



UNIVERSIDAD DE ESPECIALIDADES ESPÍRITU SANTO

Facultad de Sistemas, Telecomunicaciones y Electrónica

“Desarrollo e Implementación de una aplicación web para la enseñanza de Gestión de Riesgos utilizando el *framework* de *Gamification* como técnica de *e-learning*”

Trabajo de Titulación que se presenta como requisito para el título
de Ingeniero En Sistemas

Autor: María Gabriela Palacios del Campo

Tutor: Mse. Lohana Mariella Lema Moreta

Samborondón, abril de 2016



APROBACIÓN DEL TUTOR

En mi calidad de Tutor de la estudiante María Gabriela Palacios del Campo, que cursa estudios en el programa de TERCER nivel: Ingeniería en Sistemas, dictado en la Facultad de Sistemas, Telecomunicaciones y Electrónica de la UEES, en modalidad presencial.

CERTIFICO

Que he revisado el Trabajo de Titulación denominado: “Desarrollo e implementación de una aplicación web para la enseñanza de gestión de riesgos utilizando el framework de *gamification* como técnica de e-learning”, presentado por la estudiante María Gabriela Palacios del Campo, como requisito previo para optar por el Grado Académico de Ingeniero en Sistemas CERTIFICO que el Trabajo de Titulación ha sido analizado y reúne todos requisitos para ser presentado y sometido a los procesos de revisión estipulados por la Facultad.

Atte.

Lohana Lema Moreta

Reconocimientos

Me gustaría agradecer profundamente a mi tutora, Mse. Lohana Lema. Gracias por el apoyo, tiempo y dedicación. Sin sus conocimientos y ayuda hubiera sido muy difícil implementar la aplicación y desarrollar el tema.

Gracias a mi familia y amigos por ser mi soporte incondicional, sin ustedes no podría estar aquí.

Índice General

CAPÍTULO 1 INTRODUCCIÓN	1
1.1 Antecedentes	3
1.2 Descripción del Problema	5
1.3 Alcance y delimitación del objeto	7
1.4 Objetivos	8
1.4.1 Objetivo General.....	8
1.4.2 Objetivos Específicos.....	8
1.5 Justificación	8
CAPÍTULO 2 MARCO TEÓRICO	11
2.1 Metodologías de Aprendizaje	11
2.1.1 Modelos tradicionales.....	12
2.1.2 Modelos Interactivos.....	14
2.2 E-learning	16
2.2.1 <i>E-learning</i> síncrono.....	20
2.2.2 E-learning asíncrono.....	21
2.2.3 Aprendizaje basado en juegos.....	22
2.2.4 <i>Serious Games</i>	24
2.2.5 <i>Gamification</i>	24
2.3 Ingeniería de Software	28
2.3.1 Gestión de Proyectos.....	29
2.3.2 Gestión de Riesgos.....	29
2.4 Enseñanza de Ingeniería de Software	30
2.5 Metodologías de desarrollo ágil	32
2.5.1 <i>Scrum</i>	32
CAPÍTULO 3 METODOLOGÍA	34
3.1 Etapa 1: Preparación del Producto	35
3.2 Etapa 2: Desarrollo de la Aplicación	36
3.2.1 <i>Sprint</i>	36
3.2.2 Planificación del <i>Sprint</i>	36
3.2.3 Ciclo diario (Reunión diaria).....	37
3.2.4 Revisión del <i>Sprint</i>	37
3.2.5 Retrospectiva del <i>Sprint</i>	37

3.3	Resumen de la metodología	38
CAPÍTULO 4	DESARROLLO DE LA PROPUESTA Y RESULTADO	39
4.1	Preparación del Producto	39
4.1.1	Comparación de Técnicas de <i>E-learning</i>	40
4.1.2	Definición de la Solución	43
4.2	Desarrollo del Producto	51
4.2.1	<i>Sprint 1</i>	52
4.2.2	<i>Sprint 2</i>	62
4.2.3	<i>Sprint 3</i>	71
4.3	Pruebas del Desarrollo	81
4.3.1	Definición de pruebas	81
4.3.2	<i>Sprint 1</i>	82
4.3.3	<i>Sprint 2</i>	83
4.3.4	<i>Sprint 3</i>	84
4.3.5	Resultado de Pruebas.....	86
4.3.6	<i>Sprint 1</i>	86
4.3.7	<i>Sprint 2</i>	88
4.3.8	<i>Sprint 3</i>	90
4.4	Resumen de Pruebas	91
CAPÍTULO 5	CONCLUSIONES Y RECOMENDACIONES	92
CAPÍTULO 6	REFERENCIAS	96
CAPÍTULO 7	ANEXOS	103
7.1	Anexo 1: Diagrama de Entidad Relación	103

Índice de Tablas

<i>Tabla 2.1.1 Modelos de Aprendizaje</i>	12
<i>Tabla 2.1.2. Técnicas de Enseñanza de Modelos Tradicionales</i>	14
<i>Tabla 2.2.1. Conceptos relacionados a e-learning</i>	17
<i>Tabla 2.2.2 Beneficios de E-learning</i>	18
<i>Tabla 2.2.3 Herramientas de e-learning síncrono</i>	20
<i>Tabla 2.2.4 Técnica de E-learning asíncrono</i>	22
<i>Tabla 2.2.5 Herramientas de Aprendizaje basado en juegos</i>	23
<i>Tabla 2.2.6 Elementos de diseño de Gamification</i>	26
<i>Tabla 2.2.7 Framework 6d Gamification</i>	27
<i>Tabla 2.2.8 Mecánica y Dinámica de Gamification</i>	28
<i>Tabla 2.3.1 Procesos de la gestión de Riesgos</i>	30
<i>Tabla 3.1.1 Actividades de Preparación del Producto</i>	35
<i>Tabla 3.2.1 Sprints esperados en el desarrollo</i>	36
<i>Tabla 4.1.1 Comparación de técnicas de aprendizaje basado en juegos</i>	41
<i>Tabla 4.1.2 Árbol de decisión de técnicas de aprendizaje basadas en juegos</i>	42
<i>Tabla 4.1.3 Características Iniciales de la Aplicación</i>	44
<i>Tabla 4.1.4 Caso de Estudio de la Aplicación</i>	45
<i>Tabla 4.1.5 Riesgos del Caso de Estudio</i>	46
<i>Tabla 4.1.6 Valoración de Riesgos</i>	47
<i>Tabla 4.1.7 Valoración de Riesgos del Caso</i>	48
<i>Tabla 4.1.8 Mecanismos de Mitigación de Riesgos</i>	49
<i>Tabla 4.1.9 Historias de Usuario</i>	50
<i>Tabla 4.2.1 Lista de Tareas Sprint 1</i>	52
<i>Tabla 4.2.2 Planificación del Trabajo Sprint 1</i>	53
<i>Tabla 4.2.3 Tareas realizadas en el sprint 1</i>	59
<i>Tabla 4.2.4 Elementos de Gamification implementados</i>	59
<i>Tabla 4.2.5 Resumen de Horas de Desarrollo Sprint 1</i>	60
<i>Tabla 4.2.6 Lista de Tareas Sprint 2</i>	62
<i>Tabla 4.2.7 Planificación del Trabajo Sprint 2</i>	63
<i>Tabla 4.2.8 Lista de Tareas Realizadas Sprint 2</i>	68
<i>Tabla 4.2.9 Elementos de Gamification implementados</i>	68
<i>Tabla 4.2.10 Resumen de Horas de Desarrollo Sprint 2</i>	69
<i>Tabla 4.2.11 Lista de Tareas Sprint 3</i>	71
<i>Tabla 4.2.12 Planificación del Trabajo Sprint 3</i>	72
<i>Tabla 4.2.13 Lista de Tareas Realizadas Sprint 3</i>	77
<i>Tabla 4.2.14 Elementos de Gamification implementados</i>	78
<i>Tabla 4.2.15 Resumen de Horas de Desarrollo Sprint 3</i>	79
<i>Tabla 4.3.1 Características de Usuarios de Pruebas</i>	81

Índice de Figuras

Figura 2.1. Enseñanza de una vía.....	13
Figura 2.2 Relación de la Tecnología de Aprendizaje	16
Figura 2.3 Características de técnicas de E-learning	19
Figura 2.4 Beneficios del Aprendizaje basado en juegos.....	23
Figura 2.5 Dimensiones de Gestión de Proyectos	29
Figura 4.1 Técnicas de aprendizaje basado en juegos	40
Figura 4.2 Diagrama de Arquitectura de Aplicación Web.....	51
Figura 4.3 Selección de casos de estudio de tabla casos.....	55
Figura 4.4 Tabla de Riesgos en la Base de Datos.....	56
Figura 4.5 Tabla de Usuario en la Base de Datos.....	56
Figura 4.6 Pantalla de Inicio de Sesión	57
Figura 4.7 Pantalla de Introducción	58
Figura 4.8 Pantalla de Inicio de Sesión con error.....	58
Figura 4.9 Comparación de Listado de Tareas Pendientes Sprint 1	61
Figura 4.10 Comparación de horas acumuladas Sprint 1	61
Figura 4.11 Pantalla de Inicio de Resolución de Caso	64
Figura 4.12 Pantalla Categorización de Riesgos	65
Figura 4.13 Captura de pantalla al arrastrar valor de probabilidad	66
Figura 4.14 Captura de pantalla al soltar valor de probabilidad.....	66
Figura 4.15 Captura de pantalla al arrastrar el valor del impacto.....	66
Figura 4.16 Captura de pantalla al soltar valor de probabilidad.....	66
Figura 4.17 Captura de Pantalla de Respuestas Nivel 1.....	67
Figura 4.18 Comparación Listado Tareas Pendientes Sprint 2	70
Figura 4.19 Comparación de horas acumuladas Sprint 2	70
Figura 4.20 Pantalla de Mitigación de Riesgos	73
Figura 4.21 Función arrastrar riesgo en Pantalla de Mitigación.....	74
Figura 4.22 Mitigación de Riesgo, selección de porcentaje.....	74
Figura 4.23 Mitigación de Presupuesto Barra de Control.....	75
Figura 4.24 Barra de Control de Presupuesto de Riesgos.....	75
Figura 4.25 Resultados del Nivel 2	76
Figura 4.26 Resultados de Resolución de Caso	76
Figura 4.27 Comparación Listado de Tareas Pendientes Sprint 3.....	80
Figura 4.28 Comparación de horas acumuladas Sprint 3	80
Figura 4.32 Escenario de prueba 1 Requerimiento 1 Sprint 3.....	85
Figura 4.32 Escenario de prueba Requerimiento 3 Sprint 3.....	86
Figura 4.33 Pruebas 1 Requerimiento 6 Sprint 1	87
Figura 4.34 Pruebas 2 Requerimiento 6 Sprint 1	87
Figura 4.35 Prueba 1 Requerimiento 2 y 3 Sprint 2.....	88
Figura 4.36 Prueba 2 Requerimiento 2 y 3 Sprint 2.....	89
Figura 4.37 Prueba 1 Requerimiento 1 Sprint 3.....	90
Figura 4.38 Prueba 1 Requerimiento 3 Sprint 3.....	91

RESUMEN

La evolución tecnológica en la última década se ha visto marcada por el desarrollo de todo tipo de herramientas en una amplia diversidad de ámbitos; en el contexto del aprendizaje y su convergencia con la tecnología, el término de *e-learning* se ha posicionado estratégicamente; *Gamification*, es una de las herramientas que lo ha hecho posible. Con *Gamification*, a través del uso de elementos de juego, se motiva al usuario a aprender y crear oportunidades para explorar temas que obligatoriamente deben ser abordados desde una perspectiva teórica, por ejemplo, las teorías relacionadas a la Ingeniería de *Software*. En consecuencia, es una herramienta muy útil para enfrentar los desafíos actuales de la enseñanza de Gestión de Riesgos en administración de proyectos tecnológicos.

El presente trabajo de titulación, describe el análisis e implementación de una aplicación web que utiliza el marco de trabajo de *Gamification* que fortalezca la enseñanza de las etapas de identificación, análisis y mitigación de la Gestión de Riesgos. Para lograrlo, se utilizó una metodología con enfoque cualitativo de alcance exploratorio, además de un diseño investigativo conformado por dos fases, Preparación del Producto y Desarrollo del Producto; cuyas actividades estuvieron marcadas por la implementación de métodos ágiles y de metaanálisis.

Los hallazgos de este trabajo de titulación evidencian, por medio del uso del árbol de decisión de técnicas de aprendizaje basadas en juegos porqué *Gamification* es el marco de trabajo adecuado para la problemática planteada. Adicional al aporte teórico, cómo resultado se cuenta con la implementación una aplicación web construida de acuerdo al *Gamification Framework 6D* que puede ser usada tanto para enseñar gestión de riesgos, y en los posterior, para la experimentación de la efectividad de *Gamification* en contraste con la metodología de enseñanza tradicional.

CAPÍTULO 1 INTRODUCCIÓN

Como resultado de los avances en la tecnología y las desventajas de la enseñanza tradicional, las herramientas de aprendizaje han evolucionado. Actualmente, existen técnicas que utilizan la tecnología para fortalecer la enseñanza y se las conoce como *e-learning*. *Gamification* es una técnica de *e-learning* que, al implementar elementos de juego, captura la atención del usuario en contextos donde normalmente, la atención del mismo sería mínima. Tomando en cuenta esta información, la Gestión de Riesgos en proyectos de *software*, un tópico de enseñanza eminentemente teórico, enfrenta muchos desafíos. Para corroborar esta premisa, se evidenciará que existe una brecha investigativa de publicaciones que conjuguen a cabalidad estas dos áreas de estudio; temas como la gestión de proyectos son abordados parcialmente en un contexto gamificado, pero no representan una vasta evidencia para determinar que este ámbito se encuentra ampliamente estudiado. Por esta razón, en este trabajo de titulación, se analiza sistemáticamente el contexto planteado como base para implementar una aplicación web que, usando un marco de trabajo de *Gamification*, provea una herramienta para la enseñanza de Gestión de Riesgos en proyectos de *software* y además permita planificar la ejecución trabajos futuros que disminuyan la brecha investigativa mencionada previamente.

Para lograrlo, se ha implementado una metodología de enfoque cualitativo y alcance exploratorio, compuesta de dos etapas. La Etapa 1 denominada Preparación del Producto, donde se realiza la revisión sistemática y definición de la solución que se va a implementar; esta etapa tiene la finalidad de 1) conocer lo que ya se ha publicado y que intentos

existieron para resolver la problemática planteada; 2) las técnicas aceptadas y enfoques que podrían contribuir a la solución del objetivo; 3) las cuestiones que no han sido resueltas por la academia, la industria u otros esfuerzos de investigación. El análisis de la información recabada mediante la comparación de características específicas, llevó a justificar el uso de *Gamification* como *framework* de la aplicación. A partir de la información obtenida, se definen las características principales de la aplicación.

Luego, en la Etapa 2, Desarrollo del Producto, se utiliza *Scrum* para dar flexibilidad a la gestión e implementación de los requerimientos y alcance de la aplicación. Por medio del marco, se implementan *sprints* que entregan una versión de producto probada, es decir completamente funcional. Al finalizar los *sprints*, se entrega una versión final de la aplicación web.

Este trabajo de titulación, está estructurado en 5 capítulos. En el capítulo 1, denominado Introducción, presenta en los antecedentes y el contexto en el que se implementó la aplicación. Luego, en la descripción del problema se aborda la problemática a resolver; seguida del alcance y delimitación de la solución que se desarrolla. Enseguida, la justificación describe porqué es importante resolver este problema y finalmente se presentan los objetivos que definen este trabajo.

A continuación, sigue el Capítulo 2 llamado Marco Teórico. En él, se presenta una revisión literaria de todos los conceptos relacionados a la investigación. Se presentan los distintos modelos de aprendizaje, las técnicas y conceptos más importantes de *e-learning*, características de *Gamification* y la información relacionada a Gestión de Riesgos en proyectos de Software.

La metodología utilizada para alcanzar los objetivos definidos en el Capítulo 1, es descrita en el capítulo 3. En ella, se definen claramente el enfoque, diseño y alcance de la investigación además de detallar las actividades que conforman cada una de las fases del diseño.

En el capítulo 4, Desarrollo de la propuesta y resultados, se realiza el diseño, desarrollo y construcción de la aplicación siguiendo el marco propuesto en la metodología. Luego de cada *sprint*, se presenta una versión del producto que se evalúa considerando los elementos de diseño de *Gamification*.

Después de haber descrito el desarrollo del producto sigue la etapa de Pruebas, donde se presentan las pruebas de funcionalidad realizadas a la aplicación. Para las pruebas se diseñaron casos de estudio para comprobar el funcionamiento.

Finalmente, las Conclusiones son presentadas en el Capítulo 5, se realiza un resumen de las contribuciones más importantes del trabajo. También se describen las limitaciones presentes y las recomendaciones para trabajos futuros.

1.1 Antecedentes

A partir de los avances en las tecnologías de la información, la educación y su forma de impartirla ha cambiado. A pesar de los intentos de los maestros por ser innovadores en sus métodos de enseñanza, las metodologías tradicionales son constantemente percibidas por alumnos como inefectivas y aburridas (Lee & Hammer, 2011). Además, de acuerdo con el modelo del cono de la experiencia, propuesto por Dale, E. (1969) sólo el 10% de lo que se lee y el 20% de lo que se escucha es retenido por estudiantes. En la misma teoría se establece que los estudiantes recuerdan el 50% de lo que se observan y escuchan y el 90% de las acciones que ellos realizan, así sea por medio de una simulación. Estos datos reflejan que los métodos tradicionales de enseñanza como: dar sólo charlas o exposiciones no son tan efectivas como técnicas de aprendizaje.

En respuesta a esto y considerando la evolución de los medios tecnológicos, se han implementado técnicas bajo el nombre de *e-learning*, las cuales aprovechan los beneficios que ofrece la tecnología dentro del ámbito de la enseñanza; con el objetivo de motivar al estudiante y establecer metas dentro del proceso de aprendizaje. *E-learning* se puede definir de dos formas. Primero, como tecnología que apoya el aprendizaje (Comission of the European Communities, 2000) y segundo, como el uso de medios tecnológicos para propósitos de aprendizaje que van desde el soporte en salones de clases hasta la implementación de cursos no presenciales (Guri-Rosenblit, 2005).

De acuerdo con Dicheva, Dichev, Agre & Angelova (2015) una de las técnicas más utilizadas dentro de *e-learning* son las aplicaciones que implementan elementos de juegos. A través de ellos, los estudiantes

desarrollan sus capacidades para resolver problemas, trabajar en equipo y comunicarse. Gamification es una técnica que implementa este concepto bajo la premisa de que el entorno de aplicación con el que se asocia, no es un juego (Deterding, Sicart, Nacke, O'Hara, & Dixon, 2011) . Al usar Gamification, se obtienen las ventajas de los juegos al entretener al usuario, pero, además, se provee toda la información relevante sobre un tema específico (Gee, 2003).

Como técnica de enseñanza, Gamification ha sido principalmente implementado en las siguientes áreas de aprendizaje: ciencias de la computación, matemáticas, tecnologías de la información, programación de juegos e ingenierías (Dicheva, Dichev, Agre, & Angelova, 2015). Un ejemplo de Gamification en la educación es Fantasy Geopolitics, en esta aplicación se hace uso de los elementos de juego para motivar a estudiantes de secundaria a estar enterados de noticias internacionales (Smith, 2015). En esta herramienta, los alumnos seleccionan un grupo de países y a través de un sistema de competencias van ganando puntos siempre y cuando, se mantengan informados sobre los eventos en dichos países. En el ámbito empresarial, la compañía de consultorías Deloitte, implementó un sistema denominado Deloitte Leadership Academy (DLA); cuyo propósito es ofrecer a cerca de 20,000 empleados y clientes de Deloitte un programa de entrenamiento en línea sobre temas prácticos relacionados con consultoría (Farrall, 2015). En el sistema se ofrece contenido de conocidas escuelas de negocio como: Harvard, Melbourne y Stanford por medio de videos, pero para hacer que la experiencia sea interactiva, se hace uso de elementos como: misiones, desafíos, recompensas, tablas de posiciones y exámenes. De esta forma, se captura la atención del usuario y se los motiva a continuar con el curso (Meister, 2013). Una evaluación realizada por los creadores de DLA, afirma que los usuarios que acceden al sistema más de una vez se incrementó en un 37% en comparación con su sistema anterior y, además, aumentó el número de usuarios que terminan el ciclo del programa (Bohyun, 2015).

Con el propósito de comprobar los beneficios de implementar Gamification en procesos educativos, se han realizado varios experimentos. Por ejemplo, en el 2013, se desarrolló un sistema para un curso de producción de contenido multimedia. En él, se implementaron elementos de juego como: puntos, niveles, tablas de posiciones y premios para encapsular los temas del curso en logros y desafíos significativos (Barata, Gama, & Gonçalves, 2013). En el experimento se compararon los datos del rendimiento y desempeño de los estudiantes que participaron en el curso gamificado con los estudiantes que fueron parte del curso en años anteriores. Finalmente, los resultados demostraron que los estudiantes que participaron del curso gamificado eran más proactivos y participativos. (Barata G. , Gama, Fonseca, & Gonçalves, 2013). Por otro lado, Browne y Anand (2013) desarrollaron una aplicación móvil para un curso introductorio a ciencias de la computación. El propósito del curso era enseñar los conceptos básicos y desarrollar las capacidades para solucionar problemas relacionados a computación. El objetivo del experimento era analizar si al incorporar elementos de juego en el curso, los estudiantes iban a mostrar mayor interés y satisfacción en comparación con participar del curso que utilizaba metodologías tradicionales. Al finalizar el estudio, los datos basados en el uso de la aplicación y en la comprensión de los conceptos enseñados, demostraron que los estudiantes preferían el uso de la aplicación en comparación con la enseñanza tradicional.

1.2 Descripción del Problema

Una y otra vez, las metodologías tradicionales de aprendizaje han sido descritas como obsoletas. Son caracterizadas por “ofrecer resistencia a los cambios y sitúan al maestro como centro del proceso de enseñanza” (Benitez Murube, 2009). Por esta razón, el profesor es el que provee la información y el estudiante es un agente pasivo. De acuerdo con Vargas (2009), cuando se utiliza este modelo tradicional se deja a un lado el fenómeno de la comprensión y el “proceso de relación con sentido de los contenidos”. En consecuencia, es usual que los estudiantes terminen

estresados, abrumados por la cantidad de información que han recibido y que no han tenido tiempo de reflexionar (Gómez López, 2002). De la misma manera, el modelo tradicional presenta “dificultad para favorecer una asimilación inmediata de los contenidos y limitación del pensamiento crítico y creativo del alumno” (Beard & Hartley, 1984).

