FACULTAD 2

SISTEMA INFORMÁTICO PARA LA EVALUACIÓN DE IMPACTO DE LAS INFRAESTRUCTURAS DE TECNOLOGÍAS DE LA INFORMACIÓN EN LAS ORGANIZACIONES

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es): Ernesto Elias Elzaurdín Valdés

Tutor(as): Dra. C. Mónica Peña Casanova

Ing. Clayret Echenique Quintana

La Habana, diciembre de 2023

Año 65 de la Revolución

Si me dijeran pide un deseo,
Preferiría un rabo de nube,
Que se llevara lo feo
Y nos dejara el querube.
Un barredor de tristezas,
Un aguacero en venganza
Que cuando escampe parezca
Nuestra esperanza

Silvio Rodriguez

DECLARACIÓN DE AUTORÍA

El autor del trabajo de diploma con título "**Sistema informático para la evaluación de impacto de las infraestructuras de Tecnologías de la Información en las organizaciones**", concede a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la investigación, con carácter exclusivo. De forma similar se declara como único autor de su contenido. Para que así conste firma la presente a los ____ días del mes de _____ del año _____.

Ernesto Elías Elzaurdín Valdés

_____

Firma del Autor

Dra. C. Mónica Peña Casanova

_____

Firma del Tutor

Ing. Clayret Echenique Quintana

_____

Firma del Tutor

DATOS DE CONTACTO

Tutora: Dra. C. Mónica Peña Casanova

Especialidad: Doctora en Ciencias Técnicas

Lugar de Trabajo: Facultad 2 de la Universidad de las Ciencias Informáticas

Responsabilidades asumidas: Decana de la Facultad 2 de la Universidad de las Ciencias Informáticas

Correo electrónico: monica@uci.cu

Tutora: Ing. Clayret Echenique Quintana

Especialidad: Ingeniera en Ciencias Informáticas

Lugar de trabajo: Departamento Docente de Ciberseguridad de la Facultad 2 de la Universidad de las Ciencias Informáticas

Correo electrónico: cechenique@uci.cu

AGRADECIMIENTOS

Quiero agradecer a la familia de mi novia por como me han acogido durante estos años y como me han apoyado

A Nory por el cariño, la atención y acogerme como un hijo

A mi hermana, sabes que te quiero muchísimo, eres una personita muy especial

A mi papá por los dolores de cabeza, pero sobre todo por el amor, la preocupación y los consejos, eres una inspiración y un ejemplo para mí

A mi mamá que aunque no puede estar hoy aquí sé que está muy orgullosa y feliz, eres lo más grande que existe

A mis primas por todo el cariño y toda la ayuda durante estos años

A mi abuelo Ernesto que no puede acompañarme pero que estaría muy feliz de verme graduado, gracias por todo, también por él estoy aquí hoy

A mis abuelos, gracias por todo el amor, por preocuparse siempre, por siempre tratar de darme toda la ayuda y el apoyo posible

A mis amistades durante todos estos años, muchos ya no están, pero sigo contando con ellos, hace mucho tiempo dejaron de ser compañeros, ni siquiera amigos, después de tantos años y tantos momentos juntos hoy son mis hermanos

A Roco por ser el perrito más lindo del mundo, por alegrarme todos los días, aunque me despierte a las 8 de la mañana

Al tribunal

A la profe Mónica por su guía y su ayuda

Y a la persona más especial del mundo, gracias por estos cuatro años mágicos, gracias por ayudarme en todo, por estar siempre conmigo, por cuidarme, por ser mi tutora, por decirme dos disparates cuando me lo merecía para que dejara la bobería, nunca terminaría si te agradezco por todo lo que me das

DEDICATORIA

A mi abuelo Ernesto en donde quiera que Dios lo haya acogido por formarme desde pequeño y regalarme el amor, la inteligencia y el deseo de ser alguien importante en la vida.

