam, S., Mouratidis, H., & Weippl, E. R. (2014). An empirical study on the implementation and evaluation of a goal-driven software development risk management model. *Information and Software Technology*, 117 - 133.

ISO. (2006). ISO/IEC 16085:2006 Systems and software engineering — Life cycle processes — Risk management. Obtenido de https://www.iso.org/standard/40723.html

ISO. (19 de 07 de 2016). ISO Tools. Obtenido de https://www.isotools.org/2016/07/19/iso-31000-gestion-riesgos-cuales-directrices/

ISO. (2008). ISO/IEC 12207 :2008 Systems and software engineering — Software life cycle processes. Obtenido de https://www.iso.org/standard/43447.html

ISO. (2015). ISO / IEC / IEEE 15288: 2015 Ingeniería de sistemas y software: procesos del ciclo de vida del sistema. Obtenido de https://www.iso.org/standard/63711.html

lSOTools. (2019). Normas ISO. Retrieved from https://www.isotools.com.co/normas/

ISOTools. (2019). Sistemas de Gestión de Riesgos y Seguridad. Obtenido de https://www.isotools.org/normas/riesgos-y-seguridad/

Jiménez, M.L. (2015). Gestión de riesgos en proyecto de software a desarrollar en empresa privada. (Tesis de pregrado). Universidad Militar Nueva Granada Bogotá D.C., Colombia

Katmar Software. (2018). *Project Risk Analysis*. Obtenido de https://www.katmarsoftware.com/pra.htm

Khraiwesh, M. (2012). Risk Management Measures In CMMI. International Journal of Software Engineering & Applications (IJSEA), 149 - 163.

Kumar, C., & Yadav, D. K. (2015). A Probabilistic Software Risk Assessment and Estimation Model for Software Projects. Procedia Computer Science, 353 – 361.

López, C., & Salmeron, J. L. (2012). Risks Response Strategies for Supporting Practitioners Decision-Making in Software Projects. ScienceDirect, 1-8.

Masso, J., Pino, F. J., Pardo, C., García, F., & Piattini, M. (2020). Risk management in the software life cycle: A systematic literature review. Computer Standards & Interfaces.

Montoya, L.O. (2015). Estimación de Riesgos en la Etapa de Planeación en Proyectos de Software (Trabajo final de maestría). Universidad Nacional de Colombia, Medellín, Colombia.

Murillo, J. (02 de 03 de 2018). Métodos de investigación de enfoque experimental. Universidad Nacional de Educación Enrique Guzmán y Valle, pág. 5.

Nasir, M. H., Sahibuddin, S., Ahmad, R., & Mohd, S. S. (2015). How the PMBOK Addresses Critical Success Factors for Software Projects: A Multi-round Delphi Study. Journal of Software, 1-18.

Ordoñez, M. P., Ríos, J. R., & Castillo, F. F. (2017). Administración de base de datos con PostgreSQL. Alcoy (Alicante): Área de Innovación y

Desarrollo, S.L. Obtenido de https://books.google.com.co/books?hl=es&lr=&id=5-mkDgAAQBAJ&oi=fnd&pg=PA3&dq=+que+significa+postgresql&ots=DmGPG2VclX&sig=F_IeW1yKBj4XvfPhX1Wla7iXFoc&redir_esc=y#v=onepage&q&f=false

Olechowski, A., Oehmen, J., Seering, W., & Ben-Daya, M. (2016). La profesionalización de la gestión de riesgos: ¿qué papel pueden desempeñar los principios de gestión de riesgos ISO 31000? Revista Internacional de Gestión de Proyectos, 1568-1578.

PMI. (2013). Gestión de los Riesgos del Proyecto. En P. M. Institute, *Guía de los Fundamentos Para La Dirección de Proyectos - Quinta Edición* (págs. 28 -336). Newtown Square, Pensilvania: Project Management Institute, Inc.

Pressman, R. S. (2010). Ingeniería de software un enfoque practico séptima edición. México D.F: Mcgraw-Hill Interamericana editores, S.A. de C.V.

Proença, D., Estevens, J., Vieira, R., & Borbinha, J. (2017). Risk Management: A Maturity Model Based on ISO 31000. IEEE 19th Conference on Business Informatics (CBI) (págs. 1-10). Thessaloniki, Greece: IEEE.

Pressman, R. S. (2010). Ingeniería del software un enfoque practico: Séptima edición. México, D.F: Mc Graw Hill Educación.

Profession, P.´. (2016). *El alto costo de un bajo desempeño ¿Cómo mejorará los resultados de negocios?* Project Management Institute.

Puertas, J. P. (2014). *Creación de un portal con PHP y MySQL. 4ª Edición.* Madrid, España: RA-MA, S.A. Recuperado el 15 de Abril de 2020, de https://books.google.com.co/books?hl=es&lr=&id=RI2fDwAAQBAJ&oi=fnd&pg=PP1&dq=que+es+php&ots=y7lhF8KBS3&sig=N4b5T7upAilOX8nUzb_h_gFzz-4&redir_esc=y#v=onepage&q=que%20es%20php&f=false

Rahmani, H., Sami, A., & Khalili, A. (2016). CIP-UQIM: un modelo unificado para la mejora de la calidad en las PYME de software basadas en CMMI nivel 2 y 3. Tecnología de información y software, 27-57.

RANKOVIĆ, N., & IVANOVIĆ, M. (2018). Risk Analysis Tools for Managing Software Projects. Software Quality Analysis, Monitoring, Improvement, and Applications (págs. 1-10). Novi Sad, Serbia: CEUR Workshop Proceedings.

Real Academia Española. (2019). Riesgo. Obtenido de https://dle.rae.es/srv/fetch?id=WT8tAMI

Report, C. (2018). *Project Resolution Benchmark.* The Standish Group International.

Reyes, F. L. (2015). Gestión de riesgos en proyecto de software a desarrollar en empresa privada (Tesis de especialización). Universidad Militar Nueva Granada, Bogotá D.C., Colombia.

Sangaiah, A. K., Samuel, O. W., XiongLi, Abdel-Basset, M., & Wang, H. (2018). Towards an efficient risk assessment in software projects–Fuzzy reinforcement paradigm. *Computers & Electrical Engineering*, 833 - 846.

Sanz, L. F., & Silva, P. B. (2013, 11 28). *Scielo.* Retrieved from Gestión de riesgos en proyectos de desarrollo de software en España: estudio de la situación: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-62302014000100022

TortoiseSVN. (2020). *About TortoiseSVN*. Recuperado el 20 de Marzo de 2020, de https://tortoisesvn.net/about.html

Universidad de Alicante. (2019). Servicio de informática. Obtenido de https://si.ua.es/es/documentacion/c-sharp/documentos/pruebas/07pruebasunitarias.pdf

Universidad de Antioquia. (2019). *Aprende en Línea plataforma Académica para Pregrado y Posgrado*. Obtenido de http://aprendeenlinea.udea.edu.co/lms/moodle/pluginfile.php/219431/mod_resource/content/2/ingenieriadesoftware/Ingenieria%20de%20software.pdf

Velandia, L. N., & López, W. M. (2015). Escoger una metodología para desarrollar software, difícil decisión. *Revista Educación en Ingeniería*, 12.

Wanderley, M., Jr., J. M., Gusmão, C., & Lima, F. (2015). Proposal of risk management metrics for multiple project software development. Procedia Computer Science, 1-9.