Somerville (2005) describe la gestión de proyectos como el proceso donde se realiza la administración, planificación y temporalización de los proyectos de software. Si la gestión de proyectos es exitosa, es porque el proyecto cumple con todos los requisitos y estándares de calidad. La gestión de proyectos incluye varias dimensiones, una de estas, la gestión de riesgos. En esta dimensión se realiza la identificación, análisis y respuesta a riesgos de un proyecto (PMI, 2012). La gestión adecuada del riesgo determinará el éxito o fracaso del proyecto y en la actualidad, esta es un tema de aprendizaje que enfrenta serios desafíos.

Gao & Rusu (2015) establecen que uno de los problemas más grandes de los estudiantes que forman parte de cursos de gestión de proyectos es que ellos, tienen poca experiencia en los mismos. De la misma manera, tienen conceptos y habilidades limitadas sobre gestión de proyectos y poca experiencia de trabajo en equipo. En consecuencia, al tener conocimientos limitados en conceptos de gestión, es difícil comprender la importancia de procesos como identificar y monitorizar riesgos. Esta falta de conocimiento y experiencia en el área, produce un desafío al momento de transferir conocimientos.

Tomando en cuenta lo mencionado sobre la enseñanza de gestión de proyectos, de acuerdo con la teoría del aprendizaje basado en experiencias de David Kolb (1984), un método para que la transferencia de aprendizaje sea efectiva, es que los estudiantes ejerciten y trabajen con los

conceptos obtenidos en clases. La información recibida en el aula se convertirá en conocimiento para cada estudiante cuando se reflexione sobre la misma o se realice algún tipo de experimentación.

En el ámbito de la experimentación e investigación, existen artículos que están relacionados a la implementación parcial de Gamification como método para mejorar la enseñanza de ingeniería de software (Xie, Tillman, & De Halleux, 2013). En su mayoría, estos artículos están orientados a la problemática de la Gestión de Proyectos, sin embargo, ninguno de ellos está enfocado en Gestión de Riesgos (Berkling & Thomas, 2013). A pesar de esto, existen un artículo que sugiere que implementar elementos de juego podría beneficiar en la enseñanza de Gestión de Riesgos (Taran, 2007).

Considerando los avances en las metodologías de enseñanza, la implementación de *e-learning*, los desafíos en el aprendizaje de gestión de proyectos, específicamente gestión de riesgos y la falta de investigación sobre la relación y beneficios de Gamification en la enseñanza de gestión de riesgos; se propone que por medio del desarrollo e implementación de una aplicación que utilice los elementos de *Gamification* se provea de una herramienta que permita planificar la ejecución trabajos futuros que disminuyan la brecha investigativa de las dos áreas y sea un instrumento para experimentar formalmente la premisa de que el uso de la aplicación fortalece la enseñanza de gestión de riesgos en un ambiente controlado.

1.3 Alcance y delimitación del objeto

El presente trabajo de titulación está limitado al análisis, desarrollo, implementación y pruebas de una aplicación web para la enseñanza de gestión de riesgos, específicamente en las etapas de identificación, análisis y mitigación de riesgos.

1.4 Objetivos

1.4.1 Objetivo General

Implementar una aplicación web aplicando la técnica de Gamification para fortalecer la enseñanza de gestión de riesgos en las etapas de identificación, análisis y mitigación de riesgos.

1.4.2 Objetivos Específicos

1. Analizar las metodologías de aprendizaje y las técnicas de *e-learning* para justificar la elección de la herramienta a usar en el desarrollo de la aplicación web.
2. Analizar las funcionalidades que se deben considerar para el diseño de la herramienta web.
3. Diseñar una herramienta web que implemente los elementos de juego de Gamification para que sirva para enseñanza de gestión de riesgos.
4. Desarrollar la aplicación web utilizando una metodología de gestión de proyectos ágil.
5. Realizar pruebas de usuario considerando las funcionalidades definidas en el objetivo específico 1.

1.5 Justificación

De acuerdo con una investigación realizada por Georgetown University (2014), para el año 2018 el 51% de las plazas de trabajos relacionadas a ciencia, tecnología, ingeniería y matemática (STEM) van a pertenecer a trabajos en ciencias de la computación (Carneval, Smith, & Melton, 2014). De la misma manera, el Departamento de Comercio de Estados Unidos, identificó que entre las carreras que tienen proyectado mayor crecimiento entre los años 2014 y 2024 están: Web Developers (188%) y Analista de operaciones (30.2%) (Bureau of Labor Statistics, 2015). Además, los trabajos relacionados a las ciencias computacionales están entre los 10 mejores pagados del año 2015 (CareerCast, 2016). Por ejemplo: ingenieros biomédicos (\$89,165 mensual), científico de datos

(\$124,149), ingeniero de software (\$93,113) y analista de sistemas computacionales (\$81,150). Sin embargo, a pesar de estos avances en el ámbito laboral y el crecimiento de las ciencias computacionales, Markov (2014) asegura que los estudiantes que terminan esta carrera no están lo suficientemente preparados para la vida laboral y tienen dificultades en entender conceptos básicos sobre esta ciencia.

Por otro lado, PMI Institute (2008) establece que, 1/5 del PIB mundial está siendo destinado a proyectos. Por esta razón, se ha dado paso de métodos tradicionales de gestión de empresas a técnicas basadas en gestión de proyectos. El reporte del caos 2015 menciona que, de 50,000 proyectos tecnológicos en el mundo, el 29% ha sido considerado exitoso, 52% de los proyectos fueron desafiantes y 19% fracasaron completamente (Standish Group, 2015). Los factores de éxito de dichos proyectos incluyen: apoyo de los ejecutivos, usuarios involucrados, objetivos de negocio claros, recursos especializados, entre otros. Estos factores si no son gestionados apropiadamente se convierten en un riesgo para el éxito del proyecto. Es parte de la gestión de proyectos aprender a administrar los riesgos para que estos sean oportunidades y agreguen valor al producto/servicio final.

La APM (2014), es decir Asociación de Gestión de Proyectos, describe Gamification como una tendencia que ofrece varios beneficios a los negocios. De acuerdo con Gartener Inc., para el 2015 el 25% de los procesos de negocios van a ser rediseñados para implementar Gamification, generando alrededor de \$2.8 billones en ganancia (Meister, 2013). En la actualidad, compañías como: Nike y Starbucks ya han implementado elementos de juego para mejorar el negocio e incrementar la interacción con el consumidor. En el 2009, Starbucks desarrolló un sistema de recompensas en una aplicación móvil denominado “*My Starbucks Reward Program*”. En la aplicación, los clientes acumulaban puntos cada vez que compraban un producto de Starbucks. Dependiendo de los puntos, los usuarios eran clasificados y los clientes más destacados

recibían incentivos como: promociones y tarjetas de regalos. Hasta el 2012, los usuarios de la aplicación eran cerca de 4.5 millones de personas que alcanzaron alrededor de 3 billones en compras de tarjetas (Hector, 2015). Por otro lado, Nike, desarrolló un programa llamado Nike+. En él, los usuarios comparten y siguen rutinas de ejercicio, ganando puntos por su participación. Entre los resultados más destacados de la implementación están que Nike aumentó en un 14% su participación en el mercado de zapatos de carrera. También, los miembros de la comunidad aumentaron de 500,000 en el 2007 a 11 millones en el 2013 (Badgeville, 2013).

En el ámbito educativo hay poca información y resultados sobre los beneficios de *Gamification*. Es por eso, que, en este trabajo de investigación, se busca proveer una herramienta que sea una alternativa en el modelo de aprendizaje y tal como lo propone Kolb (1984) sirva para aprehender los conceptos, en este caso de gestión de riesgos, y convertir la información en conocimiento. Además, a través del uso de la aplicación web, los estudiantes podrán simular en un ambiente gamificado definiciones que en un inicio son percibidas como ambiguas. Por otra parte, el producto que se obtiene como resultado de este estudio sirve como herramienta para futuras pruebas y experimentos relacionados a la efectividad de la técnica de *Gamification* como parte de *e-learning*.

CAPÍTULO 2 MARCO TEÓRICO

En esta sección se describe los conceptos y teorías necesarias para el análisis de las metodologías de aprendizaje, técnicas de *e-learning*, enseñanza de gestión de riesgos y metodologías de desarrollo ágil. En el apartado 2.1 se describen las metodologías tradicionales e interactivas de enseñanza. Luego, se presentan las características de las técnicas de *e-learning* (Sección 2.2) con sus beneficios y clasificación. Dentro de esta sección se muestran las diferentes alternativas de aprendizaje basado en juegos como lo son *Serious Games* y *Gamification*.

A continuación, se describen los elementos de la ingeniería de software (Sección 2.3) particularmente de la Gestión de Riesgos en la Gestión de Proyectos. En seguida, se detalla la enseñanza actual de Ingeniería de Software (Apartado 2.4). Finalmente, el capítulo termina con la descripción de las metodologías ágiles de desarrollo de software (Sección 2.5), especialmente Scrum.

2.1 Metodologías de Aprendizaje

“La palabra método viene del latín *methodus*, que, a su vez, tiene su origen en el griego, en las palabras *meta* = meta y *hodós*= camino, Por consiguiente, método quiere decir camino para llegar a un lugar determinado” (Torres Maldonado, 2009). En este caso, es el camino para llegar al conocimiento. Las metodologías de aprendizaje o técnicas de enseñanza hacen referencia a los procesos para guiar al alumno a alcanzar ciertas metas (Parras, 2003). Estas suponen una forma específica para transmitir conocimientos y lograr los objetivos de aprendizaje (Hernández, 2007).

Hay varios aspectos que se deben considerar cuando se elige un modelo de aprendizaje. Se toman en cuenta: la experiencia del docente, los conceptos que él tiene sobre la materia y la relación entre la metodología y los objetivos de aprendizaje (Hernández, 2007). Dependiendo de estos factores, el proceso de aprendizaje va a adoptar un modelo donde el estudiante cumple un rol activo o pasivo en el aula de clases. En la Tabla 2.1.1, se pueden observar los distintos modelos de aprendizaje.

Tabla 2.1.1
Modelos de Aprendizaje

Pasivo	Activo
<ul style="list-style-type: none"> • Método Expositivo Conclusivo • Método Expositivo Suscitado 	<p>Métodos interactivos Reproductivos:</p> <ul style="list-style-type: none"> • Evaluativo • Clarificador • Tutorial <p>Métodos interactivos Productivos:</p> <ul style="list-style-type: none"> • Deductivo • Inductivo (mapas conceptuales) • Horizontal (debate)

Fuente: Elaboración propia

Considerando la Tabla 2.1.1, los modelos donde el alumno adopta un rol pasivo también son conocidos como modelos tradicionales. Por otro lado, cuando el alumno tiene un rol más activo se utilizan los métodos interactivos y didácticos.

2.1.1 Modelos tradicionales

De acuerdo con Parra (2003), los modelos tradicionales son aquellos que adoptan procesos repetitivos para la transmisión de conocimientos desde los libros y la experiencia del docente a los alumnos. En este modelo, el profesor explica y va avanzando de forma progresiva los temas del curso. En otras palabras, los profesores sirven como fuente de conocimiento y los estudiantes como receptores pasivos del mismo (Kuzu, 2007). Según Vargas (2009), “en esta perspectiva el aprendizaje es la comunicación entre

emisor(maestro) y receptor(alumno) y se ignora el fenómeno de la comprensión y el proceso de la relación con sentido de contenidos”. Esto se lo puede visualizar en la Figura 2.1.



Figura 2.1. Enseñanza de una vía

Fuente: Elaboración propia

Damodharan, V. & Rengarajan.V (2007) mencionan que, cuando se implementa únicamente este modelo, la concentración del estudiante disminuye luego de tan solo 15 minutos de clases. Además, debido a que la comunicación en este ambiente es de una sola vía, el profesor no recibe retroalimentación de parte del estudiante, lo que hace que la interacción con el estudiante sea mínima. Otra limitante de este modelo, es que los conocimientos impartidos dependen únicamente de los conocimientos del docente, lo que genera una enseñanza incompleta y parcializada a la opinión del docente.

A pesar de las limitantes antes mencionadas, la enseñanza con modelo tradicional tiene ciertos beneficios. Al utilizar este método en aulas con gran cantidad de alumnos, el profesor puede cubrir los aspectos más relevantes de un tema (Hernández, 2007). De igual manera, se requiere de menos tiempo para transmitir la materia y preparar la misma.

Por otro lado, entre las técnicas más comunes del modelo tradicional están las siguientes: clases magistrales, tutorías y trabajos guiados (Carpio,

2008). En la *Tabla 2.1.2* , se exponen las características más relevantes de cada tema.

Tabla 2.1.2.
Técnicas de Enseñanza de Modelos Tradicionales

Técnica de Enseñanza	Características
<i>Clase magistral</i>	<ul style="list-style-type: none"> • Favorecen aprendizajes superficiales y promueven la memorización. • Contenidos claro y sistemático • Toma en cuenta conocimientos previos. • Se realizan resúmenes de las partes más relevantes de la clase.
<i>Tutorías</i>	<ul style="list-style-type: none"> • Atención personalizada del profesor. • Es un soporte para la clase magistral • Puede ser individual o en grupos pequeños. • El docente realiza preguntas críticas para fomentar el pensamiento crítico en los estudiantes.
<i>Trabajos guiados</i>	<ul style="list-style-type: none"> • El profesor envía trabajos específicos para fortalecer ciertos conocimientos en sus estudiantes.

Fuente: Elaboración propia

2.1.2 Modelos Interactivos

El modelo interactivo surge como rechazo a las metodologías de enseñanza tradicional que apoyan los procesos memoristas y mecánicos. Propone una relación de dos vías entre el estudiante y el profesor. En este ambiente, el aprendizaje está centrado en el estudiante (Boumová, 2008) y el profesor, es el encargado de proveer las herramientas necesarias para que el alumno pueda emprender el proceso de aprendizaje (Scrivener, 2005). Por otro lado, este modelo describe que “el conocimiento crítico se construye, se elabora, por medio de una serie de procesos intelectuales y motrices que implican realizar asociaciones, relaciones, abstracciones,

fórmulas, conclusiones, análisis o síntesis, de forma activa y consciente” (Torres Maldonado, 2009).

Aparici, R. y Silva, M. (2012) mencionan que entre las características más comunes de este modelo están las siguientes. Primero, el estudiante trabaja con el conocimiento que recibe. Segundo, el estudiante se convierte en un agente activo en el proceso de aprendizaje. Tercero, cada estudiante tiene un experiencia única y particular con el conocimiento que está recibiendo. Por último, este modelo busca retroalimentación para el alumno en los procesos de enseñanza.

Por medio de este modelo se pueden implementar técnicas donde se promueven los debates entre alumnos que son guiados por el profesor. Otra característica de esa metodología, es que se valora la experiencia de los alumnos y su aporte al aula (Hernández, 2007). De la misma forma, en el modelo es común utilizar herramientas de la tecnología de información para motivar y facilitar el proceso de enseñanza; ya que es por medio de la misma que se puede desarrollar en los estudiantes capacidades específicas relacionadas al tema que se enseña (Boumová, 2008). Este proceso de involucrar las tecnologías de la información y la comunicación (TICs) en procesos de enseñanza se lo conoce como *E-learning*. En la Figura 2.2, que se muestra a continuación, se ilustra como la tecnología puede interactuar en el proceso de aprendizaje.



Figura 2.2 Relación de la Tecnología de Aprendizaje

Fuente: Elaboración propia

2.2 E-learning

E-learning es un término que surgió a partir de los años 80 (Moore, Dickson-Deane, & Galyen, 2011) y en la actualidad, puede ser definido básicamente como una herramienta o sistema educativo basado en tecnologías de la información que permite aprender sin importar el lugar o el tiempo (TLMS, 2014). Al inicio, esta técnica estaba básicamente enfocada en el uso de CDs, sin embargo, con el paso del tiempo *E-learning* ha incorporado elementos como aplicaciones, páginas web, sistemas, objetos y en general programas que proveen oportunidades de aprendizaje a individuos (Moore, Dickson-Deane, & Galyen, 2011). En consecuencia, hay ciertas dificultades tratando de delimitar lo que cubre *e-learning* y por esto, Aparicio y Bacao (2014) propusieron la Tabla 2.2.1 que contiene todos los conceptos relacionados al tema.

Tabla 2.2.1.
Conceptos relacionados a e-learning.

Acónimo	Descripción	Concepto
<i>CAI</i>	Computer-Assisted Instruction	Uso de computadoras para enseñanza de programación.
<i>CBE</i>	Computer-Based Education	Enfocado en el uso de computadoras en educación
<i>CAL</i>	Computer-Assited Learning	Se enfoca en el uso de computadoras para la solución de problemas
<i>LMS</i>	Learning Management Systems	Orientado a mantener servicios y proveer información entre el docente y el estudiante.
<i>CMI</i>	Computer-Managed Instructions	Controla y provee las instrucciones de los docentes
<i>CAE</i>	Computer-Assisted Education	Se enfoca en el cómo los estudiantes usan las computadoras para la educación.
<i>e-learning</i>	Electronic Learning	Uso de recursos electrónicos para la enseñanza.
<i>ALE</i>	Artificial Learning Environments	Herramientas utilizadas como interfaces para la enseñanza en un ambiente determinado.
<i>M-Learning</i>	Mobile Learning	Está enfocado en la flexibilidad de recursos y del ambiente que rodea el aprendizaje.
<i>Sre</i>	Self-Regulatory Efficacy	Conceptos dirigidos al aprendizaje independiente de cada persona
<i>CSCL</i>	Computer Support for Collaborative Learning	Enfocado al aprendizaje grupal
<i>REAL</i>	Rich Environment for Active-Learning	Desarrollo de estrategias y habilidades de aprendizajes en entornos educativos
<i>B Learning</i>	Blended Learning	Combina educación basada en medios electrónicos con el modelo tradicional
<i>C-Mooc</i>	Connective Mooc	Enfocado a cursos online masivos. Donde hay intercambio de ideas y experiencias
<i>MOOC</i>	Massive Open Online Course	Orientado a la difusión libre de contenidos.
<i>ILM</i>	Internet-based learning medium	Enfocado en apoyar el aprendizaje de estudiantes

Fuente: Elaboración propia

Como se ha mencionado anteriormente, *E-learning* es una técnica que une la educación y la tecnología (Aparicio M. B., 2016). El marco que compone *e-learning* contiene tres elementos principales, tecnologías de aprendizaje, estrategias de enseñanza y el modelo pedagógico (Dabbagh, 2005). Tecnologías de aprendizaje es utilizar la tecnología para fortalecer el proceso de enseñanza, esto incluye aprendizaje basado en computadores y el uso de herramientas multimedia y redes durante el proceso de aprendizaje (Rist & Hewer, 1996). Por otra parte, las estrategias de aprendizaje son las actividades, planes y técnicas que realizadas por el profesor para hacer práctico el modelo pedagógico (Jonassen, 1997). Finalmente, el modelo pedagógico es la base de la teoría de aprendizaje. En él se unen los principios de la teoría de *e-learning* con la práctica. El modelo pedagógico está compuesto por: aprendizaje abierto, aprendizaje distribuido, comunidades de aprendizaje, comunidades prácticas y comunidades constructoras de conocimiento (Dabbagh, 2005).

Entre las ventajas que *E-learning* otorga está que reduce tiempos para buscar la información requerida, provee acceso a recursos electrónicos como artículos de revista o periódico, diarios de investigación y bases de datos. Además, da la oportunidad al alumno de tener material complementario para sus estudios (TLMS, 2014). De la misma manera, esta herramienta proporciona otros beneficios que se detallan en la Tabla 2.2.2 a continuación:

Tabla 2.2.2
Beneficios de E-learning

Beneficio	Descripción
<i>No existen restricciones</i>	E- learning es una herramienta que supera obstáculos de tiempo y espacio que normalmente, son considerados en un ambiente de enseñanza tradicional.
<i>Es entretenido, capta la atención de los estudiantes</i>	Es común usar herramientas multimedia que hacen que el curso sea más interactivo y entretenido.
<i>Mejora los costos</i>	Elimina la necesidad de estar invirtiendo en libros y material que después de un tiempo de vuelve obsoleto.
<i>Se adapta al ambiente educativo</i>	Con la revolución de la tecnología, el uso de <i>e-learning</i> permite al docente mantenerse actualizado en su material.

Fuente: Elaboración propia basado en (TLMS, 2014)

Tal como se puede observar en la Tabla 2.2.2, *e-learning* provee varios beneficios al proceso de aprendizaje de los estudiantes. Sin embargo, su efectividad va a depender completamente de la técnica de *e-learning* que se decida utilizar. Para elegir una técnica de *e-learning* se deben considerar los aspectos que se presentan en la Figura 2.3.

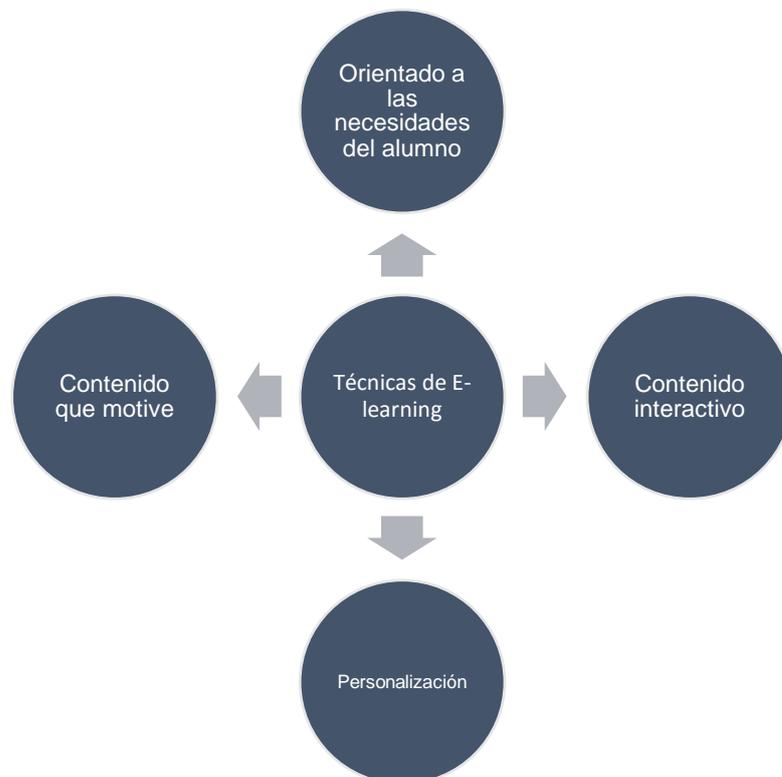


Figura 2.3 Características de técnicas de E-learning

Fuente: Elaboración propia basado en (Amit, 2015)

El primer aspecto, orientado a las necesidades del usuario, quiere decir que el contenido que se utilice en la herramienta de *e-learning* tiene que estar relacionado a las necesidades específicas del alumno. En otras palabras, la información que la herramienta ofrezca debe ser relevante para el mismo. Por otro lado, el segundo aspecto, contenido interactivo, describe que el medio de aprendizaje debe motivar al estudiante a aprender. De la misma forma, el contenido debe ser cautivador, es decir, la herramienta de

e-learning tiene que utilizar métodos y técnicas que mejoren la experiencia de aprender. Finalmente, en cualquier herramienta de *e-learning* existe un grado de personalización. Es decir, debe existir alguna característica que permita al docente seguir el avance del estudiante. A partir de estos aspectos, se puede elegir entre el modo asíncrono y síncrono de *e-learning*.