RESUMEN

En un mundo cada vez más dependiente de la infraestructura tecnológica, las organizaciones enfrentan múltiples riesgos a la seguridad de la información, lo que puede afectar a su desempeño y sostenibilidad. Gestionarlos se ha convertido en una herramienta esencial para identificar, evaluar y mitigar las amenazas, es por ello que se hace necesario implementar mecanismos que puedan ayudar a las organizaciones a tratar el impacto de estos incidentes de forma eficaz. La presente investigación pretende describir el proceso de desarrollo de un sistema informático para medir el impacto de las infraestructuras de Tecnologías de la Información en las organizaciones. Se realizó un estudio sobre principales marcos de referencia para la gestión de riesgos y técnicas para medir impacto descritas por NIST, ISO 27000, MAGERIT y COBIT5. El proceso de desarrollo de software fue llevado a cabo a través la metodología ágil Extreme Programming para desarrollar una aplicación web sobre el marco de trabajo Django y lenguaje de programación Python para backend, HTML5, CCS3 y Java Script para frontend y como gestor de base de datos PostgreSQL. Se trazó una estrategia de pruebas de acuerdo con la metodología seleccionada, además de las pruebas de carga, rendimiento y estrés, necesarias para este tipo de aplicación.

PALABRAS CLAVE: activo, amenaza, impacto de las infraestructuras de Tecnologías de la Información, riesgo, sistema de gestión de la seguridad de la información.

ABSTRACT

In a world increasingly dependent on technological infrastructure, organizations face multiple risks to information security, which can affect their performance and sustainability. Managing them has become an essential tool to identify, evaluate and mitigate threats, which is why it is necessary to implement mechanisms that can help organizations deal with the impact of these incidents effectively. This research aims to describe the development process of a computer system to measure the impact of Information Technologies infrastructures on organizations. A study was carried out on the main reference frameworks for risk management

and techniques to measure impact described by NIST, ISO 27000, MAGERIT and COBIT5. The software development process was carried out through the agile Extreme Programming methodology to develop a web application on the Django framework and Python programming language for backend, HTML5, CCS3 and Java Script for frontend and as database manager. PostgreSQL data. A testing strategy was drawn up in accordance with the selected methodology, in addition to the load, performance and stress tests, necessary for this type of application.

KEYWORDS: *asset, threat, impact of Information Technologies infrastructures, risk, information security management systems.*

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Desde hace unos años las organizaciones apuestan por la transformación digital pues son evidente los beneficios que aportan al negocio. Es importante destacar que esta pretende obtener beneficios en cuanto al manejo de la información, procesos, recursos e incluso reduce errores humanos. Sin embargo, no se trata de implementar nuevas tecnologías, sino de lograr un cambio cultural y organizacional que garantice el éxito en las organizaciones (Liendo Afonso, 2023).

Es evidente que una vez realizada las inversiones en infraestructura tecnológica se debe garantizar la seguridad de las mismas y es que hoy están muy presentes las amenazas cibernéticas, las cuales representan una vulnerabilidad para las organizaciones. Es por ello que se hace necesario centrar esfuerzos en implementar medidas de seguridad adecuadas para reducir los niveles de riesgos a los que se expone la información (Martínez Landrove, 2019).

El crecimiento de las infraestructuras de Tecnologías de la Información (TI), trae consigo un auge de las investigaciones sobre el impacto que ellas representan para las organizaciones (Casanova & Calderón, 2020a). Los resultados de varias investigaciones permitieron identificar que en la medida que sea mayor la alineación entre las TI y los objetivos del negocio, más amplio es el valor añadido que representan las infraestructuras TI para una organización (Pérez Lorences, 2014).

El impacto se mide de acuerdo a la misión de la entidad, por lo que es vital comprender todos los activos de TI. Cada activo tiene un valor, muchos son componentes clave para respaldar los servicios críticos que se brindan a los usuarios. Estos, además, influyen directamente en el capital y la valoración de la organización, y los riesgos de TI pueden tener un impacto directo en el presupuesto. Para cada organización, es vital y desafiante determinar las condiciones que realmente impactan a la misión; es muy importante analizar y comprender continuamente los recursos que permiten cumplir con los objetivos y que pueden verse comprometidos por los riesgos de ciberseguridad (Quinn et al., 2022).