2.2.1 *E-learning* síncrono

Esta modalidad de *E-learning* es un estilo de enseñanza que ocurre en tiempo real (Amit, 2015). Requiere que las personas involucradas en el proceso de aprendizaje estén disponibles al mismo tiempo en la plataforma que se vaya a utilizar. En este modo, el estudiante tiene una experiencia parecida a estar en un entorno de clases tradicional (Higley, 2013). Además, ofrece un ambiente donde la interacción cara a cara es valorada y los estudiantes están sujetos a participar de las mismas actividades (Harris, Mishra, & Koehler, 2009). A continuación, se muestran en la Tabla 2.2.3 las herramientas más usadas dentro de esta modalidad.

Tabla 2.2.3
Herramientas de e-learning síncrono

Herramienta	Descripción
<i>Conferencias</i>	Contiene video conferencias o conferencias telefónicas que representan un salón de clases tradicional. El docente expone el tema y los estudiantes interactúan y participan.
<i>Chat</i>	Es una herramienta que permite la transferencia de archivos de texto, imágenes y videos entre varias personas. Es un medio para la transmisión de información.
<i>Mensajería Instantánea</i>	Esta herramienta se basa en envíos de manera instantánea entre dos individuos.

Fuente: Elaboración propia basado en (Higley, 2013)

2.2.2 E-learning asíncrono

De acuerdo con Higley (2013), *e-learning* asíncrono es una modalidad de *e-learning* donde el estudiante o el profesor pueden estar offline. En este ambiente el estudiante puede participar activamente en su proceso de aprendizaje. También, puede interactuar con otros compañeros, recibir retroalimentación de las actividades que realiza y monitorear el progreso de sus metas de enseñanza (Er, Özden, Arifoglu, & A., 2009). Normalmente, al utilizar este modo de *e-learning* se espera que el estudiante trabaje con los conocimientos y habilidades adquiridas. Para cumplir con este propósito, se realizan actividades donde el alumno crea, sintetiza, explica y aplica los conocimientos (Harris, Mishra, & Koehler, Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed, 2009).

Meloni (2010) menciona que esta modalidad es la más usada ya que, el estudiante puede avanzar el proceso de aprendizaje en su propio tiempo, necesita de menos recursos y es menos costosa. Además, provee de ciertos beneficios a quien la utiliza, por ejemplo: aumenta el portafolio electrónico del estudiante, fortalece el trabajo colaborativo y respeta las necesidades particulares de cada individuo (Hrastinski, 2008). El modo de *e-learning* asíncrono contiene varias herramientas que se las puede observar en la Tabla 2.2.4.

Tabla 2.2.4

Técnica de E-learning asíncrono.

Herramienta	Descripción
<i>Auto-estudio</i>	Estas herramientas incluyen blogs, wikis, foros y presentaciones <i>power point</i> .
<i>Transmisión de Audio y video</i>	Se utilizan videos para impartir información que puede ser reproducida un sin número de veces por el estudiante.
<i>CBT y WBTs</i>	Contiene cursos electrónicos basados en internet o por cds. El individuo completa el ciclo de aprendizaje en el tiempo que lo desee.
<i>M-learning</i>	Contiene material adaptado para la enseñanza a través de aplicaciones móviles.
<i>Aprendizaje Social</i>	Utiliza las redes sociales para intercambiar conocimiento.
<i>Simulación</i>	Esta técnica usa software de simulación para enseñar habilidades a los alumnos.
<i>Aprendizaje basado en juegos</i>	Los juegos o los elementos de un juego son herramientas que se utilizan para motivar al estudiante y ofrecen una oportunidad para usar el aprendizaje basado en la experiencia.

Fuente: Elaboración propia basado en (Higley, 2013)

2.2.3 Aprendizaje basado en juegos

El aprendizaje basado en juegos o elementos de juego, ofrecen una plataforma que complementa las estrategias de enseñanza tradicional y hacen que aprender sea más entretenido para el estudiante (Boyle, 2011).

De acuerdo con Fuszard (2001), a través de los juegos, los estudiantes son incentivados a usar un comportamiento creativo y pensamiento divergente. En la siguiente Figura 2.4, se muestran los beneficios del aprendizaje basado en juegos.



Figura 2.4 Beneficios del Aprendizaje basado en juegos

Fuente: Elaboración propia

Por otra parte, para implementar la técnica de aprendizaje basado en juegos, hay distintas herramientas que se pueden utilizar. Su elección va a depender de los elementos del juego y las características que se deseen implementar en la misma. Las alternativas se muestran a continuación en la Tabla 2.2.5

Tabla 2.2.5

Herramientas de Aprendizaje basado en juegos.

	Pensamiento de Juego	Elementos de Juego	Tácticas de Juego	Diversión
Diseño Inspirado en Juego	■	■	■	■
Gamification	■	■	■	■
Serious Games	■	■	■	■
Juego	■	■	■	■

Fuente: Elaboración Propia basado en (Marczewski, 2013)

2.2.4 Serious Games

El término *Serious Games* ha sido usado en los últimos 40 años. Se utiliza para describir juegos o simulaciones para administradores, estudiantes y profesores en un contexto educativo. El principal propósito de estos juegos es el aprendizaje, pero esto no quiere decir que el entretenimiento no es importante (Abt, 1970). Egenfeldt-Nielsen (2007), describe *serious game* como cualquier juego que tiene un propósito distinto a sólo entretenimiento como por ejemplo enseñanza. El propósito principal de esta técnica es utilizar la tecnología como medio de enseñanza o de entrenamiento (Felicia, 2009).

De acuerdo con Egenfeldt-Nielsen (2008), los términos que se muestran a continuación también describen *serious games*:

- Juegos Educativos
- Simulación
- Realidad Virtual
- Aprendizaje basado en juegos
- Juegos de impacto social
- Juegos para el cambio
- Juegos para el bien

2.2.5 Gamification

De acuerdo con McGonigal (2011), el término *Gamification* ha ido tomando importancia en los últimos años, Nick Pelling fue el primero en usarlo alrededor del año 2002 (Pelling, 2011). *Gamification* es una herramienta enfocada en mejorar la experiencia de aprendizaje por medio del uso de elementos de juego en ambientes que no son un juego (Deterding, Sicart, Nacke, O'Hara, & Dixon, 2011). Marczewski (2012) menciona que *Gamification* lleva las metáforas del juego a aplicaciones de la vida real influyendo en el comportamiento y mejorando la motivación del estudiante. De la misma forma Kapp (2012) establece que, a través de la

implementación de las mecánicas del juego, se promueven procesos de aprendizaje y habilidades para resolver problemas. Es importante mencionar, que *Gamification* es una práctica que se implementa para mejorar un proceso o servicio que ya existe (Leaning, 2015).

Por otro lado, Kapp (2013) menciona que hay dos tipos de *Gamification*: estructural y conceptual. En *Gamification* Estructural se utilizan los elementos del juego en una aplicación sin alterar el contenido. Es decir, los elementos de diseño de juego se agregan al contenido que se busca enseñar. Usualmente, esta técnica se aplica cuando se quiere motivar al estudiante a completar el proceso de aprendizaje. Es por esta razón que los elementos más utilizados son puntos, premios, niveles, tabla de posiciones. Por otro lado, *Gamification* Conceptual implementa una historia o un contexto al contenido que se busca enseñar. En consecuencia, se utilizan elementos como personajes, historias y desafíos.

Hay varias prácticas que han sido consideradas exitosas al usar *Gamification* (Bhasin, 2014). Las prácticas de diseño que se pueden agregar en un proceso gamificado son las que se muestran en la Tabla 2.2.6.

Tabla 2.2.6
 Elementos de diseño de Gamification
 Principios de diseño

<i>Principios de diseño</i>	<i>Descripción</i>	<i>Mecánicas de Juego</i>
<i>Metas</i>	Metas específicas y claras	
<i>Desafíos</i>	Claros, concretos y con las dificultades relacionadas al tema de aprendizaje	Niveles, premios
<i>Personalización</i>	Experiencia personalizada para el conocimiento y necesidades del usuario	Usuarios
<i>Progreso</i>	El progreso es visible para el usuario	Puntos, niveles, barras de progreso.
<i>Retroalimentación</i>	El usuario recibe retroalimentación inmediata de las actividades que realiza	
<i>Competencia</i>	Se involucra con otros usuarios	Tabla de posiciones, puntos, placas.
<i>Verificar Status</i>	El usuario puede ver el progreso del proceso de enseñanza	Puntos, tablas
<i>Contar Historias</i>	La actividad gamificada puede contar una historia para que el usuario se sienta identificado.	

Fuente: Elaboración propia basado en (Dicheva, Dichev, Agre, & Angelova, 2015)

Para diseñar una aplicación gamificada, Kevin Werbach (2013), propuso un marco de trabajo denominado 6d Framework. En él, se definen seis pasos para desarrollar la aplicación. El primer paso es definir el objetivo de negocio; es decir, se debe identificar cuál es el propósito del proyecto. El segundo paso es definir el comportamiento de la aplicación; en este paso se deben especificar las tareas que se van a cumplir y la forma en la que se va a medir el éxito de las mismas. Por otro lado, el tercer paso es describir los jugadores; su edad, demografía y comportamiento. El

cuarto paso es definir actividades repetitivas. Pueden ser actividades repetitivas motivadores o progresivas, pero se debe definir el comportamiento de las mismas. El quinto paso es no olvidar ser divertido; este paso quiere decir que el proyecto debe capturar y motivar al usuario. El último paso es usar las herramientas apropiadas para mantener un diseño interactivo.

Este *framework* se puede resumir en Tabla que se muestra a continuación.

Tabla 2.2.7

Framework 6d Gamification

Framework 6D de Gamification

1. Definir Objetivos de Negocio
2. Definir el Comportamiento de la Aplicación
3. Describir los usuarios/jugadores
4. Definir las actividades repetitivas
5. No dejar de capturar la atención
6. Designar herramientas de diseño

Fuente: Elaboración propia basado en (Werbach, 2013)

Por otro lado, dos aspectos importantes en aplicaciones que utilizan *Gamification* son la mecánica y la dinámica del juego (BunchBall, 2010). La mecánica del juego hace referencia a las herramientas de control, acciones y comportamientos que se utilizan para Gamificar una aplicación. Por otro lado, la experiencia motivadora y como resultado las acciones y deseos del usuario se denominan dinámica del juego. En la siguiente Tabla se muestran los componentes de estos aspectos.

Tabla 2.2.8
Mecánica y Dinámica de Gamification

Mecánica del Juego	Dinámica del Juego
Puntos	Recompensa
Desafíos	Logros
Niveles	Status
Recompensas	Competencia
Tablas de Posición	
Desafíos	

Fuente: Elaboración propia basado en (BunchBall, 2010)

2.3 Ingeniería de Software

La ingeniería de *software* es el área que se encarga de los “principios y metodologías para el desarrollo y mantenimiento de software” (Montoya-Suárez & Pulgarín-Mejía, 2013). Esto comprende, el diseño, desarrollo, pruebas y mantenimiento del *software*. Además, comprende todas las metodologías que aseguran la implementación eficiente de aplicaciones (Boehm, 2001). Es decir, combinar las metodologías, herramientas y procesos para producir *software*.

Dentro de la ingeniería de *software*, se encuentra la gestión de proyectos. Aquí el proyecto es un esfuerzo temporal de tiempo y recursos que tiene como resultado un producto o servicio. En este caso el resultado del proyecto es el *software*.

2.3.1 Gestión de Proyectos

La Gestión de Proyectos tiene como finalidad principal la planificación y administración de recursos para la implementación de un sistema de información. Es un proceso continuo, que busca que el resultado final agregue valor y vaya de acuerdo con los objetivos de la organización (Bedini González, 2006) . El estándar más reconocido para manejar la gestión de proyectos es el PMBOK (Ospina, 2013). Esta guía, incluye las buenas prácticas en administración de proyecto y en ella mencionan 9 áreas de conocimiento que se muestran en la Figura 2.5.



Figura 2.5 Dimensiones de Gestión de Proyectos

Fuente: Elaboración propia basado en (PMI, 2012)

2.3.2 Gestión de Riesgos

De acuerdo con Isaca (2015) un riesgo es “un evento probable que si se produce podría tener un impacto positivo/negativo sobre los objetivos del proyecto”. Los riesgos están compuestos por el evento, la probabilidad y el impacto. Además, la Gestión de los Riesgos del Proyecto incluye los procesos relacionados con llevar a cabo la planificación de la gestión, la identificación, el análisis, la planificación de respuesta a los riesgos, así como su monitoreo y control en un proyecto (López, 2013). La gestión de riesgos busca disminuir los riesgos que causan un impacto negativo en el proyecto y aprovechar las oportunidades que se presentan.

En la siguiente Tabla 2.3.1 se muestran los procesos de la gestión de riesgos y su descripción.

Tabla 2.3.1
Procesos de la gestión de Riesgos

Proceso	Descripción
<i>Planificar de Riesgos</i>	Define las actividades que se van a seguir para administrar los riesgos de un proyecto.
<i>Identificación de Riesgos</i>	Se descubren e identifican los riesgos del proyecto.
<i>Análisis Cualitativo de Riesgos</i>	Se realiza un análisis de la importancia de riesgos tomando en cuenta el impacto y la probabilidad.
<i>Análisis Cuantitativo de Riesgos</i>	Es un análisis considerando el costo de los riesgos y su impacto en la organización.
<i>Planificar respuesta a riesgos</i>	Son las acciones que se van a tomar en respuesta a los riesgos.
<i>Monitorear Riesgos</i>	Por medio de este proceso se ejecutan los planes de respuesta a riesgos cuando es necesario. También se identifican nuevos riesgos.

Fuente: Elaboración propia basado en (ISACA, 2015)

2.4 Enseñanza de Ingeniería de Software

Zapata-Jaramillo,C & Awad-Aubad (2007),establecen que para participar en cursos de Ingeniería de software se necesitan habilidades administrativas que muchas veces no se logran practicar si se utilizan metodologías tradicionales de enseñanza. Como se ha visto anteriormente, la enseñanza no se basa en los conocimientos que imparte el docente sino,

en el estudiante. Por esta razón en la actualidad es común implementar aprendizaje basado en juegos como herramienta pedagógica (Zapata-Jaramillo & Villegas, Reglas de consistencia entre modelos de requisitos de UN-Metodo, 2012).

Comúnmente, se ha considerado la Ingeniería de Software como un proceso bastante complicado de comprender. Es normal observar retrasos en el tiempo de los proyectos o que los recursos designados inicialmente no se manejen adecuadamente (Zapata-Jaramillo, Gelbukh, Arango, & Hernández, 2006). También es común que, a pesar de los esfuerzos, los proyectos de desarrollo de software fracasen (Zapata-Jaramillo C. M., 2006). Por estas razones Boehm (2001) propone una enseñanza de ingeniería de *software* que combine los aspectos gerenciales, conceptuales y técnicos con el propósito de mejorar y practicar los principios y estrategias de la materia. Una alternativa para lograr esto es utilizar aprendizaje basado en juegos el cual logra que el estudiante “afianzar, estructurar, analizar los conocimientos sobre modelado, métodos de desarrollo de software, trabajo en equipo y comunicación en la ingeniería de software” (Montoya-Suárez & Pulgarín-Mejía, 2013).

Se han definido tres elementos que deben interactuar en la enseñanza de ingeniería software y son: elementos conceptuales, procedimentales e integradores (Guitart, Rodríguez, & Cabot, 2006). Los elementos conceptuales son aquellos que describen el conocimiento teórico. Por otra parte, los elementos procedimentales manejan el conocimiento práctico sobre un tema. Finalmente, los elementos integradores es la combinación entre el conocimiento práctico y teórico de un tema.

2.5 Metodologías de desarrollo ágil

Hay varias metodologías que se pueden implementar al desarrollar Software. Las metodologías orientadas al desarrollo ágil son aquellas que están enfocadas de manera especial en el individuo y al desarrollo del producto por medio de constantes entregas (Penadés, 2006). Los métodos ágiles son muy efectivos en ambientes donde los requerimientos del cliente pueden cambiar, pero desean mantener el tiempo del proyecto (Abrahamsson, Salo, Ronkainen, & Warsta, 2002). Entre las metodologías de desarrollo ágil más destacadas están las siguientes: *Scrum*, *Crystal Methodologies*, *Dynamic Systems Development Methods* y *Adaptive Software Development*.

2.5.1 Scrum

Scrum puede ser definido como un marco para la administración y organización de proyectos tecnológicos. Está enfocada en proyectos que tienen requerimientos variables donde cada liberación del producto se lleva a cabo a través de iteraciones denominadas Sprint (Schwaber, Beedle, & Martin, 2003). El Sprint es el elemento más importante dentro del marco de trabajo de *Scrum*. De acuerdo con Shuber (2013), es el “corazón de *Scrum*” y consiste en un periodo de tiempo específico, donde se definen actividades que terminan con un incremento del producto. Tomando en cuenta el marco de *Scrum*, cada *Sprint* del proyecto contiene cuatro eventos que ayudan a controlar la inspección y adaptación del producto, estos son: planificación del sprint, reunión diaria, revisión del sprint y retrospectiva del sprint.

Además, cada *Sprint* del proyecto puede ser considerado como un proyecto particular ya que es temporal y acaba con el alcance de los objetivos planteados inicialmente. Shuber (2013) menciona que, durante cada *Sprint*, se pueden realizar cambios, sin embargo, estos no pueden cambiar o afectar los objetivos del proyecto. De la misma forma, “cada

Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante”.

CAPÍTULO 3 METODOLOGÍA

En la Metodología, se expone el proceso y las herramientas utilizadas para cumplir con los objetivos planteados en el Capítulo 1. Es decir, se describen de forma detallada los pasos que se van a seguir para cumplir con el propósito de la investigación dentro del alcance previamente especificado. A continuación, se van a detallar el enfoque, el alcance y el diseño de este trabajo.

La presente investigación tiene un enfoque cualitativo y alcance exploratorio. Es por esto, que se va a realizar una revisión sistemática en relación a las metodologías de aprendizaje, *e-learning* y sus técnicas, *Gamification* y Gestión de Proyectos, especialmente Gestión de Riesgos. Luego de esta revisión, se van a analizar las distintas técnicas de *e-learning* y sus beneficios para un modelo interactivo de aprendizaje. De la misma forma, se va a desarrollar una aplicación web utilizando los elementos de *Gamification* para la enseñanza de Gestión de Riesgos.

El diseño de la investigación consiste en dos etapas; preparación del producto y desarrollo de la aplicación. En la primera etapa, preparación del producto, se va a preparar todo el ambiente y la información necesaria para elaborar la aplicación web. Por otro lado, para la etapa de desarrollo de la aplicación se va a utilizar un marco de trabajo ágil, el cual está orientado a resolver problemas complejos por medio de soluciones que entregan el máximo valor posible (Shuber, 2013). La metodología ágil que se va a usar es *Scrum*.

3.1 Etapa 1: Preparación del Producto

Durante la etapa de preparación para el desarrollo del producto, se realizaron las siguientes actividades:

1. Definición del problema a solucionar.
2. Revisión sistemática de la información relacionada al tema, esto incluye:
 - a. Definición y comparación de metodologías de enseñanza
 - b. Definición de *e-learning* y sus técnicas
 - c. Definición de *Gamification*
 - d. Análisis comparativo entre distintas técnicas de *e-learning*
 - e. Elección y justificación de técnica de *e-learning*
 - f. Áreas de aprendizaje en gestión de proyectos
 - g. Definición e importancia de gestión de riesgos
3. Búsqueda y elección de solución apropiada considerando los resultados obtenidos de la revisión sistemática. En la tabla 3.1.1, se muestra un resumen de las actividades realizadas en esta etapa.

Tabla 3.1.1
Actividades de Preparación del Producto

Tarea	Entrada	Resultado
<i>Análisis de técnicas de e-learning</i>	Información sobre técnicas de <i>e-learning</i> (<i>Gamification</i> , <i>Serious Games</i>)	Tabla comparativa y elección de técnica más apropiada
<i>Propuesta para solucionar el problema.</i>	Información actual sobre enseñanza Gestión de Proyectos e investigación sobre otros experimentos realizados. Entrevistas para conocer los requerimientos de la solución	Ficha de especificaciones técnicas de solución. Definición de caso de estudio y requerimientos básicos de la aplicación

Fuente: Elaboración propia

3.2 Etapa 2: Desarrollo de la Aplicación

Tal como se mencionó anteriormente, para la Etapa 2; desarrollo de la aplicación, se va a usar el marco de trabajo *Scrum*. De acuerdo con Shuber (2013), *Scrum* es “ligero y fácil de entender”. Por otro lado, *Scrum* cuenta con artefactos, reglas y herramientas que tienen un propósito específico y son esenciales para éxito en el desarrollo del producto. En el presente trabajo de titulación se van a utilizar las herramientas y artefactos que se describen a continuación.

3.2.1 Sprint

El desarrollo del producto se realizó a través de *Sprints*. En la siguiente Tabla 3.2.1 se describen los *sprints* esperados para el presente trabajo de titulación.

Tabla 3.2.1
Sprints esperados en el desarrollo

Número de Sprint	Duración	Resultado
1	3 semanas	Aplicación con un caso, riesgos e ingreso de usuario
2	3 semanas	Aplicación con funciones para clasificar riesgos y mitigarlos
3	3 semanas	Aplicación con retroalimentación sobre actividades de gestión de riesgo

Fuente: Elaboración propia

3.2.2 Planificación del Sprint

La planificación del *Sprint* es el evento donde se van a definir las distintas tareas de los *Sprint* 1,2 y 3. Para planificar cada tarea del trabajo se debe

realizar una estimación del esfuerzo en horas de la misma. Como resultado de este evento se debe generar una tabla por planificación que contenga las tareas con la estimación y el número total de horas planificados por *Sprint*.

3.2.3 Ciclo diario (Reunión diaria)

En este evento, se revisan las actividades diarias y los posibles impedimentos que pueden surgir para terminar con lo planificado. Por medio de las reuniones diarias se pueden evitar impedimentos y desafíos que puedan surgir y además se promueve el conocimiento sobre los avances del proyecto.

En el presente trabajo de titulación, luego de realizar la reunión diaria, se procede a desarrollar las tareas asignadas para el día. Estas tareas van de acuerdo a la lista de pendientes de cada *Sprint*.

3.2.4 Revisión del *Sprint*

En esta etapa se revisa lo que se ha hecho durante el *Sprint*. Se detallan las tareas terminadas y no terminadas. Como resultado de la etapa, se obtiene un listado de tareas realizadas y pendientes.

En la revisión del *sprint* de este trabajo de titulación, también se van a realizar las siguientes actividades:

1. Pruebas de la aplicación
2. Revisión de características de la aplicación
3. Aprobación de características

3.2.5 Retrospectiva del *Sprint*

Durante esta etapa se evalúa si el *sprint* ha sido exitoso o no y, cuáles pueden ser las estrategias de mejora. Además, se identifican los elementos eficientes del *sprint* y como respuesta se realiza una planificación para implementar las mejoras.

Luego de acabar con esta etapa, se procede a liberar un incremento del producto y a iniciar el nuevo *sprint*. Al terminar con los *sprints* que se han planificado, se evalúa el producto y se aprueba la finalización del desarrollo del mismo.

3.3 Resumen de la metodología

En resumen, la metodología para el desarrollo de la aplicación gamificada que va a servir para la enseñanza de gestión de riesgos, va a utilizar el marco metodológico de *Scrum*. El uso de este marco es apropiado debido a la incertidumbre de la solución y los requisitos que se deben cumplir. *Scrum* es adaptable a las condiciones y cambios de requisitos. Además, por medio de la continua retroalimentación se puede corregir rápidamente cualquier error o definición que no sea adecuada. En la siguiente Figura 3 se describe el proceso.

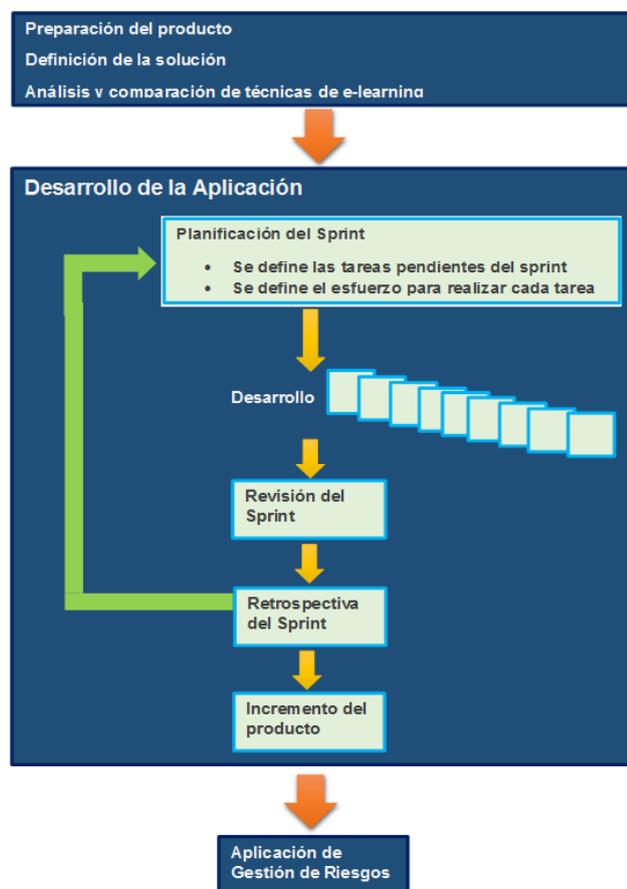


Figura 3 Resumen de Metodología

Fuente: Elaboración Propia

CAPÍTULO 4 DESARROLLO DE LA PROPUESTA Y RESULTADO

En los últimos años a causa de la evolución de la tecnología, han surgido técnicas de *e-learning* que sirven como soporte por los modelos de aprendizaje. En el capítulo uno y dos se describieron estos modelos, así como también las técnicas de *e-learning* y cómo es la situación actual de la enseñanza de ingeniería de software y gestión de proyectos.

Tal como se definió en la metodología de éste trabajo, las definiciones en el marco teórico son parte de la revisión sistemática de la preparación del producto que se va a desarrollar. En la revisión sistemática se logró: definir modelos de aprendizaje con sus características, *e-learning*, técnicas de *e-learning*, *Gamification*, *Serious Games*, gestión de proyectos, gestión de riesgos y cuál es el estado de la enseñanza de Ingeniería de Software.

4.1 Preparación del Producto

La etapa de preparación del producto se llevó a cabo previo al desarrollo. Durante la etapa de preparación del producto se cubren también los siguientes procesos:

1. Comparación de Técnicas de *E-learning*
2. Definición de solución

4.1.1 Comparación de Técnicas de *E-learning*

Para la comparación de técnicas de e-learning se tomaron en cuenta las técnicas de aprendizaje basado en juegos o elementos de juego. La siguiente Figura 4.1 resume las técnicas más importantes:



Figura 4.1 Técnicas de aprendizaje basado en juegos

Fuente: Elaboración propia

A partir de esta Figura 4.1 y de la revisión sistemática antes realizada en el Capítulo 2, se resume en la siguiente Tabla 4.1.1 la comparación entre las distintas técnicas antes presentadas.

En la Tabla 4.1.1, se puede apreciar que cada técnica tiene un propósito diferente. Por un lado, en el diseño basado en juegos sólo se

quiere ofrecer una interface llamativa para el usuario final; es por eso que se implementan ideas de juego en menús o elementos de la aplicación que mejoran la percepción de la página. En cuanto a *Gamification*, el fin es capturar la atención del usuario a través de la adhesión de mecánicas de juego en un contexto diferente a juego. Finalmente, *Serious Games*, son juegos desarrollados con un propósito específico.

Tabla 4.1.1
Comparación de técnicas de aprendizaje basado en juegos

	Diseño de Juego	Gamification	Serious Games
<i>Propósito</i>	Brindar mejor experiencia al usuario	Influir en el comportamiento del usuario, capturando la atención	Tiene un propósito que no es sólo entretenimiento
<i>Implementación</i>	Estética y usabilidad basada en funcionamiento de juegos	Toma elementos e ideas de juegos en un contexto que no es un juego	Implementa todas las características de un juego, es un juego que tiene un fin específico
<i>Elementos</i>	Sólo se usan elementos en la interface para hacerla más agradable	Puntos, desafíos, niveles, usuarios, tabla de posiciones, historias, misiones	Mecánicas de juego, desafíos, búsqueda, simulaciones, animaciones,

Fuente: Elaboración propia basado en Figura 4.1 y Secciones 2.2.3, 2.2.4, 2.2.5

Tras haber comparado Diseño basado en Juegos, *Gamification* y *Serious Games*, es evidente que la técnica Diseño basado en Juegos no

puede ser usada para el desarrollo de la aplicación de Gestión de Riesgos ya que no ofrece ninguna ventaja significativa. Por otro lado, las dos técnicas: *Gamification* y *Serious Games* sí. Para justificar la elección de una de las técnicas se va a tomar en cuenta el árbol de decisión propuesto por Marczewski (2013).

Tabla 4.1.2

Árbol de decisión de técnicas de aprendizaje basadas en juegos.

<i>¿Es el único propósito entretenimiento?</i>					
Sí	No				
Juego	<i>¿Necesita crear un ambiente virtual para probar situaciones reales?</i>				
	Sí	No			
	Simulación	<i>¿Necesita de un juego que cumple con un propósito y enseñe algún tema?</i>			
		Sí	No		
		Serious Games	<i>¿Necesitas implementar elementos de diseño para motivar y capturar la atención del usuario?</i>		
			Sí	No	
	<i>Gamification</i>	Diseño inspirado en Juegos			

Fuente: Elaboración propia basado en (Marczewski, 2013)

Al realizar las preguntas propuestas por Marczewski (2013), se pudo concluir que la técnica más apropiada, considerando el contexto de la aplicación, es *Gamification*. Si se aprovecha las herramientas que provee esta técnica, los usuarios de la aplicación van a estar motivados a seguir con el proceso de enseñanza. Tomando en cuenta esto, y la revisión

sistemática previa. Por medio de la implementación de una aplicación Gamificada, los conceptos y las acciones relacionadas a gestión de riesgos van a ser mejor asimilados por los estudiantes que participen de cursos de Gestión de proyectos.

A continuación, luego de evaluar las técnicas de *e-learning* y justificar la elección de *Gamification*, se va a definir las características iniciales de la aplicación Gamificada.

4.1.2 Definición de la Solución

Para definir la solución se tomó en cuenta la propuesta de Uyaguari, Intriago & Salazar (2015). En ella exponían la situación actual del programa de Maestría en Ingeniería de Sistemas de la Universidad de las Fuerzas Armadas de Ecuador (ESPE). Durante siete años, se ha enseñado como parte del programa, un curso de Gestión de Riesgos. El propósito del mismo es que los estudiantes aprendan a: identificar, analizar y gestionar riesgos.

Actualmente, en el curso utilizan un modelo tradicional de enseñanza. Es decir, considerando la revisión literaria, el profesor es el centro del proceso de aprendizaje y el estudiante es un receptor pasivo de información. El profesor utiliza casos de estudio que los expone a la clase y guía a los estudiantes a resolverlos. Sin embargo, como se ha expuesto anteriormente, a través de una aplicación Gamificada el sistema de aprendizaje de los estudiantes del curso de Gestión de Riesgos va a mejorar considerablemente. En consecuencia, la aplicación es desarrollada para este curso de gestión de riesgos, aunque puede ser usada por otros usuarios también.

Entre las características iniciales de la solución se han definido lo siguientes basados en marco de trabajo 6D:

Tabla 4.1.3
Características Iniciales de la Aplicación
Premisas

1. <i>Definir objetivo del negocio</i>	Desarrollar aplicación que genere valor en el sistema de enseñanza de gestión de proyectos específicamente gestión de riesgos.
2. <i>Delinear el comportamiento de la aplicación</i>	La aplicación va a tener un ingreso de usuarios donde los estudiantes podrán ingresar con su usuario y contraseña. Luego se presentará un caso con el cual van a aplicar los conceptos anteriormente visto en el curso. Durante el desarrollo del caso, los estudiantes van a identificar los riesgos, analizarlos (darles valoración) y establecer el mecanismo de mitigación. Durante cada etapa se va a realizar un proceso de retroalimentación para clarificar cualquier concepto.
3. <i>Describir jugadores</i>	Los usuarios de la aplicación van a ser aquellos estudiantes que pertenezcan al curso de gestión de riesgos
4. <i>Describir lazos de actividades</i>	Durante la etapa de valorización del riesgo el estudiante debe hacer la misma actividad de definir probabilidad e impacto del riesgo hasta que acaben los riesgos de los casos que se han presentados. Luego de recibir las correcciones, procederá a decidir bajo un presupuesto cuántos riesgos va a mitigar.
5. <i>No olvidar la diversión</i>	Con el propósito de mantener entretenido al usuario la aplicación va a ser desarrollada con estilos y diseños amigables.
6. <i>Implementar herramientas apropiadas</i>	Para desarrollar la aplicación se van a utilizar las siguientes herramientas: <ul style="list-style-type: none"> • Lenguaje: Java, Ajax JavaScript, CSS • Plataforma: Aplicación WEB • Base de Datos: MySql • Servidor de Pruebas: Glassfish

Fuente: Elaboración propia

Para desarrollar las premisas dos y cuatro del marco 6D, se va a considerar el siguiente caso de estudio. Con este caso y los riesgos que se exponen, los estudiantes van a proceder a realizar la identificación, análisis y gestión de riesgos.

Tabla 4.1.4

Caso de Estudio de la Aplicación

Caso de Estudio: Identificación, Análisis y Gestión de Riesgos

Los servicios públicos locales de salud almacenan datos de consultas de pacientes de hospitales y centros de salud. Actualmente los datos son usados únicamente con propósitos estadísticos. Los hospitales y centros de salud tienen diferente número de historia clínica, esto significa que no se tiene un identificador único para los pacientes. El Ministerio de Salud desea usar un sistema como base para establecer una base de datos de historias clínicas. La base de datos ofrecerá a los hospitales y centros de salud, acceso a las historias clínicas de los pacientes, permitiendo que las decisiones sean tomadas de forma inmediata sin tener que esperar que sean transferidos los registros médicos. El Sistema Médico intenta migrar todos los datos existentes, en una nueva base de datos histórica. Una vez que el nuevo sistema es desplegado, se intenta que todos los pacientes reciban tratamiento estén con un único número de historia clínica que sea usado en los centros médicos y hospitales. El tiempo de entrega del sistema funcionando será de dos años y el presupuesto es de 4 millones de dólares americanos.

Se tiene realizado el levantamiento de los requisitos mediante un proyecto anterior de ingeniería de requerimientos. Se va a desarrollar con la técnica *Test Driven Development* (TDD). El equipo de desarrollo está constituido por desarrolladores junior con máximo dos años de experiencia en programación y sin experiencia en TDD.

Evaluar los niveles de integridad de seguridad del sistema. Existe por ejemplo un número de riesgos de seguridad inherente en el sistema propuesto. Por ejemplo, el riesgo de que los datos existentes en los sistemas antiguos no sean identificados correctamente y asignados al paciente correcto o que el paciente en el nuevo sistema no se le enlace su historial previo correctamente. En ambos casos podría ocurrir un diagnóstico o tratamiento incorrecto debido a que la historia clínica completa no está disponible.

Fuente: Elaboración propia

Los riesgos que se van a utilizar para la aplicación Gamificada, se presentan en la siguiente *Tabla 4.1.5*

Tabla 4.1.5
Riesgos del Caso de Estudio

ID	Descripción del Riesgo
1	Poca definición de los requerimientos.
2	Es posible que no se hayan identificado interfaces con otros sistemas.
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.
4	Las especificaciones podrían no estar a suficiente detalle para escribir el código.
5	La integración del producto no estaría bien especificada. Lo relativo a los códigos de los pacientes.
6	Posiblemente no haya especificaciones de requerimientos de seguridad de los datos de los pacientes.
7	Posiblemente no se haya especificado los planes para todas las actividades del desarrollo. Especialmente para la fase requerimientos y gestión.
8	Los desarrolladores no podrían estar familiarizados con la técnica TDD, incluso luego de haber recibido capacitación.
9	Posiblemente no se maneje bien el control de cambios.
10	Poca familiaridad con Eclipse y JUnit por parte de los desarrolladores
11	Demoras en el tiempo de desarrollo debido a que el personal podría no estar entrenado adecuadamente para el proyecto.
12	El cronograma no podría ser realista, posiblemente no se haya incluido la etapa de migración de los datos.

Fuente: Elaboración propia

Estos riesgos que se presentan en la *Tabla 4.1.5* van a ser valorizados considerando dos aspectos: el impacto y la probabilidad. El impacto y la probabilidad pueden ser medidos como: Muy Alto, Alto, Medio,

Bajo y Muy Bajo. El resultado de la valoración se basa en los resultados de la medición del impacto y de la probabilidad como se puede ver en la *Tabla 4.1.6* a continuación.

Tabla 4.1.6
Valoración de Riesgos

Riesgo		Impacto				
		Muy Alto	Alto	Medio	Bajo	Muy Bajo
Probabilidad	Muy Alto		Muy Alto	Alto	Medio	Bajo
	Alto	Muy Alto	Muy Alto	Alto	Medio	Bajo
	Medio	Muy Alto	Alto	Medio	Bajo	Muy Bajo
	Bajo	Muy Alto	Alto	Medio	Bajo	Muy Bajo
	Muy Bajo	Alto	Medio	Bajo	Muy Bajo	Muy Bajo

Fuente: Elaboración propia

Tomando en cuenta los valores de la Tabla 4.1.6 *Valoración de Riesgos*, los riesgos del caso que se presentaron anteriormente, toman la siguiente valoración:

Tabla 4.1.7
Valoración de Riesgos del Caso

ID	Descripción del Riesgo	Impacto	Probabilidad	Valoración
1	Poca definición de los requerimientos.	Alto	Alta	Alto
2	Es posible que no se hayan identificado interfaces con otros sistemas.	Muy Alto	Muy Alta	Muy Alto
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.	Alto	Media	Alto
4	Las especificaciones podrían no estar a suficiente detalle para escribir el código.	Alto	Alta	Alto
5	La integración del producto no estaría bien especificada. Lo relativo a los códigos de los pacientes.	Muy Alto	Muy Alta	Muy Alto
6	Posiblemente no haya especificaciones de requerimientos de seguridad de los datos de los pacientes.	Alto	Baja	Alto
7	Posiblemente no se haya especificado los planes para todas las actividades del desarrollo. Especialmente para la fase requerimientos y gestión.	Medio	Alta	Alto
8	Los desarrolladores no podrían estar familiarizados con la técnica TDD, incluso luego de haber recibido capacitación.	Alto	Media	Alto
9	Posiblemente no se maneje bien el control de cambios.	Medio	Alta	Alto
10	Poca familiaridad con Eclipse y JUnit por parte de los desarrolladores. Podría causar retrasos y bajar la calidad del software.	Medio	Alta	Alto
11	Demoras en el tiempo de desarrollo debido a que el personal podría no estar entrenado adecuadamente para el proyecto.	Alto	Alta	Alto
12	El cronograma no podría ser realista, posiblemente no se haya incluido la etapa de migración de los datos.	Alto	Alta	Alto

Fuente: Elaboración propia

Luego de las etapas de identificación y valoración (Análisis) de riesgos, se procede a ordenar los riesgos y elegir qué riesgos del caso se van a mitigar; para esto, se definió un costo y estrategia de mitigación.

Tabla 4.1.8

Mecanismos de Mitigación de Riesgos

ID	Descripción del Riesgo	Mecanismo de Mitigación	Costo (USD.)
2	Es posible que no se hayan identificado interfaces con otros sistemas.	Identificar las interfaces con otros sistemas y sugerir considerar los costos dentro del total del proyecto.	2.000
5	La integración del producto no estaría bien especificada. Lo relativo a los códigos de los pacientes.	Realizar un análisis detallado de la migración de los datos. Presupuestar y asignar recursos a las actividades relacionadas.	15.000
1	Poca definición de los requerimientos.	Realizar un proceso de validación de los documentos de requerimientos.	2.000
4	Las especificaciones podrían no estar a suficiente detalle para escribir el código.	El mecanismo que se aplica a este riesgo es el mismo del riesgo 1.	
11	Demoras en el tiempo de desarrollo debido a que el personal podría no estar entrenado adecuadamente para el proyecto.	Implementar <i>pair programming</i> con TDD en el primer mes de desarrollo.	20.000
12	El cronograma no podría ser realista, posiblemente no se haya incluido la etapa de migración de los datos.	Revisar el cronograma, los recursos, el tiempo y el alcance del proyecto. Definir las actividades para equilibrar los tres parámetros en caso de que sea necesario.	2.000
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.	Seleccionar una técnica de testing (Particiones de Equivalencia) y elaborar los casos de prueba con la técnica seleccionada. En caso de no tener conocimiento de las técnicas contratar una persona con experiencia en pruebas por el tiempo de las pruebas.	12.000
8	Los desarrolladores no podrían estar familiarizados con la técnica TDD, incluso luego de haber recibido capacitación.	Hacer pares de programación en donde haya una persona con habilidades para programar con TDD, se incluiría en el mecanismo para mitigar el riesgo 11.	
6	Posiblemente no haya especificaciones de requerimientos de seguridad de los datos de los pacientes.	Se confía en la seguridad del software de base y en la red en donde va a funcionar el software.	
7	Posiblemente no se haya especificado los planes para todas las actividades del desarrollo. Especialmente para la fase requerimientos y gestión.	Elaborar los planes de gestión (G. de la Configuración, Gestión de la calidad, G. de proyecto) y de ingeniería de requerimientos. Ajustar las actividades al cronograma del proyecto.	3.000
9	Posiblemente no se maneje bien el control de cambios.	Implementar la gestión de la configuración de acuerdo a un plan.	5.000
10	Poca familiaridad con Eclipse y JUnit por parte de los desarrolladores. Podría causar retrasos y bajar la calidad del software.	Entrenar a los desarrolladores en el uso de JUnit y Eclipse.	10.000

Fuente: Elaboración propia

Tras haber revisado la información del caso de estudio; es decir, la descripción del caso, los riesgos, la valoración de los riesgos y los mecanismos de mitigación y también, haber definido las especificaciones básicas de la solución se procede a iniciar el desarrollo de la misma por medio de una planificación preliminar del desarrollo. En la planificación preliminar, se definen las historias de usuarios que se deberán cumplir al finalizar todo el desarrollo de la aplicación. Las historias que tengan una prioridad más alta son las que van a formar parte de la lista de tareas del primer *Sprint*. Además, cada historia tiene una estimación del esfuerzo que refleja el peso de cada historia para el desarrollo del proyecto

A continuación, se muestra las historias de usuario en la Tabla 4.1.9

Tabla 4.1.9
Historias de Usuario

ID	Descripción	Usuario	Razón	Prioridad	Esfuerzo
1	Ingresar casos de gestión de riesgos	Administrador	Ingresar riesgos en una plataforma Web para que los estudiantes practiquen la teoría de Gestión de Riesgos	ALTA	3
2	Clasificar riesgos	Estudiante	Clasificar riesgos considerando el impacto y la probabilidad para definir cuáles son muy altos, altos, medios, bajos o muy bajos	MUY ALTA	4,5
3	Mitigar riesgos	Estudiante	Seleccionar los riesgos que se van a mitigar para ver si se cumple con el presupuesto, aplicando la mejor solución	ALTA	2,5
4	Acceder con un usuario y clave	Estudiante Profesor	Cada estudiante debe acceder con un usuario y clave para poder resolver el caso que ha sido asignado por su profesor	ALTA	3
5	Ver puntuaciones del caso	Estudiante Profesor	Ver puntuaciones para ver cuáles han sido los resultados de la resolución del caso	MEDIA	3
6	Ver correcciones de resultados	Estudiante Profesor	Ver correcciones para ver de qué manera hubiera sido mejor resolver el caso	MEDIA	3

Fuente: Elaboración propia

4.2 Desarrollo del Producto

Antes de empezar los *Sprints*, se definió el siguiente diagrama de arquitectura de aplicación. En él se ilustran los componentes de la solución. Como se puede observar la aplicación va a ser utilizada por el usuario a través de un navegador de Internet. La aplicación va a estar compuesta por tres capas, el diseño, el controlador y la interface de conexión. En la capa de diseño se van a definir estilos y demás funcionalidades visibles para el usuario que hacen atractiva la aplicación. Por otro lado, la capa del controlador va a validar que la información que el usuario ingrese a la aplicación sea válida. Luego la interface de conexión se va a encargar de enviar la aplicación al servidor de base de datos.