Un estudio exploratorio realizado por Casanova (2020) describe el estado de la gestión de las infraestructuras de TI en Cuba, este arrojó que en las organizaciones las mayores medias de impacto “se corresponden con la lentitud en la respuesta a las necesidades de la organización, lo cual implica un serio problema de alineamiento” (p. 41). En segundo lugar, destaca “los problemas de implementar nuevos sistemas, debido a que conlleva un incremento de la complejidad de la gestión” (p. 41).

Las organizaciones cubanas dependen tecnológicamente de sus proveedores, lo que dificulta la renovación periódica y pertinente de las infraestructuras TI. De ahí que surge una conciencia sobre la importancia de la evaluación del impacto de la infraestructura. Cuba por su condición de ser un país en desarrollo no produce tecnología y por tanto debe importarla, lo que representa un mayor esfuerzo debido al bloqueo, de ahí que la evaluación del impacto reviste una importancia especial (Casanova & Calderón, 2020a).

Para ello existen marcos de referencia que describen de manera general como medir el impacto de la seguridad de la información. Esto implica vencer la barrera de la diversidad de estructura, procesos y términos, para la integración, alineamiento, disminución de la complejidad y aplicación pertinente en las organizaciones, por lo que deben ser adaptados a los objetivos y no existen herramientas que ayuden a documentar este análisis.

De lo planteado anteriormente plantado se deriva el siguiente **problema a resolver**: ¿Cómo contribuir a la reducción de la complejidad en la gestión de información del proceso de evaluación del impacto de las TI en las organizaciones cubanas? Definiendo como **objetivo general:** Desarrollar un sistema informático que contribuya a la gestión de información del proceso de medición de impacto de la infraestructura de TI en las organizaciones cubanas.

Se define como **objeto de estudio** el proceso de medición de impacto de la infraestructura de TI en las organizaciones, enmarcado en el **campo de acción** de los sistemas informáticos para la evaluación de impacto de la infraestructura de TI en las organizaciones cubanas.

Para dar seguimiento al objetivo general se trazan los siguientes **objetivos específicos:**

1. Sistematizar los fundamentos teóricos-metodológicos que sustentan a la evaluación de impacto de las TI en las organizaciones.
2. Modelar un sistema para la evaluación de impacto de las TI en las organizaciones.
3. Implementar el sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones.
4. Validar el funcionamiento del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones.

Con el propósito de cumplir los objetivos específicos se definen las siguientes **tareas de investigación**:

1. Revisión de los conceptos relacionados con el objeto de estudio y el campo de acción.
2. Descripción y análisis del estado actual de las soluciones informáticas para la evaluación de impacto de las TI en las organizaciones.
3. Descripción de las herramientas, lenguajes, tecnologías y metodología de desarrollo de software a utilizar.
4. Modelación del proceso de evaluación de impacto de las TI en las organizaciones.
5. Definición de los requisitos funcionales y no funcionales del sistema.
6. Realización de los artefactos resultantes de la Ingeniería de Requisitos desarrollada.
7. Aplicación de la estrategia de pruebas al sistema de evaluación de impacto de las TI en las organizaciones.

Los **métodos teóricos** utilizados son:

- **Analítico-sintético:** Se utiliza este método para el procesamiento de la información referente la evaluación de impacto en las organizaciones y arribar a las conclusiones de la investigación, así como para precisar las características del trabajo a realizar.

- **Inductivo-deductivo:** Se emplea principalmente para la elaboración del marco teórico de la investigación.
- **Modelado**: se emplea para modelar los artefactos resultantes en la etapa de diseño de la propuesta de solución.
- **Sistémico-estructural-funcional**: Se aplica en la elaboración de la estrategia metodológica para el diseño e implementación del sistema informático de evaluación de impacto de la infraestructura. Posibilitó la integración de todos los elementos investigados de manera independiente para conformar toda la investigación realizada.

Se utilizó como **método empírico**:

- **Entrevista no estructurada**: Se aplica al cliente con el objetivo de conocer acerca de la evaluación de impacto de TI en las organizaciones. Permite esclarecer el estado actual del proceso de evaluación de impacto de TI en Cuba.