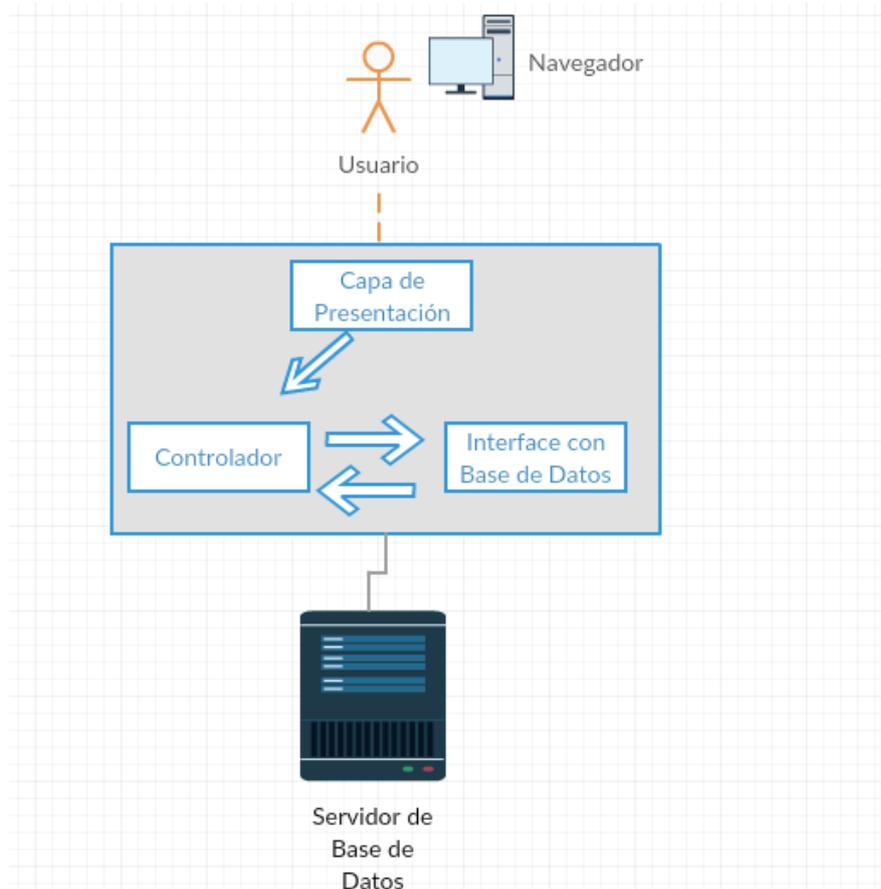


Figura 4.2 Diagrama de Arquitectura de Aplicación Web

Fuente: Elaboración propia

4.2.1 Sprint 1

4.2.1.1 Planificación del Sprint 1

Después de definir las historias de usuario, se definen las tareas que van a ser parte de este Sprint. Estas tareas también son evaluadas por el esfuerzo en horas que, se estiman van a requerirse para que sean completadas. Estas tareas se detallan a continuación:

Tabla 4.2.1

Lista de Tareas Sprint 1

Lista de Tareas Sprint 1	Esfuerzo
1. Modelar la base de datos	6
2. Crear tablas para ingresar el caso	3
3. Registrar el caso de estudio	1,5
4. Registrar riesgos en la base de datos	2
5. Guardar usuario	6
6. Validar usuario	6
Total	24,5

Fuente: Elaboración propia

A partir de estas tareas se realizar una planificación del trabajo en horas necesarias para llevarlas a cabo. La planificación del trabajo se muestra a continuación en la Tabla 4.2.2.

En resumen, en la Planificación del *Sprint1*, se busca entregar una versión del producto que contenga el caso de estudio presentado anteriormente y el ingreso y validación de los usuarios.

Tabla 4.2.2
Planificación del Trabajo Sprint 1

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Tarea 6	Total
Día 1	1h	-	-	-	-	-	1h
Día 2	2h	-	-	-	-	-	2h
Día 3	1h	-	-	-	-	-	1h
Día 4	2h	-	-	-	-	-	2h
Día 5	1h	-	-	-	-	-	1h
Día 6	-	-	-	-	-	-	-
Día 7	-	-	-	-	-	-	-
Día 8	-	2h	-	-	-	-	2h
Día 9	-	-	1h30	-	-	-	1h30
Día 10	-	-	-	1h	-	-	1h
Día 11	-	-	-	1h	-	-	1h
Día 12	-	-	-	-	1h	-	1h
Día 13	-	-	-	-	-	-	-
Día 14	-	-	-	-	-	-	-
Día 15	-	-	-	-	2h	-	2h
Día 16	-	-	-	-	3h	-	3h
Día 18	-	-	-	-	-	2h	2h
Día 19	-	-	-	-	-	1h	1h
Día 20	-	-	-	-	-	3h	3h
Día 21	-	-	-	-	-	-	-
Total	7h	2h	1h30	2h	6h	6h	24h30

Fuente: Elaboración propia

4.2.1.2 Desarrollo del Sprint 1

Durante el desarrollo del *Sprint 1*, se deben llevar a cabo las tareas planteadas en la Lista de la Tabla 4.2.1. La primera tarea que se va a desarrollar es el modelado de la base de datos.

1. Modelado de la Base de datos

El modelado de la base de datos fue el proceso donde se definieron las tablas y las relaciones entre las mismas. Luego de haber definido la estructura de la base de datos se procede crear las tablas para ingresar la información. El diagrama de entidad-relación se puede observar en el Anexo 1.

2. Crear tablas para el ingreso de casos

Luego de haber modelado la base de datos de la aplicación con todas sus tablas y relaciones, se procede a realizar la creación de todas las tablas, estas incluyen:

- Caso: contiene el caso que se va a presentar con un identificador, fecha de creación y usuario que lo creo.
- Usuario: esta tabla almacena la información de los usuarios incluyendo el usuario y la contraseña. El rol puede ser de profesor, estudiante o administrador.
- Riesgo: la tabla contiene los riesgos con su descripción y el impacto y probabilidad predeterminados en el caso de estudio.
- Curso: Cada estudiante pertenece a un curso que a su vez es manejado por un profesor. Esta tabla se creó con el propósito de brindarle escalabilidad al proyecto.
- Partida: Cada vez que el estudiante participa y resuelve un caso se debe generar una nueva partida.

De la misma forma, se hicieron las tablas que manejan las relaciones estas son:

- Clasificación_Riesgo: guarda la clasificación/ valoración que cada estudiante le da al riesgo.

- **Detalle_Partida:** esta tabla almacena qué riesgos han sido seleccionados para ser mitigados.

3. Registrar el caso de estudio

Como no es parte de las tareas del Sprint desarrollar un módulo de administración y configuración, se ingresa el caso de estudio de gestión de riesgos como se lo puede observar en la Figura 4.3.

	idcaso	caso_descripcion	caso_fecha	idusuario_caso
✎	1	Los servicios públicos locales	12/01/2016	2
✳	NULL	NULL	NULL	NULL

Figura 4.3 Selección de casos de estudio de tabla casos.

Fuente: Elaboración propia

En la figura se muestra que el caso de estudio utilizado en el presente proyecto se ha ingresado con éxito. Además, se ha asignado un número de identificación para el caso, la fecha de ingreso del caso y la identificación del usuario que creó el caso. Con esta parte, se cumple con la primera historia de usuario del proyecto.

4. Registrar riesgos

Después de haber ingresado el caso, se ingresan también los riesgos asignados al caso. Esto se muestra a continuación en la Figura 4.4

idriesgo	riesgo_descripcion	riesgo_impacto	riesgo_probabilidad	riesgo_costo	idcaso_riesgo	riesgo_valoracion
1	Poca definición de los requerimientos.	Alto	Alto	2000	1	Alto
2	Es posible que no se hayan identificado interfac...	Muy Alto	Muy Alto	2000	1	Muy Alto
3	El diseño de las pruebas podría hacerse no muy ...	Alto	Medio	12000	1	Alto
4	Las especificaciones podrían no estar a suficien...	Alto	Alto	0	1	Alto
5	La integración del producto no estaría bien espe...	Muy Alto	Muy Alto	15000	1	Muy Alto
6	Posiblemente no haya especificaciones de requ...	Alto	Bajo	0	1	Alto
7	Posiblemente no se haya especificado los plane...	Medio	Alto	3000	1	Alto
8	Los desarrolladores no podrían estar familiariza...	Alto	Medio	0	1	Alto
9	Posiblemente no se maneje bien el control de ca...	Medio	Alto	0	1	Alto
10	Poca familiaridad con Eclipse y JUnit por parte d...	Medio	Alto	10000	1	Alto
11	Demoras en el tiempo de desarrollo debido a qu...	Alto	Alto	20000	1	Alto
12	El cronograma no podría ser realista, posibleme...	Alto	Alto	2000	1	Alto
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 4.4 Tabla de Riesgos en la Base de Datos

Fuente: Elaboración propia

Tras haber ingresado estos riesgos y el caso se cumplen con los puntos 1-4 de la lista de tareas del Sprint 1. Luego, se procede a trabajar con las tareas 5, 6 que constituyen la historia de usuario de acceder con usuario y clave.

5. Guardar usuario

Los usuarios que se ingresaron en la base de datos y que sirvan para el ingreso a la aplicación web se muestra en seguida.

idusuario	usuario_nombre	usuario_rol	usuario_user	usuario_pass	user_mail
1	gabipalacios	estudiante	gabipalacios	mariagabriela	gabipalacios@uees.edu.ec
2	gabi	admin	gabi	gabi	mgpalacios@yoveri.com.ec
3	Fernando Uyaguari	admin	fuyaguari	fuyaguari	NULL

Figura 4.5 Tabla de Usuario en la Base de Datos

Fuente: Elaboración propia

Luego de guardar los usuarios se procede a desarrollar el ingreso de usuarios de la aplicación. En ella el usuario ingresa con su nombre de usuario y contraseña.



Figura 4.6 Pantalla de Inicio de Sesión

Fuente: Elaboración propia

6. Validar Usuario

La última tarea de este Sprint es la validación de usuario. Esta se realizó por medio de Java Servlet, que realizó una consulta a la base de datos para comprobar que el usuario y contraseña pertenecen a un usuario de la base de datos. En el caso que la entrada sea exitosa, se dirige hacia la siguiente pantalla.



Figura 4.7 Pantalla de Introducción

Fuente: Elaboración propia

Por otro lado, en el caso que el usuario esté incorrecto se muestra el siguiente mensaje en la pantalla.



Figura 4.8 Pantalla de Inicio de Sesión con error

Fuente: Elaboración propia

Con esta validación se completan las seis tareas de la lista de tareas del Sprint. A continuación se debe realizar la revisión del Sprint.

4.2.1.3 Revisión del Sprint 1

Durante el *sprint 1*, se desarrollaron las tareas relacionadas al ingreso del caso de estudio y la validación de usuarios en la pantalla en la siguiente tabla se verifica que todo lo planteado haya sido realizado

Tabla 4.2.3
Tareas realizadas en el sprint 1

Lista de Tareas Sprint 1	Realizado
1. Modelar la base de datos	✓
2. Crear tablas para ingresar el caso	✓
3. Registrar el caso de estudio	✓
4. Registrar riesgos en la base de datos	✓
5. Guardar usuario	✓
6. Validar usuario	✓

Fuente: Elaboración propia

Tomando en cuenta que el objetivo general del presente trabajo de titulación es implementar una aplicación utilizando la técnica de Gamification, los elementos de diseño implementado en este Sprint, se detallan a continuación:

Tabla 4.2.4
Elementos de Gamification implementados

Elemento de diseño	Elemento de la Aplicación
<i>Personalización</i>	Uso de usuarios para desarrollar casos.
<i>Propósito</i>	Explicación sobre gestión de riesgos: explicación de en qué consiste y cuál es el objetivo

Fuente: Elaboración propia

Por otro lado, la siguiente Tabla 4.2.5 resume los tiempos de desarrollo del Sprint 1.

Tabla 4.2.5
Resumen de Horas de Desarrollo Sprint 1

	<i>Lista de Tareas</i> <i>Pendientes</i>	<i>Lista de Tareas</i> <i>Entradas</i>	<i>Horas</i>
Día 1	6	0	3h
Día 2	6	0	1h
Día 3	6	0	1h
Día 4	6	0	1h
Día 5	5	1	2h
Día 6	5	1	0h
Día 7	5	1	0h
Día 8	5	1	1h
Día 9	4	2	1h
Día 10	3	3	1h
Día 11	3	3	2h
Día 12	2	4	2h
Día 13	2	4	0h
Día 14	2	4	0h
Día 15	2	4	3h
Día 16	2	4	2h
Día 18	2	4	3h
Día 19	1	5	2h
Día 20	1	5	3h
Día 21	0	6	3h
Total	0	6	31h

Fuente: Elaboración propia

Tomando en cuenta los tiempos de desarrollo del *Sprint* y la planificación inicial del *Sprint*, se realiza un cuadro comparativo del listado de tareas pendientes; este se muestra enseguida.

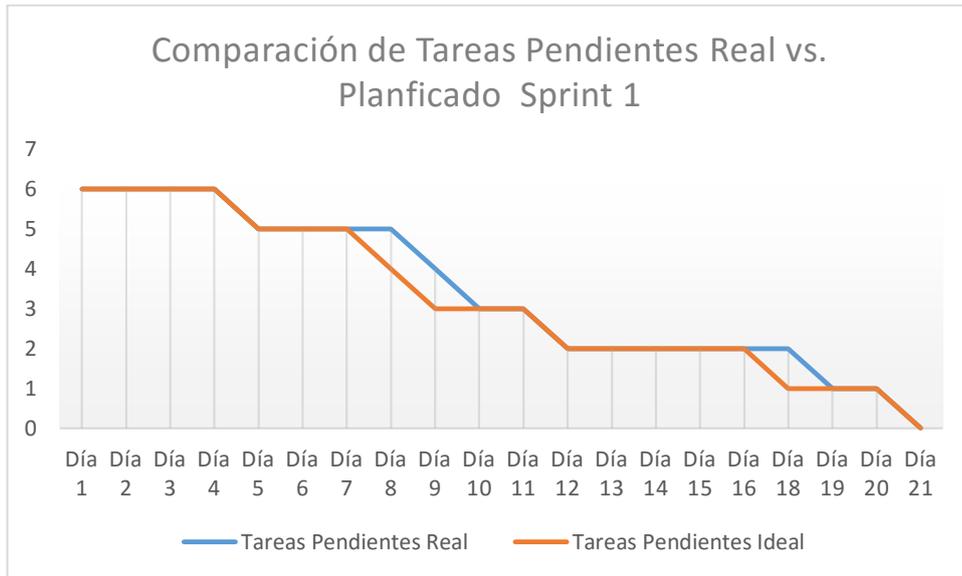


Figura 4.9 Comparación de Listado de Tareas Pendientes Sprint 1

Fuente: Elaboración propia

Además, se realiza una comparación de los tiempos designados al desarrollo de cada tarea. Estos, se muestran a continuación.

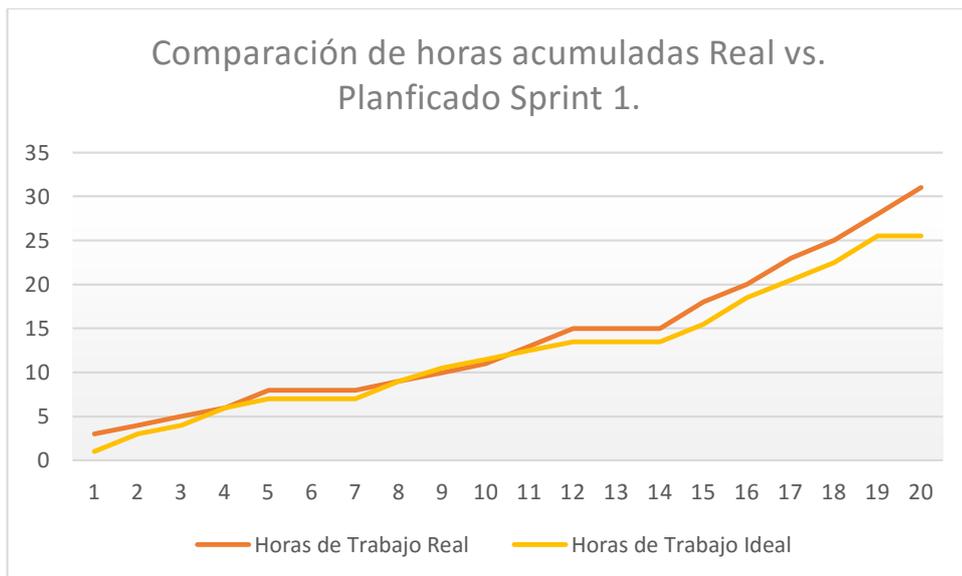


Figura 4.10 Comparación de horas acumuladas Sprint 1

Fuente: Elaboración propia

Como se puede observar en la Figura 4.10 se necesitaron de 5 horas más de las planificadas para terminar a tiempo con todas las tareas inicialmente asignadas a este Sprint.

4.2.1.4 Retrospectiva del Sprint 1

El sprint 1 fue exitoso ya que se cumplieron con las metas planteadas. En él se completaron dos de las seis historias de usuario definidas. Entre las cosas que se deben tener en cuenta durante los siguientes dos sprints están las siguientes:

- Mantener una interface amigable para el usuario final
- Implementar los elementos de Gamification
- Mantener el enfoque de la aplicación hacia la enseñanza de gestión de riesgos.

4.2.2 Sprint 2

4.2.2.1 Planificación del Sprint 2

El Sprint 2 está enfocado en tener una aplicación que pueda realizar el análisis cualitativo del riesgo; y obtenga como resultado la valoración del mismo. Con el objetivo de cumplir este propósito se definen las tareas que van a ser parte del Sprint

Tabla 4.2.6
Lista de Tareas Sprint 2

Lista de Tareas Sprint 2	Esfuerzo
1. Iniciar resolución de caso	2,5
2. Elegir impacto del riesgo	8
3. Elegir probabilidad del riesgo	3,5
4. Finalizar etapa de valoración del riesgo	8
Total	22

Fuente: Elaboración propia

A partir de estas tareas se realiza una planificación del trabajo estimado en horas necesarias para completar los objetivos. La planificación del trabajo se muestra a continuación:

Tabla 4.2.7

Planificación del Trabajo Sprint 2

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Total
Día 1	0,5h	-	-	-	0,5h
Día 2	-	-	-	-	-
Día 3	-	-	-	-	-
Día 4	1h	-	-	-	1h
Día 5	1h	-	-	-	1h
Día 6	-	-	-	-	-
Día 7	-	-	-	-	-
Día 8	-	1h	-	-	1h
Día 9	-	1h	-	-	1h
Día 10	-	2h	-	-	2h
Día 11	-	2h	-	-	2h
Día 12	-	2h	1h	-	3h
Día 13	-	-	1h	-	1h
Día 14	-	-	1,5h	-	1,5h
Día 15	-	-	-	-	-
Día 16	-	-	-	-	-
Día 18	-	-	-	2h	2h
Día 19	-	-	-	2h	2h
Día 20	-	-	-	2h	2h
Día 21	-	-	-	2h	2h
Total	2,5h	8h	3,5h	8h	22h00

En la planificación de horas se busca asignar la mayor cantidad de tiempo a las 3 tareas más importantes: elegir impacto o probabilidad y concluir con la etapa de valoración del riesgo. En este Sprint se va a cubrir con la mayor parte del alcance de la aplicación ya que, se va a lograr

identificar y analizar los riesgos que forman parte de la gestión de un proyecto.

4.2.2.2 Desarrollo del Sprint 2

El Sprint 2 está basado en las tareas detalladas anteriormente en la tabla 4.2.6 La primera tarea que se va a llevar a cabo es el inicio de la resolución del caso.

1. Inicio de la resolución del caso.

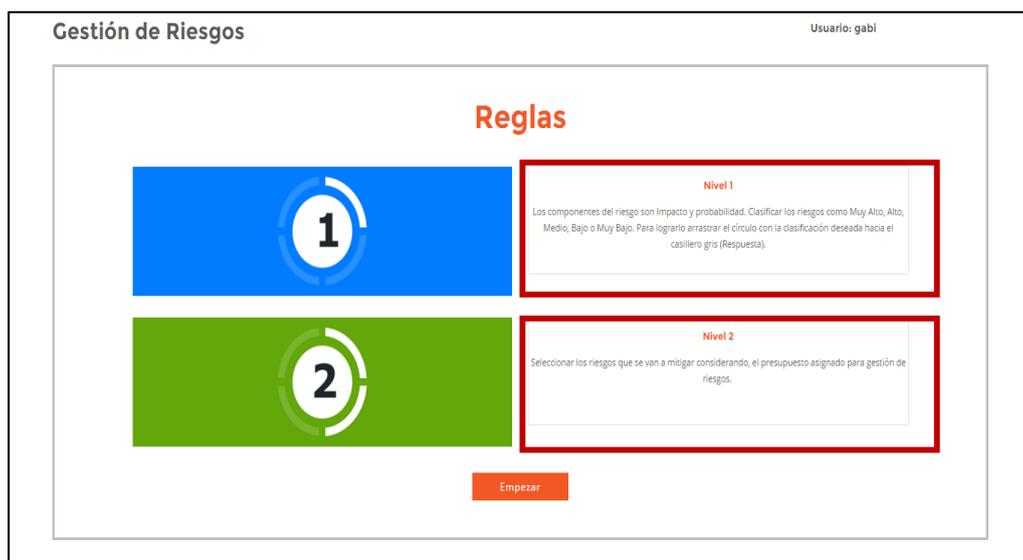


Figura 4.11 Pantalla de Inicio de Resolución de Caso

Fuente: Elaboración propia

Tal como se puede observar en la Figura 4.11, antes de iniciar la solución del caso se explican las reglas de las actividades que se van a realizar. En el nivel 1, se detalla que los componentes del riesgo son Impacto y probabilidad. Luego, se indica que el usuario debe clasificar los riesgos como: Muy Alto, Alto, Medio, Bajo o Muy Bajo. Por otro lado, en el nivel 2 se explica que el usuario debe seleccionar los riesgos que se van a mitigar considerando, el presupuesto asignado para gestión de riesgos.

Al presionar Empezar, el usuario genera una nueva partida, tomando el curso y el caso asignado al mismo. También se guarda la fecha de inicio de partida y el usuario que empezó la resolución del caso.

2. Elegir impacto y probabilidad del Riesgo

Para completar las dos actividades, se diseñó una sola pantalla. En ella, se puede observar de nuevo el caso y también permite definir el impacto y la probabilidad de los riesgos en el proyecto. Este nivel cuenta con los riesgos asignados previamente en la Tabla 4.1.7. Una captura de pantalla se muestra a continuación.



Figura 4.12 Pantalla Categorización de Riesgos

Fuente: Elaboración propia

Para categorizar los riesgos, el usuario utiliza la funcionalidad de arrastras y soltar. En el nivel, el usuario arrastra los círculos con los distintos valores de probabilidad o impacto hacia el casillero de respuesta. Una vez seleccionado los valores, selecciona siguiente y realiza la misma actividad hasta completar con todos los riesgos que han sido asignados al caso de estudio. Cuando termina de categorizar todos los riesgos, se guardan los valores ingresados en la base de datos y se continúa con la

retroalimentación del nivel 1. A continuación se muestran capturas de pantalla con el proceso de selección de impacto y probabilidad del riesgo.

Elegir probabilidad



Figura 4.13 Captura de pantalla al arrastrar valor de probabilidad

Fuente: Elaboración propia



Figura 4.14 Captura de pantalla al soltar valor de probabilidad

Fuente: Elaboración propia

Elegir impacto



Figura 4.15 Captura de pantalla al arrastrar el valor del impacto

Fuente: Elaboración propia



Figura 4.16 Captura de pantalla al soltar valor de probabilidad

Fuente: Elaboración propia

3. Finalizar etapa de valoración del riesgo

Luego de haber categorizado todos los riesgos, el usuario accede a una pantalla que muestra la valoración de cada riesgo que analizó. En esta pantalla, recibe una retroalimentación de cuál era el resultado esperado de este nivel. Al usuario se lo evalúa dependiendo de qué tan cercano estuvo a la respuesta ideal. En el caso que la valoración del riesgo sea igual a la valoración esperada del mismo, el usuario recibe 10 puntos. Por otro lado, si es una valoración cercana recibe 5 puntos. En el caso que la valoración del usuario esté completamente alejada de la valoración ideal recibe 0 puntos por el riesgo. Por otro lado, en esta tarea también se da una puntuación al nivel que se muestra en el cuadro rojo en la Figura 4.17.

Esta pantalla se muestra a continuación:

Puntaje total:75

Resultados: Nivel 1				
	Descrpción del Riesgo	Puntuación	Valoración	Respuesta Correcta
1	Poca definición de los requerimientos.	5	Muy Alto	Alto
2	Es posible que no se hayan identificado interfaces con otros sistemas.	10	Muy Alto	Muy Alto
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.	5	Muy Alto	Alto
4	Las especificaciones podrían no estar a suficiente detalle para escribir el código.	5	Muy Alto	Alto
	La integración del producto no estaría bien especificada. Lo relativo a los			Muy

Figura 4.17 Captura de Pantalla de Respuestas Nivel 1

Fuente: Elaboración propia

Al terminar esta funcionalidad, se completan las tareas del Sprint 2. Es decir, se obtiene un producto que puede identificar y analizar riesgos. Luego de haber realizado estas actividades, se presiona siguiente para empezar con la etapa de mitigación de riesgos que corresponde al Sprint 3.

4.2.2.3 Revisión del Sprint 2

Durante el sprint 2, se desarrollaron las tareas relacionadas a la categorización de los riesgos asignados al caso de estudio. Para esto se lo evalúa tomando en cuenta el impacto y la probabilidad del mismo. Luego, el usuario recibe retroalimentación de la valoración que ha realizado. Para comprobar que las tareas han sido realizadas se muestra la tabla a continuación.

Tabla 4.2.8
Lista de Tareas Realizadas Sprint 2

Lista de Tareas Sprint 2	Realizado
1. Iniciar resolución de caso	✓
2. Elegir impacto del riesgo	✓
3. Elegir probabilidad del riesgo	✓
4. Finalizar etapa de valoración del riesgo	✓

Fuente: Elaboración propia

Tomando en cuenta que el objetivo general del presente trabajo de titulación es implementar una aplicación utilizando la técnica de Gamification, los principios de diseño implementado en este Sprint 2, se detallan a continuación:

Tabla 4.2.9
Elementos de Gamification implementados

Elemento de diseño	Elemento de la Aplicación
Metas	La meta del usuario es definir la valorización del riesgo
Desafíos	Los desafíos son los dos niveles de gestión de riesgos: análisis y mitigación de riesgos.
Retroalimentación	Al finalizar el nivel se muestra una comparación entre la valorización ideal y la valorización esperada.
Progreso	El progreso se muestra a través de los puntos que obtiene al haber terminado el nivel

Fuente: Elaboración propia

Por otro lado, la siguiente Tabla 4.2.10 resume los tiempos de desarrollo del Sprint 2.

Tabla 4.2.10
Resumen de Horas de Desarrollo Sprint 2

	<i>Lista de Tareas</i> <i>Pendientes</i>	<i>Lista de Tareas</i> <i>Entradas</i>	<i>Horas</i>
Día 1	4	0	0h
Día 2	4	0	0h
Día 3	4	0	0h
Día 4	4	0	0h
Día 5	4	0	0h
Día 6	3	1	3h
Día 7	3	1	0h
Día 8	3	1	2h
Día 9	3	1	2h
Día 10	3	1	5h
Día 11	3	1	4h
Día 12	3	1	1h
Día 13	3	1	0h
Día 14	3	1	0h
Día 15	3	1	1h
Día 16	3	1	3h
Día 18	1	3	2h
Día 19	1	3	2h
Día 20	1	3	2h
Día 21	0	4	3h
Total	0	4	30h

Fuente: Elaboración propia

Al igual que en el Sprint 1, tomando en cuenta los tiempos de desarrollo del Sprint 2 y la planificación inicial del mismo, se realizan unos cuadros

comparativo del listado de tareas pendientes así como también de las horas dedicadas a completar el Sprint 2; estos se muestra en seguida.

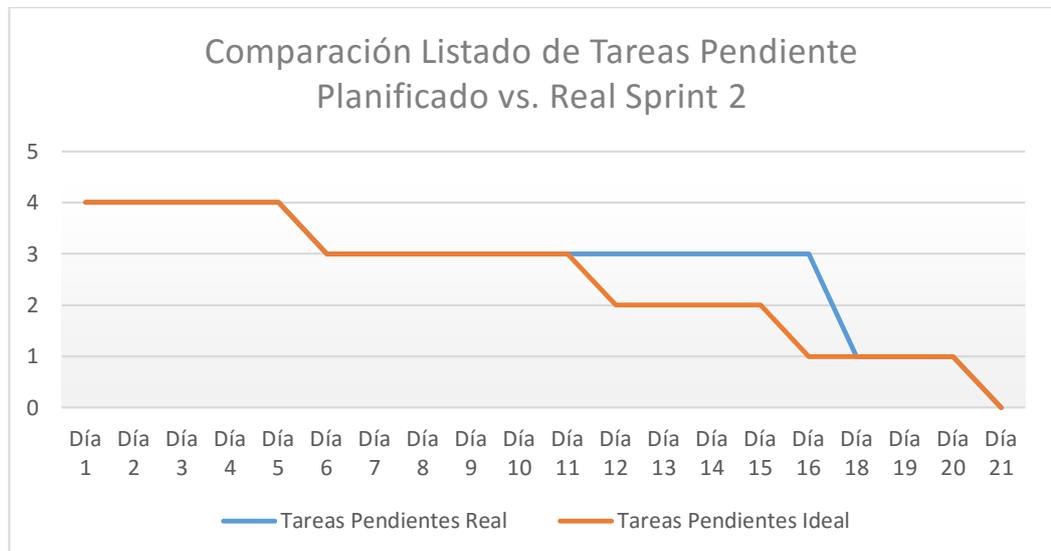


Figura 4.18 Comparación Listado Tareas Pendientes Sprint 2

Fuente: Elaboración propia

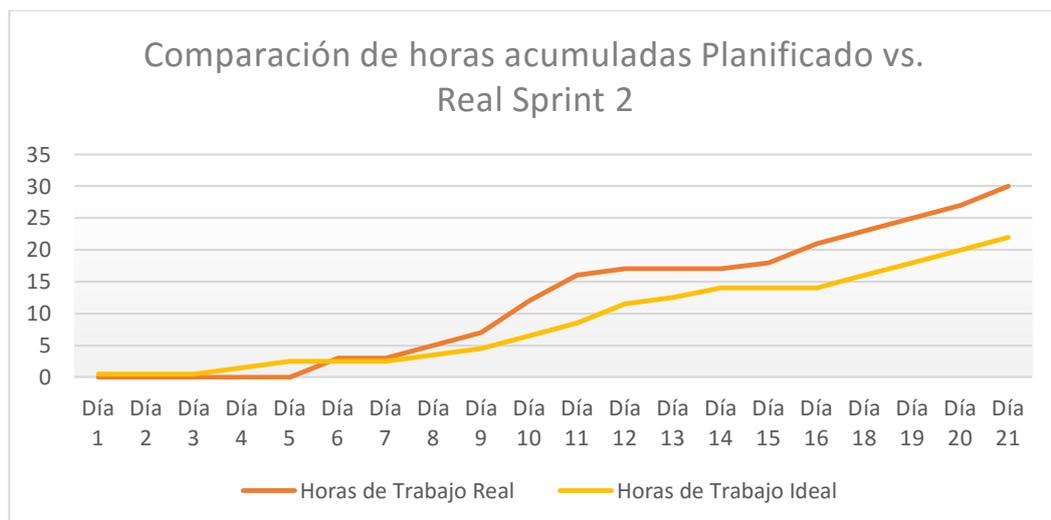


Figura 4.19 Comparación de horas acumuladas Sprint 2

Fuente: Elaboración propia

Como se puede ver en ambas ilustraciones, 4.18 y 4.19, la planificación del Sprint 2 no fue tan acertada como en el Sprint 1. En ambos casos se

logró cumplir con los objetivos. Sin embargo, en el Sprint 2 bajó el rendimiento y se necesitaron de 8 horas más que las planificadas para cumplir con las tareas.

4.2.2.4 Retrospectiva del Sprint 2

El sprint 2 se completó en el tiempo planificado y se realizaron todas las tareas de la Lista. Sin embargo, la planificación de los tiempos no fue exitosa. En consecuencia, para el siguiente Sprint 3 se sugiere:

- Mejorar la estimación de esfuerzo
- Motivar el desarrollo de las tareas del Sprint
- Mantener una interfaz amigable
- Siempre retroalimentar al usuario

4.2.3 Sprint 3

4.2.3.1 Planificación del Sprint 3

El Sprint 3 está enfocado en tener una aplicación que pueda realizar la mitigación del riesgo; y obtenga como resultado cantidad de riesgos que van a ser mitigados bajo el presupuesto que se determina para el proyecto de software. A continuación, con el objetivo de cumplir este propósito se definen las tareas que van a ser parte del Sprint 3:

Tabla 4.2.11
Lista de Tareas Sprint 3

Lista de Tareas Sprint 3	Esfuerzo
1. Seleccionar riesgos a mitigar	9
2. Elegir presupuesto	3
3. Comparar costo de mitigación vs. Presupuesto asignado a riesgos	6
4. Visualizar la cantidad de riesgos mitigados	2
5. Mostrar puntuaciones finales	6
Total	26

Fuente: Elaboración propia

A partir de estas tareas se realiza una planificación del trabajo en horas necesarias para llevarlas a cabo. La planificación del trabajo se muestra a continuación:

Tabla 4.2.12

Planificación del Trabajo Sprint 3

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Total
Día 1	1h	-	-	-	-	1h
Día 2	1h	-	-	-	-	1h
Día 3	3h	-	-	-	-	3h
Día 4	2h	-	-	-	-	2h
Día 5	2h	-	-	-	-	2h
Día 6	-	-	-	-	-	-
Día 7	-	-	-	-	-	-
Día 8	-	1h	-	-	-	1h
Día 9	-	1h	-	-	-	1h
Día 10	-	1h	-	-	-	1h
Día 11	-	-	-	-	-	-
Día 12	-	-	2h	-	-	2h
Día 13	-	-	3h	-	-	3h
Día 14	-	-	1h	-	-	1h
Día 15	-	-	-	-	-	-
Día 16	-	-	-	-	-	-
Día 18	-	-	-	1h	-	1h
Día 19	-	-	-	1h	-	1h
Día 20	-	-	-	-	3h	3h
Día 21	-	-	-	-	3h	3h
Total	9h	3h	6h	2h	6h	26h00

Fuente: Elaboración propia

En este Sprint 3, las tareas más importantes están relacionadas a elegir cuáles son los riesgos que van a ser mitigados durante el Sprint. Sin embargo, para poder completar el objetivo, también se debe tener en cuenta cuál es el presupuesto que se ha asignado para realizar la mitigación de riesgos.

4.2.3.2 Desarrollo del Sprint 3

El Sprint 3 está basado en las tareas detalladas anteriormente en la Tabla 4.2.11. La primera tarea que se va a llevar a cabo seleccionar riesgos a mitigar.

1. Seleccionar riesgos a mitigar.

Para seleccionar los riesgos que se van a mitigar se utilizaron dos tablas. La primera con todos los riesgos del caso, con el costo de mitigar cada uno de ellos y la segunda tabla con el propósito de arrastrar los riesgos que se deseen seleccionar. En las siguientes ilustraciones, 4.20 4.21 se muestra la funcionalidad.

Nivel 2: Riesgos			
ID Riesgo	Descripción del Riesgo	Mecanismo de Mitigación	Costo
1	Poca definición de los requerimientos.	Identificar las interfaces con otros sistemas y sugerir considerar los costos dentro del total del proyecto.	\$2,000.0 USD
2	Es posible que no se hayan identificado interfaces con otros sistemas.	Realizar un análisis detallado de la migración de los datos. Presupuestar y asignar recursos a las actividades relacionadas.	\$15,000.0 USD
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.	Realizar un proceso de validación de los documentos de requerimientos	\$2,000.0 USD
4	Las especificaciones podrían no estar a suficiente detalle para escribir el código.	El mecanismo que se aplica a este riesgo es el mismo del riesgo 1.	\$0.0 USD
	La integración del producto no estaría bien	Implementar pair programming con TDD en el	

<< << Página 1 de 1 >> >>

Nivel 2: Riesgos seleccionados			
ID Riesgo	Descripción del Riesgo	Mecanismo de Mitigación	Costo
3	El diseño de las pruebas podría hacerse no muy rigurosamente (suficientes pruebas) al no hacerlo con una técnica adecuada.	Realizar un proceso de validación de los documentos de requerimientos	\$2,000.0 USD

Figura 4.20 Pantalla de Mitigación de Riesgos

Fuente: Elaboración propia

Nivel 2: Riesgos seleccionados		
Descripción del Riesgo	Mecanismo de Mitigación	Costo
Posiblemente no se maneje bien el control de cambios.	Se confía en la seguridad del software de base y en la red en donde va a funcionar el software. Se asume el riesgo.	\$0.0 USD
Los desarrolladores no podrían estar familiarizados con la técnica TDD, incluso luego de haber recibido capacitación.	Hacer pares de programación en donde haya una persona con habilidades para programar con TDD, se incluiría en el mecanismo para mitigar el riesgo 11.	\$0.0 USD

Página 1 de 0

Figura 4.21 Función arrastrar riesgo en Pantalla de Mitigación

Fuente: Elaboración propia

2. Elegir el presupuesto para Mitigación

El proyecto tiene asignado un presupuesto de \$71,000, solo una parte de este presupuesto está destinado para la gestión de riesgos. En la aplicación se puede configurar por medio de una lista qué porcentaje del presupuesto va a estar destinado para administrar los riesgos. Esta funcionalidad se muestra en seguida:

Nivel 2: Mitigación de Riesgos

Presupuesto del Proyecto: \$71,000

Porcentaje para gestión de riesgos:

Costo total: \$3000.00

Nivel 2: Riesgos	Descripción	Mecanismo de Mitigación
	pacientes.	... los "era" parámetros en caso de que sea o.
	Posiblemente no se haya todas las actividades d para la fase requerimient	... har una técnica de testing (Particiones de ncia) y elaborar los casos de prueba con la seleccionada. En caso de no tener lento de las técnicas contratar una persona eriencia en pruebas por el tiempo de las pruebas.

Figura 4.22 Mitigación de Riesgo, selección de porcentaje.

Fuente: Elaboración propia

3. Comparar costo de mitigación vs. Presupuesto asignado a riesgos

En la aplicación, el usuario debe tener la posibilidad de controlar cuánto del presupuesto asignado para la gestión de riesgos está utilizando con los riesgos seleccionados. Para lograrlo se utiliza una barra que mide el presupuesto; esta barra se muestra a continuación.



Figura 4.23 Mitigación de Presupuesto Barra de Control

Fuente: Elaboración propia

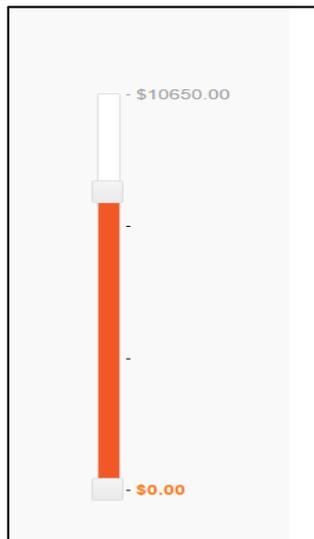


Figura 4.24 Barra de Control de Presupuesto de Riesgos

Fuente: Elaboración propia

4. Visualizar cantidad de riesgos mitigados

Al igual que en el Nivel 2, al finalizar la selección de riesgos mitigados se muestra un resumen del nivel. En él se muestran el presupuesto asignado para riesgos, así como también el costo total de los riesgos escogidos y el total de riesgos. A continuación se muestra la funcionalidad en la Figura 4.25.

Resultados: Nivel 2

Resumen: Nivel 2		
Presupuesto para Riesgos	Costo Mitigación de Riesgos	Total de Riesgos Mitigados
71000	10000	6

Figura 4.25 Resultados del Nivel 2

Fuente: Elaboración propia

5. Mostrar Puntuaciones Finales

Para finalizar las tareas del Sprint 3, se debe mostrar un resumen de todos los resultados del usuario y de otro usuario. En esta tabla, se muestran los puntajes de mayor a menos y la cantidad de riesgos que fueron mitigados durante el desarrollo del caso.

Caso de Estudio

Tabla de Resultados

Tabla de Resultados		
Usuario	Puntaje	Riesgos Mitigados
gabi	75	5
gabi	0	5

Figura 4.26 Resultados de Resolución de Caso

Fuente: Elaboración propia

4.2.3.3 Revisión del Sprint 3

Durante el sprint 3, se desarrollaron las tareas relacionadas a la mitigación de los riesgos asignados al caso de estudio. Para esto se eligen los riesgos que se desean mitigar. Luego, el usuario recibe retroalimentación de la mitigación que ha realizado. Para comprobar que las tareas han sido realizadas.

Tabla 4.2.13
Lista de Tareas Realizadas Sprint 3

Lista de Tareas Sprint 3	Realizado
1. <i>Seleccionar riesgos a mitigar</i>	✓
2. <i>Elegir presupuesto</i>	✓
3. <i>Comparar costo de mitigación vs. Presupuesto asignado a riesgos</i>	✓
4. <i>Visualizar la cantidad de riesgos mitigados</i>	✓
5. <i>Mostrar puntuaciones finales</i>	✓

Fuente: Elaboración propia

Tomando en cuenta que el objetivo general del presente trabajo de titulación es implementar una aplicación utilizando la técnica de Gamification, los principios de diseño implementado en este Sprint 3, se detallan a continuación:

Tabla 4.2.14

Elementos de Gamification implementados

Elemento de diseño	Elemento de la Aplicación
<i>Metas</i>	Se establece la meta del nivel 2: Mitigar riesgos.
<i>Desafíos</i>	La dificultad del nivel es que el usuario debe mitigar la mayor cantidad de riesgos con el presupuesto asignado para el proyecto.
<i>Progreso</i>	El progreso de los riesgos seleccionados se muestra con la barra de costo que se implementó en el nivel 2
<i>Retroalimentación</i>	El usuario recibe retroalimentación al finaliza el nivel. Aquí, él puede observar la cantidad de riesgos mitigados con el porcentaje del proyecto que fue asignado a la gestión de riesgos.
<i>Competencia</i>	Al finalizar de resolver el caso, el usuario puede observar el desempeño de los otros usuarios.

Fuente: Elaboración propia

Por otro lado, la siguiente Tabla 4.2.15 resume los tiempos de desarrollo del Sprint 3.

Tabla 4.2.15

Resumen de Horas de Desarrollo Sprint 3

	<i>Lista de Tareas</i>	<i>Lista de Tareas</i>	<i>Horas</i>
	<i>Pendientes</i>	<i>Entradas</i>	
<i>Día 1</i>	5	0	1h
<i>Día 2</i>	5	0	1h
<i>Día 3</i>	5	0	1h
<i>Día 4</i>	5	0	2h
<i>Día 5</i>	5	0	1h
<i>Día 6</i>	5	0	1h
<i>Día 7</i>	5	0	1h
<i>Día 8</i>	5	0	2h
<i>Día 9</i>	5	0	3h
<i>Día 10</i>	4	1	2h
<i>Día 11</i>	4	1	2h
<i>Día 12</i>	3	2	1h
<i>Día 13</i>	3	2	0h
<i>Día 14</i>	3	2	2h
<i>Día 15</i>	3	2	2h
<i>Día 16</i>	3	2	1h
<i>Día 18</i>	2	3	4h
<i>Día 19</i>	2	3	1h
<i>Día 20</i>	1	4	3h
<i>Día 21</i>	0	5	3h
<i>Total</i>	0	5	34h

Fuente: Elaboración propia

Al igual que en el Sprint 1 y 2, tomando en cuenta los tiempos de desarrollo del Sprint 3 y la planificación inicial del mismo, se realizan unos cuadros comparativo del listado de tareas pendientes así como también de las horas dedicadas a completar el Sprint 3; estos se muestra en seguida.