El presente trabajo de diploma se **estructura** en tres capítulos:

Capítulo 1: Fundamentos teóricos-metodológicos de la evaluación de impacto de la infraestructura de TI en las organizaciones

Se realiza un análisis de los principales términos y definiciones relacionados con la evaluación de impacto, del estado del arte y de la metodología, herramientas, tecnologías y lenguajes que se utilizan en el desarrollo de la solución.

Capítulo 2: Planeación, diseño y codificación del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones

Se brida una explicación del proceso a informatizar, se describe la solución y se detallan los artefactos resultantes de la metodología de desarrollo de software utilizada durante el proceso de planeación, diseño e implementación de la propuesta de solución.

Capítulo 3: Validación del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones

Se describe la validación de la solución obtenida mediante una estrategia de pruebas, entre las que se incluyen las pruebas de unitarias, de aceptación y de rendimiento del sistema.

CAPÍTULO I: Fundamentos teóricos-metodológicos de la evaluación de impacto de las TI en las organizaciones

Introducción

El presente capítulo tiene como objetivo la exposición de los principales conceptos relacionados a los sistemas de medición de impacto. También se realiza la presentación de un estudio del estado del arte y las soluciones informáticas para medir el impacto de las TI en las organizaciones. Además, se enuncian las principales herramientas, tecnologías, lenguajes y la metodología de desarrollo de software a implementar en la propuesta de solución.

1.1. Proceso de medición de impacto de las infraestructuras de TI

El proceso de evaluación de impacto de las infraestructuras de TI en las organizaciones es un proceso estratégico que implica la identificación, medición y evaluación de los efectos de la infraestructura de TI en la organización (Casanova & Calderón, 2020b). Este proceso puede incluir varios pasos, como la identificación de los objetivos de la organización, la identificación de los beneficios esperados de la infraestructura de TI, la medición de los efectos de la infraestructura de TI en la organización y la evaluación de los resultados (Casanova & Calderón, 2020b).

Infraestructura de TI

La infraestructura de TI es la base de los sistemas que van a dar soporte al trabajo y funcionamiento de las comunicaciones. Los servicios de TI en general, desde aplicaciones hasta sistemas de servicios corporativos, registros y productos inteligentes necesitan de una infraestructura de TI, por lo tanto, todo servicio tecnológico requiere de una infraestructura para funcionar (Loarte Cabello, 2021).

1.2. Impacto en los Sistemas de Gestión de Seguridad de la Información

La norma ISO 27001 define el impacto generado sobre un activo de información en los Sistemas de Gestión de la Seguridad de la Información (SGSI) como la consecuencia de la materialización de una amenaza. El impacto es, la diferencia entre las estimaciones del estado de seguridad del activo antes y después de

materializar las amenazas. En un SGSI al materializarse una amenaza el activo cambia de estado, es decir, antes de producirse la amenaza tenemos un activo y después de que efectúe la amenaza tenemos la diferencia entre el estado anterior y el posterior a la amenaza (Toro, 2015).

1.2.1. Impacto potencial en organizaciones e individuos

Si se conoce el valor de los activos para las organizaciones y los efectos que causan las amenazas, se puede medir el impacto que tienen sobre el sistema. El valor del sistema se centra en la información que maneja y los servicios que presta; pero las amenazas suelen materializarse en los medios. Se ha demostrado que el análisis cualitativo a través de métodos simples sin ser muy preciso, aciertan en identificación de la importancia que tiene una materialización de una amenaza sobre un activo (Lopez Crespo, Amutio Gomez, & Candau, 2006).

Para medir ese impacto potencial a través de estos métodos se define una escala de impacto que valora su magnitud en organizaciones o individuos en caso de que haya una violación de la seguridad, es decir, una pérdida de confidencialidad, integridad o disponibilidad. La aplicación de estas definiciones debe realizarse dentro del contexto de cada organización y del interés nacional general (Lopez Crespo et al., 2006)

Esta escala de manera general se define como:

- El impacto potencial es BAJO si:

Se podría esperar que la pérdida de confidencialidad, integridad o disponibilidad tenga un efecto adverso limitado en las operaciones de la organización, los activos de la organización o los individuos.

- El impacto potencial es MODERADO si:

Se podría esperar que la pérdida de confidencialidad, integridad o disponibilidad tenga un efecto adverso grave en las operaciones de la organización, los activos de la organización o los individuos.