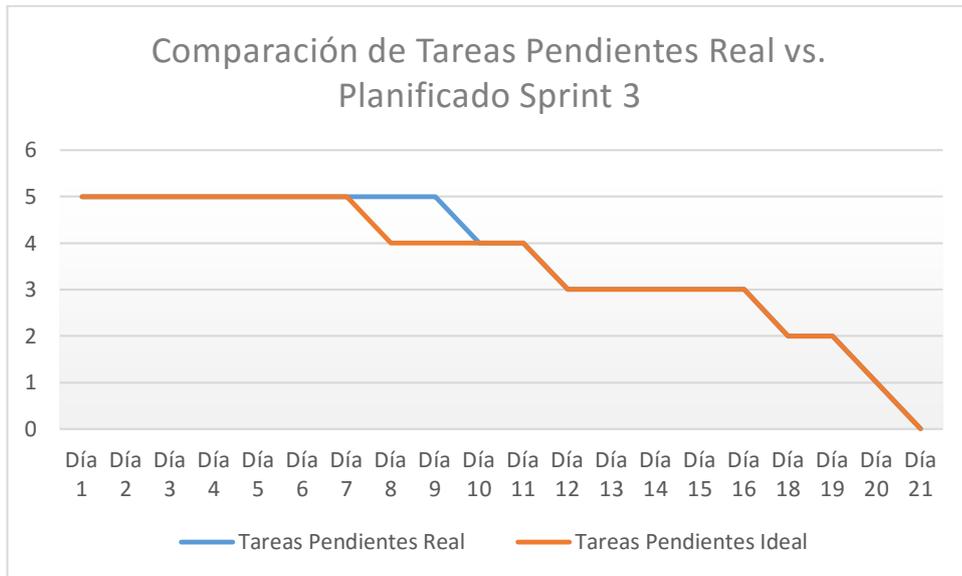


Figura 4.27 Comparación Listado de Tareas Pendientes Sprint 3

Fuente: Elaboración propia

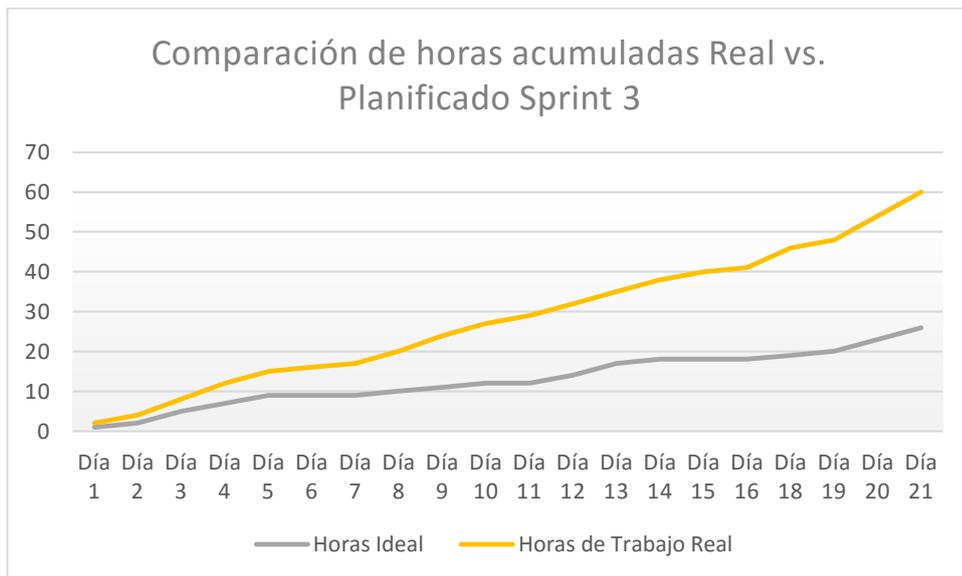


Figura 4.28 Comparación de horas acumuladas Sprint 3

Fuente: Elaboración propia

Como se puede observar en la Figura 4.28, los tiempos planificados para el Sprint 3 no se cumplieron a pesar que sí se cumplieron las tareas planificadas como se puede evidenciar en la Figura 4.27.

4.2.3.4 Retrospectiva del Sprint 2

El sprint 3 se completó en el tiempo planificado y se realizaron todas las tareas de la Lista. Sin embargo, la planificación de los tiempos no fue igual a la real. Hubo sólo un día en el que hubo 0 horas. Al finalizar este Sprint se acaba con la planificación realizada para completar el desarrollo de la aplicación.

4.3 Pruebas del Desarrollo

En el Capítulo 3, Metodología, se definió que las pruebas de la aplicación eran parte de la revisión del Sprint. En cada Sprint se realizaron pruebas funcionales para asegurar que se ejecuten adecuadamente las partes del sistema planteadas. Para ello se definieron escenarios de pruebas dependiendo del requerimiento del Sprint que se vaya evaluar.

4.3.1 Definición de pruebas

Al haber realizado pruebas evaluando la funcionalidad de la aplicación. Se eligieron 5 usuarios con las siguientes características:

Tabla 4.3.1
Características de Usuarios de Pruebas

Usuario	Edad	Carrera
1	20 años	Medicina
2	22 años	Ing. Civil
3	23 años	Ing. Marketing
4	47 años	Ing. Sistemas
5	32 años	Ing. Sistemas

Fuente: Elaboración propia

Estos usuarios fueron elegidos por sus características. Ellos realizaron pruebas que generan datos cualitativos del funcionamiento de la aplicación. Además, fueron escogidos con profesiones diferentes ya que Gamification puede ser implementada por más de un área de enseñanza. De la misma forma la Gestión de Riesgo también es aplicada por varias disciplinas. Los usuarios probaron la aplicación y verificaron que los

resultados eran los esperados. Para cumplirlo, utilizaron los escenarios de pruebas que se definen a continuación.

4.3.2 Sprint 1

Del Sprint 1 el requerimiento que va a formar parte de la etapa de pruebas funcionales es requerimiento 6: Validación de usuario. Los otros requerimientos forman parte de la configuración de la aplicación y el usuario final no tiene como probar el funcionamiento.

Requerimiento 6: Validación de Usuario

El escenario de prueba diseñado para validar este requerimiento se muestra a continuación:

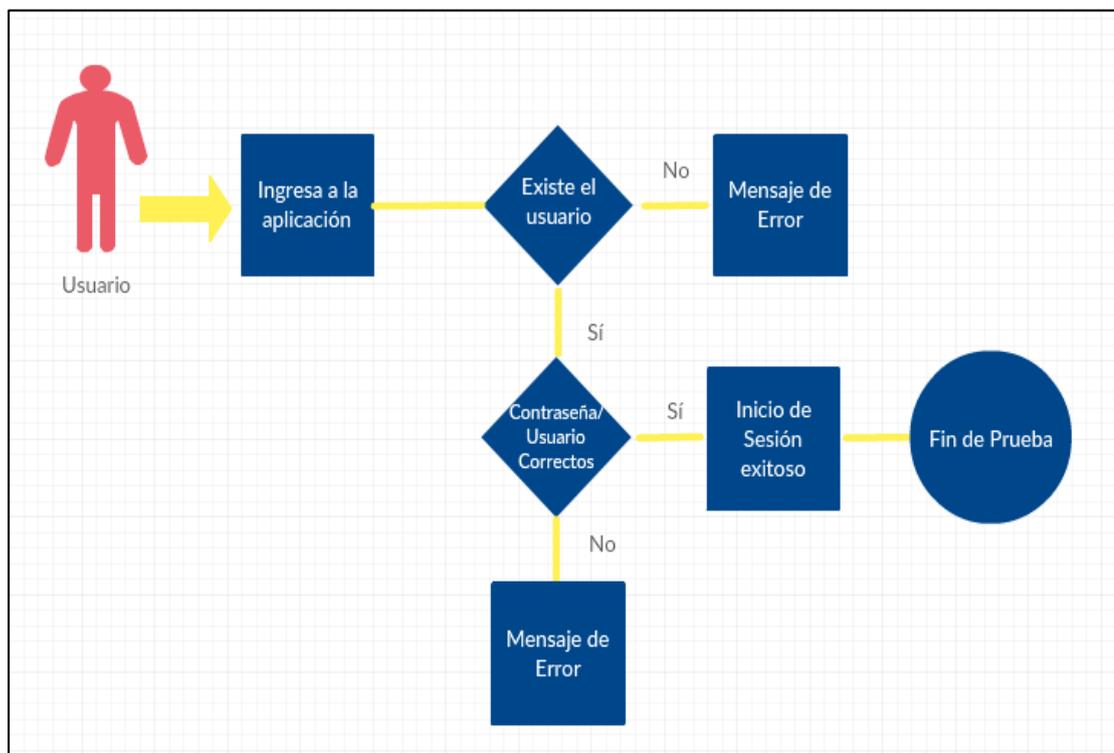


Figura 4.29 Escenario de prueba Requerimiento 6 Sprint 1

Fuente: Elaboración propia

En este escenario de prueba se evalúa el proceso de ingreso de usuarios. Si el usuario existe, entonces se valida que el usuario y

contraseña sean correctos. Si alguna de las validaciones no es exitosa, debería de mostrarse un mensaje de error. Si las validaciones son exitosas, el usuario puede ingresar a la aplicación.

4.3.3 Sprint 2

En el Sprint 2, los requerimientos considerados para las pruebas de usabilidad son el 2 y 3. Estos requerimientos son evaluados ya que involucran un mayor grado de interacción del usuario que los requerimientos 1 y 4 del Sprint 2. Con los requerimientos 1 y 4 se realizaron pruebas unitarias para evaluar asegurar que el funcionamiento sea efectivo.

4.3.3.1 *Requerimiento 2 y 3: Elegir Impacto y Probabilidad*

Los escenarios de prueba diseñados para validar estos requerimientos se muestran a continuación:

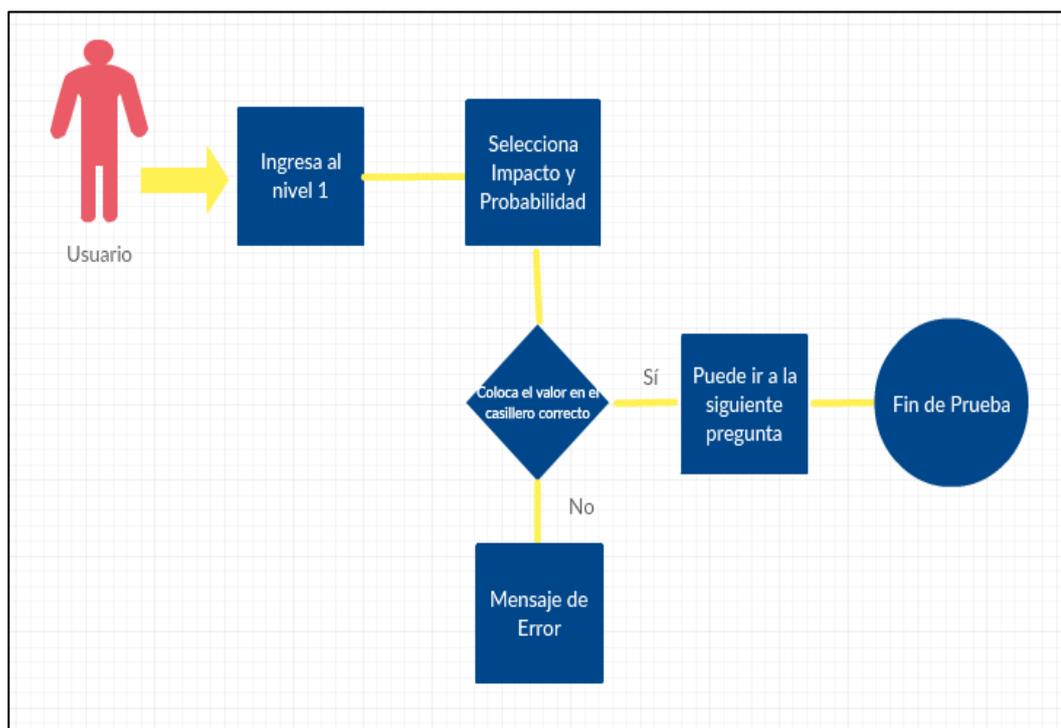


Figura 4.30 Escenario de prueba 1 Requerimiento 2 y 3 Sprint 2

Fuente: Elaboración propia

Este caso valida que las imágenes de los valores de impacto y probabilidad sean arrastradas y soltadas en el casillero de respuesta correspondiente. En el caso que no los sean, se deberá mostrar un mensaje de error.

Otro escenario de prueba asignado a estos requerimientos, permite validar que el usuario no pase a otra pregunta sin haber asignado una respuesta a los riesgos. Esto se muestra a continuación.

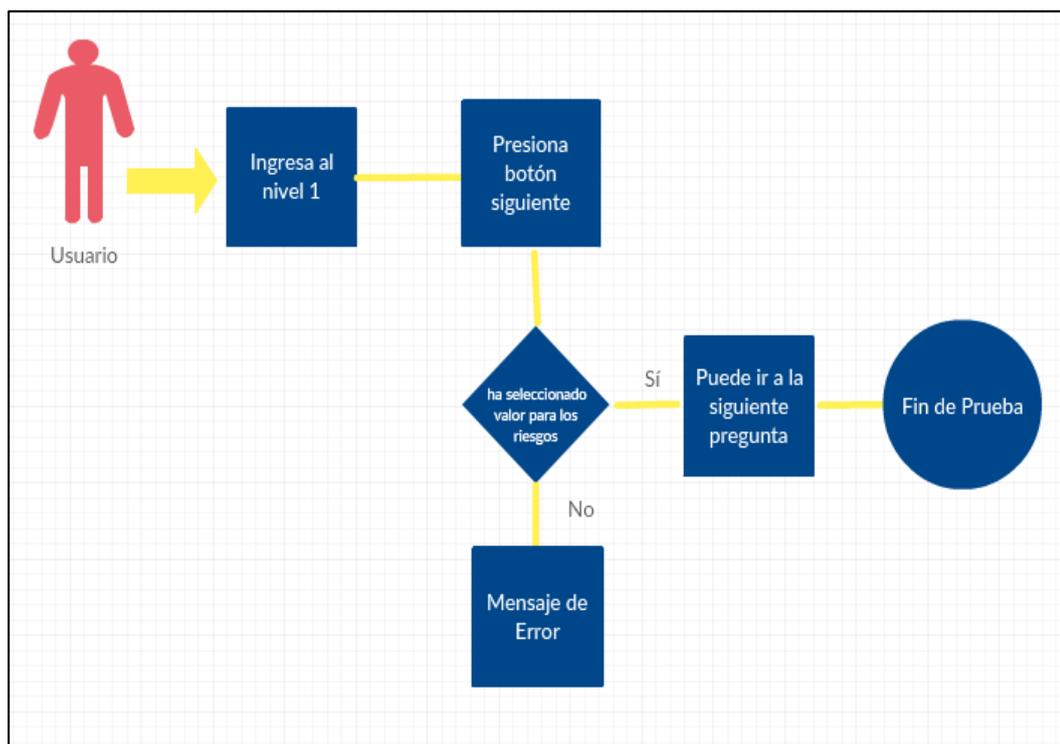


Figura 4.31 Escenario de prueba 2 Requerimientos 2 y 3 Sprint 2

Fuente: Elaboración propia

4.3.4 Sprint 3

En el Sprint 3, los requerimientos considerados en las pruebas funcionales son el requerimiento 1, Seleccionar Riesgos y el requerimiento 3, Comparar costo de mitigación vs. Presupuesto asignado a riesgos. Los

otros tres requerimientos del Sprint 3 fueron evaluados a través de pruebas unitarias. En ellas se verificó que se pueda elegir el presupuesto, se muestren la cantidad de riesgos mitigados y se visualicen las puntuaciones de los otros usuarios.

4.3.4.1 Requerimiento 1: Seleccionar riesgos

A continuación, se muestra el escenario de prueba asignado a evaluar este requerimiento:

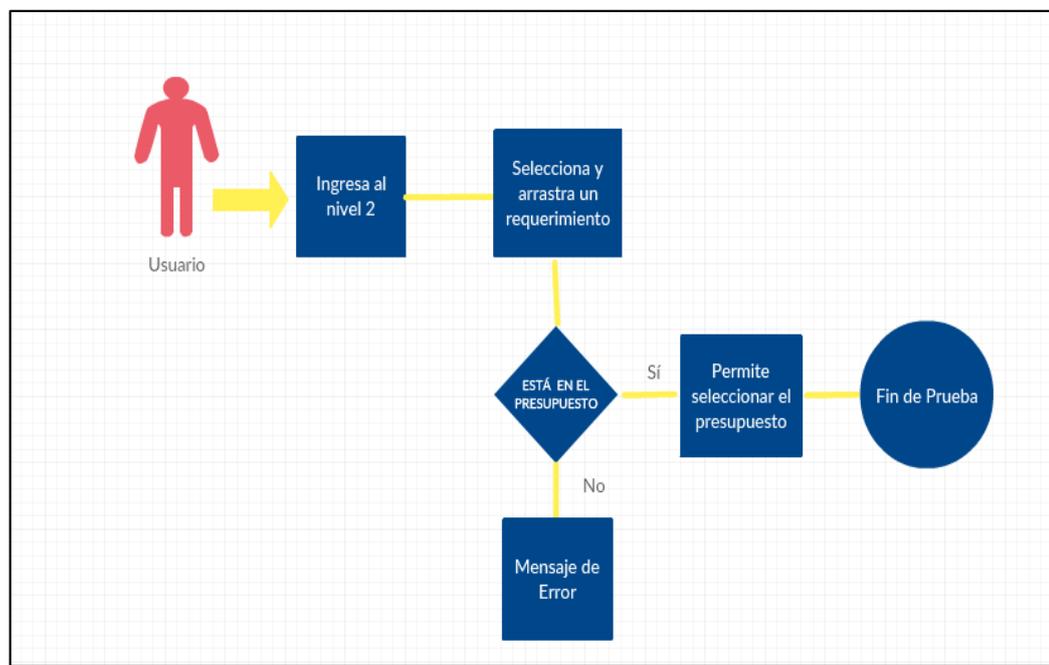


Figura 29.32 Escenario de prueba 1 Requerimiento 1 Sprint 3

Fuente: Elaboración propia

Este escenario de prueba evalúa que solo se puedan seleccionar riesgos siempre y cuando, estén dentro del presupuesto permitido. En el caso que haya sobrepasado el presupuesto, se debería mostrar un mensaje.

4.3.4.2 Comparar costo de mitigación vs. Presupuesto asignado a riesgos

Este requerimiento, prueba que la barra de presupuesto funciones adecuadamente. Es decir, que indique el costo total de mitigar los riesgos seleccionados.

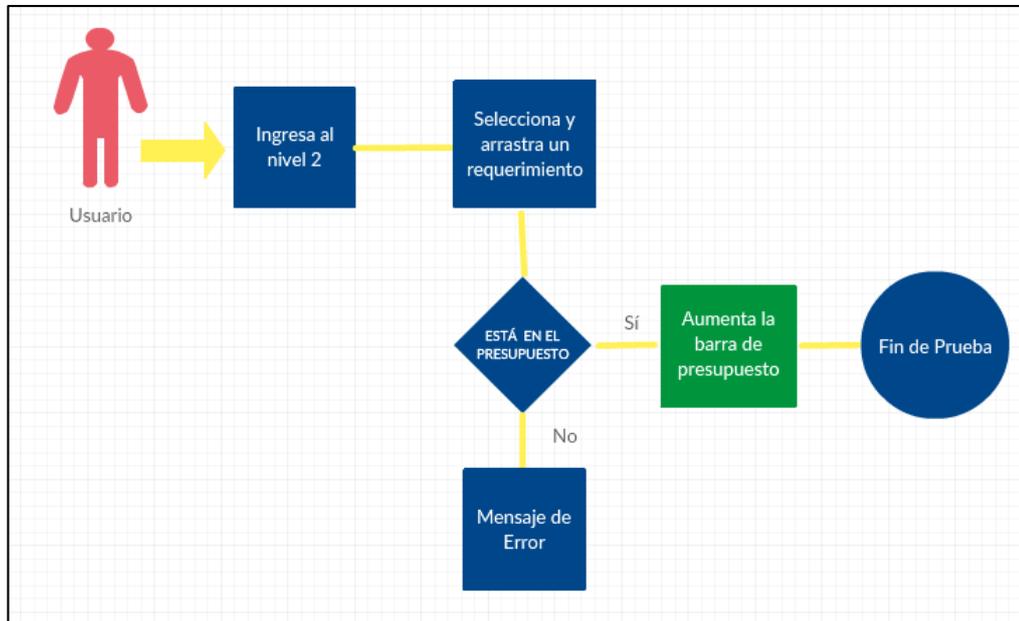


Figura 30.32 Escenario de prueba Requerimiento 3 Sprint 3

Fuente: Elaboración propia

4.3.5 Resultado de Pruebas

Los resultados obtenidos a partir de los escenarios de prueba definidos en la sección 5.1 se muestran a continuación.

4.3.6 Sprint 1

4.3.6.1 Requisito 6: Validación de Usuario

Al ingresar un usuario que no existe, el mensaje que aparece es el siguiente:



Figura 31.33 Pruebas 1 Requerimiento 6 Sprint 1

Fuente: Elaboración propia

Por otro lado, al ingresar un usuario o contraseña incorrecto el mensaje que muestra es el siguiente:



Figura 32.34 Pruebas 2 Requerimiento 6 Sprint 1

Fuente: Elaboración propia

Entre las sugerencias que los usuarios de prueba hicieron, están las siguientes:

1. Mostrar mensajes personalizados para el ingreso de usuario. Es decir, indicar si el usuario o contraseña es incorrecto, o si el usuario existe.

4.3.7 Sprint 2

4.3.7.1 *Requerimiento 2 y 3: Elegir Impacto y Probabilidad*

Al tratar de arrastrar la imagen de impacto al casillero de respuesta de probabilidad el resultado fue el siguiente:



Figura 33.35 Prueba 1 Requerimiento 2 y 3 Sprint 2

Fuente: Elaboración propia

Este mismo mensaje: "No puede realizar este movimiento", apareció al tratar de arrastrar la imagen de probabilidad al casillero de respuesta de Impacto. Por otro lado, al tratar de seguir a otra pregunta sin arrastrar ninguna imagen, el resultado fue el siguiente:



Figura 34.36 Prueba 2 Requerimiento 2 y 3 Sprint 2

Fuente: Elaboración propia

De estas pruebas, los comentarios del usuario fueron:

1. Usar mensajes con que indiquen con cuál elección se equivocó el usuario.
2. Indicar el número total de preguntas

4.3.8 Sprint 3

4.3.8.1 *Requerimiento 1: Seleccionar Riesgos*

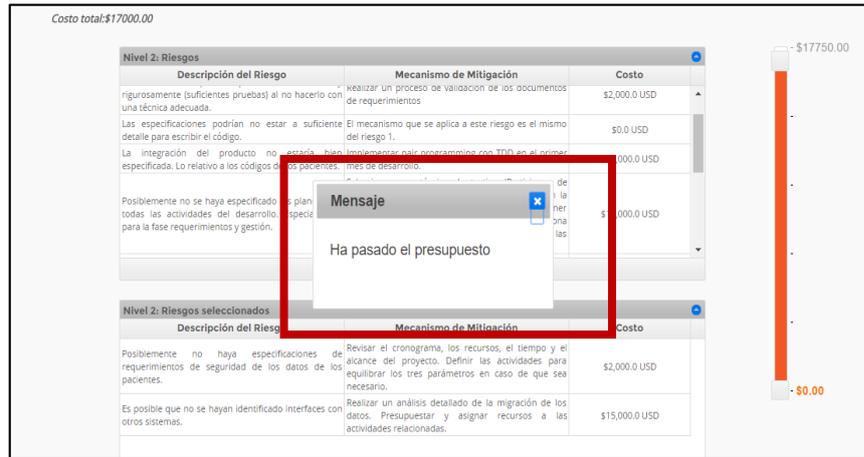


Figura 35.37 Prueba 1 Requerimiento 1 Sprint 3

Fuente: Elaboración propia

La aplicación muestra un mensaje cuando el usuario intenta soltar en la tabla de riesgos seleccionados un riesgo que tiene un costo mayor que el presupuesto disponible. Sin embargo, el usuario tiene los siguientes comentarios:

1. Agregar el presupuesto disponible a lado del costo total
2. Mostrar un mensaje cuando estés cerca de alcanzar el presupuesto
3. Ordenar los riesgos por costo

4.3.8.2 *Requerimiento 3: Comparar costo de mitigación vs. Presupuesto asignado a riesgos*

Nivel 2: Riesgos		
Descripción del Riesgo	Mecanismo de Mitigación	Costo
Posiblemente no se haya especificado los planes para todas las actividades del desarrollo. Especialmente para la fase requerimientos y gestión.	Seleccionar una técnica de testing (Particiones de Equivalencia) y elaborar los casos de prueba con la técnica seleccionada. En caso de no tener conocimiento de las técnicas contratar una persona con experiencia en pruebas por el tiempo de las pruebas.	\$12,000.0 USD
Poca familiaridad con Eclipse y JUnit por parte de los desarrolladores. Podría causar retrasos y bajar la calidad del software.	Elaborar los planes de gestión (G. de la Configuración, Gestión de la calidad, G. de proyecto) y de ingeniería de requerimientos. Ajustar las actividades al cronograma del proyecto.	\$3,000.0 USD
El cronograma no podría ser realista, posiblemente no se haya incluido la etapa de migración de los datos.	Entrenar a los desarrolladores en el uso de JUnit y Eclipse.	\$10,000.0 USD

Página 1 de 1

Nivel 2: Riesgos seleccionados		
Descripción del Riesgo	Mecanismo de Mitigación	Costo
Demoras en el tiempo de desarrollo debido a que el personal podría no estar entrenado adecuadamente para el proyecto.	Implementar la gestión de la configuración de acuerdo a un plan	\$5,000.0 USD
Posiblemente no se maneje bien el control de cambios.	Se confía en la seguridad del software de base y en la red en donde va a funcionar el software. Se asume el riesgo	\$0.0 USD
Los desarrolladores no podrían estar familiarizados con la técnica TDD, incluso luego de haber recibido capacitación.	Hacer pares de programación en donde haya una persona con habilidades para programar con TDD, se incluiría en el mecanismo para mitigar el riesgo T1.	\$0.0 USD
Las especificaciones podrían no estar a suficiente detalle para escribir el código.	El mecanismo que se aplica a este riesgo es el mismo del riesgo 1.	\$0.0 USD

-\$24850.00
-\$0.00

Figura 36.38 Prueba 1 Requerimiento 3 Sprint 3

La barra que indica el costo sí aumenta dependiendo del riesgo que se arrastra y se suelta. Sin embargo, se realizaron los siguientes comentarios:

1. Sería bueno que se muestren todos los valores en la barra
2. Cada vez que se suelta un riesgo, debería aparecer un mensaje que recuerde cuanto presupuesto está disponible

4.4 Resumen de Pruebas

Las pruebas realizadas por medio del diseño de escenarios de prueba fueron exitosas. Se comprobó que todas las funcionalidades principales de la aplicación funcionan adecuadamente. Sin embargo, sí existen ciertas mejoras que una vez implementadas van a facilitar la interacción entre el usuario y la aplicación.

CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

A partir de los avances de los medios tecnológicos y los desafíos de la metodología tradicional de aprendizaje, las herramientas de enseñanza han evolucionado. En la actualidad, existen técnicas conocidas como *e-learning* que son exitosas al implementar elementos de juego para fomentar el aprendizaje; por ejemplo, *Gamification*. Luego de haber realizado una revisión sistemática, se han analizado estas técnicas de *e-learning* y se han utilizado los elementos de *Gamification* para capturar la atención del usuario en contextos donde normalmente la atención sería mínima.

El presente trabajo de titulación, contribuye a la enseñanza de Ingeniería de *Software*, específicamente de la Gestión Riesgos en las etapas de identificación, análisis y mitigación de riesgos en la Gestión de Proyectos; por medio de la implementación de una aplicación Web utilizando el marco de trabajo *Gamification*.

Para lograrlo se definieron varios objetivos. El primer objetivo fue analizar las metodologías de aprendizaje y las técnicas de *e-learning* para justificar la elección de la herramienta a usar en el desarrollo de la aplicación web. Luego, se analizaron las funcionalidades que se deben considerar para el diseño de esta aplicación. Después de esto, se procedió al diseño de la herramienta que implemente los elementos de juego de *Gamification*. Para completar el desarrollo de la aplicación, se utilizó una metodología de gestión de proyectos ágil. Finalmente, se realizaron pruebas de usuario considerando las funcionalidades definidas en el segundo objetivo.

El primer objetivo planteado, se completó en el Marco Teórico, Capítulo 2, donde se realizó una revisión sistemática de las metodologías de enseñanza, e-learning, Gamification y Gestión de Riesgos. Por otra parte, para lograr el segundo objetivo se definió la Etapa 1 del proyecto, Preparación del Producto, referirse a la Sección 3.1. En ella se aplicaron dos actividades principales: revisión sistemática y búsqueda de una solución apropiadas. La revisión sistemática incluía: definición de conceptos ubicados en el Capítulo 2, análisis de modelos de enseñanza en la Tabla 2.2.1, elección y justificación de técnicas de e-learning ilustrados en la *Tabla 4.1.1*. A partir de la revisión sistemática se pudo definir las características de la solución que están detallados en la *Tabla 4.1.3*.

Para alcanzar el segundo, tercer y cuarto objetivo, se definió la Etapa 2 del proyecto, Desarrollo del Producto, ubicado en la Sección 3.2. En relación al objetivo 2 se utilizó el marco de diseño 6d, descrito en la Sección 2.1.7, para asegurar que la solución implemente los elementos de juego de Gamification. Luego al finalizar cada Sprint, se realizó una revisión para comprobar que se implementaron los elementos de diseño de Gamification; esto se lo puede observar en las Tablas 4.2.4, 4.2.9 y 4.2.14.

Para lograr el tercer objetivo, se utilizó Scrum como marco de desarrollo ágil para la Etapa 2, Desarrollo del producto. En cada Sprint se desarrollaron las funcionalidades de la aplicación. En la planificación del Sprint se definieron la lista de tareas a realizarse, se puede observar en las secciones 4.2.1, 4.2.2, 4.2.3. Luego, en la revisión del Sprint se comprobó que todas las tareas del Sprint se hayan completado en el tiempo programado, referirse a Tablas 4.2.3, 4.2.8 y 4.2.13.

Para completar el cuarto objetivo del trabajo de titulación se realizaron pruebas de funcionalidad para comprobar que las funcionalidades definidas

inicialmente en la *Tabla 4.1.3.* se hayan implementado adecuadamente. Las pruebas de funcionalidad se implementaron por medio del diseño de escenarios de prueba, descritos en la sección 5.3.

Por otra parte, entre las contribuciones principales de este trabajo de titulación están las siguientes:

- Aplicación Web que implementa los elementos de Gamification orientado a la enseñanza de Gestión de riesgos por medio del uso de casos de estudio.
- Análisis comparativo de las técnicas de e-learning y análisis de las técnicas de enseñanza que implementan elementos de juego.
- Revisión sistemática sobre Gamification, técnicas de e-learning y metodologías de aprendizaje.

Este trabajo de titulación ha estado limitada a la información disponible sobre la situación actual de la enseñanza de Ingeniería de Software, Gestión de Proyectos y Gestión de riesgos. Además, solamente se consideraron las etapas de identificación, análisis y mitigación de riesgos en proyectos de software. A partir de esto, se desarrolló una aplicación que se encuentra únicamente disponible a plataforma web. Además, la aplicación fue desarrollada pensando en las necesidades específicas del programa de Maestría en Ingeniería de Sistemas de la Universidad de las Fuerzas Armadas de Ecuador (ESPE).

Tomando en cuenta las limitaciones, así como también, las contribuciones de este trabajo se realizan las siguientes recomendaciones para trabajos futuros:

- Realizar un trabajo de experimentación con la aplicación web para comprobar la efectividad de usar *Gamification* en la enseñanza de Gestión de Riesgos en proyectos de Software.
- Implementar por medio de un plan de mejoras las sugerencias que los usuarios emitieron luego de las pruebas funcionales.
- Desarrollar una aplicación móvil a partir de la aplicación web para llegar a un mayor número de usuarios
- Implementar un mayor número de casos de estudio
- Implementar todas las etapas de Gestión de Riesgos en la aplicación

CAPÍTULO 6 REFERENCIAS

- Abrahamsson, P., Salo, O., Ronkainen, J., & Warsta, J. (2002). Agile software development methods Review and analysis. *Publishing VT*.
- Abt, C. (1970). *Serious Games*. Viking Press.
- Amit, S. (24 de 05 de 2015). *Choosing The Right eLearning Methods*. Obtenido de elearningindustri:
<http://elearningindustry.com/choosing-right-elearning-methods-factors-elements>
- Aparici, R., & Silva, M. (2012). Pedagogía de la interactividad. *Comunicar*.
- Aparicio, & Bacao. (2014). *Trends in the e-learning ecosystem: A Bibliometric study*. Obtenido de n Proceedings of 20th:
<http://aisel.aisnet.org/amcis2014/Posters/ISEducation/7>
- Aparicio, M. B. (2016). An e-Learning Theoretical Framework. *Educational Technology & Society*, 292–307. .
- APM. (2014). Introduction to Gamification. *Emerging Trends*.
- Badgeville. (2013). *Nike+*. Obtenido de Badgeville:
<https://badgeville.com/wiki/Nike>
- Barata, G., Gama, S., & Gonçalves, D. (2013). Engaging engineering students with gamification. *VSGAMES*.
- Barata, G., Gama, S., Fonseca, M. J., & Gonçalves, D. (2013). Improving Student Creativity with Gamification and Virtual Worlds. *Proceedings*. Ontario.
- Beard, R., & Hartley, J. (1984). *Teaching and learning in higher education*. London: Harper Row.
- Bedini González, A. (2006). *Gestión de Proyectos de Software*. Obtenido de
<http://www.inf.utfsm.cl/~guerra/publicaciones/Gestion%20de%20Proyectos%20de%20Software.pdf>
- Benitez Murube, I. M. (02 de 2009). *La concepción tradicional y constructivista del aprendizaje*. Obtenido de
<http://www.eduinnova.es/feb09/Concepcion%20Tradicional%20versus%20Constructivista%20del%20Aprendizaje.pdf>
- Berkling, K., & Thomas, C. (2013). Gamification of a Software Engineering Course. *International Conference on Interactive Collaborative Learning*, 525–530.

- Bhasin, K. (27 de 01 de 2014). *Gamification, Game-based Learning, Serious Games: Any Difference?* Obtenido de Learning Solutions Magazine:
<http://www.learningsolutionsmag.com/articles/1337/gamification-game-based-learning-serious-games-any-difference>
- Boehm, B. W. (2001). Software engineering economics. *IEEE Transactions on Pioneers and Their Contributions to Software Engineering*, 99-150.
- Bohyun, K. (2015). Understanding Gamification. *Library Technology Reports*, 51(2).
- Boumová, V. (2008). Traditional vs. Modern Teaching Methods.
- Boyle, S. (10 de 2011). *An Introduction to Game Based Learning*. Obtenido de UCD: <https://www.ucd.ie/t4cms/UCDTLT0044.pdf.pdf>
- Browne, K., & Anand, C. (2013). Gamification and Serious Game Approaches for Introductory Computer Science Tablet Software. *Proceedings*. Ontario.
- BunchBall. (2010). Gamification 101: An Introduction to the Use of Game Dynamics to Influence Behavior. *White Paper*.
- Bureau of Labor Statistics. (8 de 12 de 2015). *Employment Projections*. Obtenido de United States Department of labor:
http://www.bls.gov/emp/ep_table_103.htm
- CareerCast. (2016). *The Best Jobs of 2015*. Obtenido de <http://www.careercast.com/slide/best-jobs-2015-no-10-computer-systems-analyst>
- Carneval, A., Smith, N., & Melton, M. (2014). *STEM*. Center on Education and Workforce.
- Carpio, C. (2008). Metodos de enseñanza-aprendizaje aplicables en magisterio en el marco del espacio europeo de educación superior. *Docencia e Investigación*.
- Cechanowicz, J., Gutwin, C., Brownell, B., & Goodfellow, L. (2013). Effects of Gamification on Participation and Data Quality in a Real-World Market Research Domain. *Proceedings*. Ontario.
- Comission of the European Communities. (2000). *E-Learning: Designing Tomorrow's Education*. Obtenido de <http://europa.eu/eur-lex/en/com/cnc/2000/>
- Dabbagh, N. (2005). Pedagogical models for e-learning: A Theory-based design framework. *International Journal of Technology in Teaching and Learning*, 25-44.

- Dale, E. (1969). *Audiovisual Methods in Teaching*. NY: Dryden Press.
Obtenido de talent LMS.
- Damodharan V, R. (2007). *Innovative Methods of Teaching . Proceedings*.
Oman.
- Deterding, S., Sicart, M., Nacke, L., O'Hara, K., & Dixon, D. (2011).
Gamification. using game-design elements in non-gaming contexts.
ACM.
- Dicheva, D., Dichev, C., Agre, G., & Angelova. (2015). Gamification in
Education: A Systematic Mapping Study. *Educational Technology &
Society*, 75–88.
- Egenfeldt-Nielsen, S. (2007). Third generation educational use of
computer games. *Journal of educational Multimedia and
Hypermedia*, 263.
- Egenfeldt-Nielsen, S., & Smith, J. &. (2008). *Understanding video Games:
the Essential Introduction*. Routledge.
- Er, E., Özden, M., Arifoglu, &. & A. (2009). A blended e-learning
environment: A model proposition for integration of asynchronous
and synchronous e-learning. . *International Journal Of Learning*,
449-460.
- Farrall, F. (10 de 03 de 2015). *Case Study : Deloitte*. Obtenido de
BadgeVille: <https://badgeville.com/deloitte/>
- Felicia, P. (2009). *Digital games in schools: A handbook for teachers,
european Schoolnet*. Obtenido de Partnership AiSbl: belgium:
http://games.eun.org/upload/GIS_HANDBOOK_EN.PDF.
- Fernández-Pampillón Cesteros, A. (2009). Las plataformas e-learning
para la enseñanza y el aprendizaje universitario en Internet. *Del
mito a la realidad*, 45-73. Obtenido de Universidad Complutense.
- FUSZARD, B. (2001). *Fuszard's innovative teaching strategies in nursing*.
Gaithersburg, MD: Aspen Publishers.
- Gee, J. (2003). What video games have to teach us about. *Comput.
Entertain.*, 20.
- Gómez López, R. (2002). Análisis de los métodos didácticos en la
enseñanza. *Publicaciones*.
- Guitart, I., Rodríguez, M. E., & Cabot, J. &. (2006). Elección del modelo de
evaluación: caso práctico para asignaturas de ingeniería del
software," *Actas las XII Jornadas . Enseñanza Univ. Informática*,
191-198.

- Guri-Rosenblit, S. (2005). Distance education'and 'e-learning': Not the same thing. *Higher Education*, 467-493.
- Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, 393-416.
- Harris, J., Mishra, P., & Koehler, M. (2009). Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed. *Journal of Research on Technology in Education*, 393-416.
- Hector, M. (20 de 05 de 2015). *Starbucks Uses Gamification to Enhance End-User Loyalty*. Obtenido de App Maker: <https://www.appmakr.com/blog/starbucks-app/>
- Hernández, C. (2007). *Metodologías de enseñanza y aprendizaje en altas capacidades*. Obtenido de Universidad de La Laguna: <https://gtisd.webs.ull.es/metodologias.pdf>
- Higley, M. (15 de 10 de 2013). *Advantages Of Using Both Synchronous and Asynchronous Technologies In An Online Learning Environment*. Obtenido de Elearning Industry: <http://elearningindustry.com/benefits-of-synchronous-and-asynchronous-e-learning>
- Hrastinski, S. (2008). Asynchronous & synchronous e-learning. . *EDUCAUSE Quarterly*, 51-55.
- ISACA. (2015). Gestión práctica de riesgos para proyectos exitosos . *Conferencia Latinoamericana CACS/ ISRM*. Argentina.
- Jonassen, D. H. (1997). Analyzing and selecting instructional strategies and tactics. *Performance Improvement Quarterly*, 34-54.
- Kapp, K. (2012). Games, gamification, and the quest for learner engagement. *T+D*, 64–68.
- Kapp, K. (25 de 03 de 2013). *Two Types of #Gamification*. Obtenido de Kapp Notes: <http://karlkapp.com/two-types-of-gamification/>
- Kapp, M., & Karl. (2012). *The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education*. New York: Wiley.
- Kuzu, A. (2007). Views of Pre-Service Teachers on Blog Use for Instruction and Social Interaction. *Turkish Online Journal of Distance*.

- Leaning, M. (2015). A study of the use of games and gamification to enhance student engagement, experience and achievement on a theory-based course of an undergraduate media degree. *Journal of Media Practice*, 150-170.
- Lee, J., & Hammer, J. (2011). Gamification in education: What, how, why bother? *Academic Exchange Quarterly*, 146.
- López, J. (2013). *Gestión de Riesgos*. Obtenido de Metodología del PMBOK: <http://pmbokproyectos.blogspot.com/p/gestion-de-riesgos.html>
- Marczewski. (03 de 2013). *Game Thinking Decision Tree*. Obtenido de Gamification.UK: <http://www.gamified.uk/wp-content/uploads/2013/02/Game-Thinking-Decision-Trees-Small.jpg>
- Marczewski, A. (2012). *Gamification: A Simple Introduction and a Bit More*. Seattle, WA: Amazon Digital Services.
- Marczewski, A. (3 de 11 de 2013). *What's the difference between Gamification and Serious Games?* Obtenido de Gamasutra: http://www.gamasutra.com/blogs/AndrzejMarczewski/20130311/188218/Whats_the_difference_between_Gamification_and_Serious_Games.php
- Markov, I. (8 de 01 de 2014). *Are There Too Many Students Going Into Computer Science?* Obtenido de Forbes Tech: <http://www.forbes.com/sites/quora/2014/01/08/are-there-too-many-students-going-into-computer-science/#70df7dba6bbf>
- McGonigal, J. (2011). *Reality Is Broken: Why Games Make Us Better and How They Can Change the World*. . *Jonathan Cape*. London.
- Meister, J. C. (02 de 01 de 2013). *How Deloitte Made Learning a Game*. Obtenido de HBR free Webinar: <https://hbr.org/2013/01/how-deloitte-made-learning-a-g/>
- Meloni, J. (11 de 01 de 2010). *Tools for synchronous and asynchronous classroom discussion*. Obtenido de The Chronicle of Higher Education.: <http://chronicle.com/blogs/profhacker/tools-for-synchronousasynchronous-classroom-discussion/22902>
- Montoya-Suárez, L. M., & Pulgarín-Mejía, E. (2013). ENSEÑANZA EN LA INGENIERÍA DE SOFTWARE: APROXIMACIÓN A UN ESTADO DEL ARTE. *Lámpsakos*, 76-91.
- Moore, J., Dickson-Deane, C., & Galyen, K. (2011). e-Learning, online learning, and distance learning environments: Are they the same? *Internet and Higher Education*, 129-135.

- Ospina, M. G. (2013). *Metodología para la gestión de proyectos bajo los lineamientos del Project Management Institute en una Empresa Eléctrica*. Obtenido de <http://www.bdigital.unal.edu.co/11161/1/940429.2013.pdf>
- Parras, D. (2003). *Manual de Estrategias Enseñanza-Aprendizaje*. Medellín-Colombia: SENA.
- Pelling, N. (2011). *The (Short) Prehistory of 'Gamification*. Obtenido de Funding Startups (& other impossibilities).: <https://nanodome.wordpress.com/2011/08/09/the-short-prehistory-of-gamification/>.
- Penadés, M. C. (2006). Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). *Técnica Administrativa, Buenos Aires*.
- PMI. (2008). *Should You Be teaching PM*. Obtenido de PMI: http://www.mosaicprojects.com.au/PDF/Why_teach_PM.pdf
- PMI. (2012). *A guide to Project Management: Body of Knowledge*. Pennsylvania: Project Management Institute, Inc.
- Rist, R., & Hower, S. (1996). *Implementing Learning Technology*. Edinburgh: Learning Technology Dissemination Initiative.
- Schwaber, K., Beedle, M., & Martin, R. (2003). *Agile Software Development with SCRUM*. Prentice Hall.
- Scrivener, J. (2005). *Lerning, Teaching*. Oxford: Macmillan.
- Shang Gao, L. R. (2015). *Modern Techniques for Successful IT Project Management*. IGI Global.
- Shuber, K. (2013). *Scrum Guide*. Obtenido de Scrum Guides: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Smith, D. F. (14 de 09 de 2015). *Gamified Social Studies Takes Off in Fantasy Geopolitics*. Obtenido de <http://www.edtechmagazine.com/k12/article/2015/09/gamified-social-studies-takes-fantasy-geopolitics>
- Sommerville, I. (2005). *Ingeniería de Software*. Pearson Educación.
- Standish Group. (4 de 10 de 2015). *Chaos Report*. Obtenido de InfowQ: <https://www.infoq.com/articles/standish-chaos-2015>
- Taran, G. (2007). Using Games in Software Engineering Education to Teach Risk Management, 20th. *Conference on Software Engineering Education & Training*.

- TLMS. (01 de 2014). *E-LEARNING CONCEPTS , TRENDS , APPLICATIONS*. Obtenido de Talent Tlms:
<http://www.talentlms.com/elearning/elearning-101-jan2014-v1.1.pdf>
- Torres Maldonado, H. (2009). *Didáctica General*. Obtenido de
<http://unpan1.un.org/intradoc/groups/public/documents/icap/unpan039746.pdf>
- Uyaguari, F., Intriago, M., & Salazar, E. (2015). Gamification Proposal for a Software Engineering Risk Management Course. *Springer*.
- Vargas, A. (2009). Métodos de Enseñanza. *CSI.F*.
- Werbach, K. (29 de 05 de 2013). *Gamification Design Steps: An introduction*. Obtenido de
<http://www.epicwinblog.net/2013/05/gamification-design-steps-introduction-i.html>
- Xie, T., Tillman, N., & De Halleux, J. (2013). Educational Software Engineering: Where Software Engineering, Teaching and Gaming Meet. *Games and Software Engineering*, 33-36.
- Zapata-Jaramillo, C. M. (2006). Alineación entre metas organizacionales y elicitación de requisitos del software. *DYNA*, 101-110.
- Zapata-Jaramillo, C. M., & Villegas, S. M. (2012). Reglas de consistencia entre modelos de requisitos de UN-Metodo. *Revista Universidad Eafit*, 40-59.
- Zapata-Jaramillo, C. M., Gelbukh, A., Arango, F., & Hernández, A. &. (2006). UN-Lencep: Obtención automática de diagramas UML a partir de un lenguaje controlado. *Avances en la Ciencia de la Computación*, 254-256.
- Zapata-Jaramillo, C., & Awad-Aubad, G. (2007). Requirements Game: Teaching Software Project Management. *CLEI Electronic Journal*.

CAPÍTULO 7 ANEXOS

7.1 Anexo 1: Diagrama de Entidad Relación