- El impacto potencial es ALTO si:

Se podría esperar que la pérdida de confidencialidad, integridad o disponibilidad tenga un efecto adverso grave o catastrófico en las operaciones de la

organización, los activos de la organización o los individuos (*Estándares para la categorización de seguridad de Información federal y sistemas de información*, 2004).

El análisis del impacto potencial debe considerar el impacto a la misión de las organizaciones en caso de una pérdida o degradación del activo. En el informe interinstitucional de NIST IR 8286D (2022), se asegura que “es probable que la interrupción, el deterioro o la divulgación no autorizada de un recurso clave cause implicaciones financieras, de reputación y operativas para la empresa con posibles consecuencias fiscales, regulatorias o de competencia” (p. 7). A través de la colaboración entre la directiva, los especialistas y con la aplicación de métodos correspondientes, se logra comprender la importancia del activo que sirve luego para analizar los riesgos con respecto a los requisitos de confidencialidad, integridad y disponibilidad.

1.2.2. Medición de impacto

La medición de impacto se encuentra estrechamente relacionada con el análisis de riesgos, se trabaja con múltiples elementos que hay que combinar en un sistema para ordenarlo por importancia (Lopez Crespo et al., 2006). Afortunadamente existen muchas técnicas eficaces para analizar los riesgos potenciales que tienen más probabilidades de tener un impacto significativo. La comprensión de la probabilidad de eventos de amenaza y sus impactos potenciales también se basa en la experimentación, la investigación de eventos de riesgo anteriores y la investigación de experiencias de riesgo de organizaciones similares (Stine, Quinn, Witte, & Gardner, 2020).

Técnicas específicas de MAGERIT

La guía de técnicas propuesta por la Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información (MAGERIT) en su versión 3.0 explica métodos de análisis mediante tablas, algorítmico y árboles de ataque para estimar la magnitud del impacto y de los riesgos. En este epígrafe solo se explicarán los dos primeros métodos.

El análisis mediante tablas define una escala útil para calificar la importancia de los activos definida por los valores: muy bajo (MB), bajo (B), medio (M), alto (A) y muy alto (MA) que son introducidos en una tabla de doble entrada (valor, degradación) en la que aquellos activos que reciben calificación de impacto muy alto son objetos de atención inmediata (Lopez Crespo et al., 2006).

impacto		***degradación***		
		1%	10%	100%
valor	MA	M	A	MA
	A	B	M	A
	M	MB	B	M
	B	MB	MB	B
	MB	MB	MB	MB

Figura 1: Matriz de impacto según la degradación y el valor del activo.

A su vez, modela impacto, probabilidad y riesgo a través de escalas similares como se muestra en la **figura 2.** Luego, estos valores son combinados con el objetivo de medir o estimar riesgos.

escalas		
impacto	probabilidad	riesgo
MA: muy alto	**MA:** prácticamente seguro	**MA:** crítico
A: alto	**A:** probable	**A:** importante
M: medio	**M:** posible	**M:** apreciable
B: bajo	**B:** poco probable	**B:** bajo
MB: muy bajo	**MB:** muy raro	**MB:** despreciable

Figura 2: Escalas según el impacto, la probabilidad y el riesgo

riesgo		probabilidad				
		MB	B	M	A	MA
impacto	MA	A	MA	MA	MA	MA
	A	M	A	A	MA	MA
	M	B	M	M	A	A
	B	MB	B	B	M	M
	MB	MB	MB	MB	B	B

Figura 3: Matriz de riesgo según el impacto y la probabilidad de que se materialice una amenaza

El análisis algorítmico presenta dos modelos, el cualitativo que busca una valoración relativa del riesgo que corren los activos y el cuantitativo que busca la forma de responder a la pregunta de cuánto más y cuánto menos. En el modelo cualitativo, los activos tienen una escala de valor relativo, definiendo un valor “v0” entre los valores que preocupan y los que no. Sobre esta escala se mide el valor del propio activo o el acumulado, el impacto de una amenaza y el riesgo al que está expuesto. El impacto mide el valor de ocurrencia potencial y el riesgo lo pondera con la frecuencia estimada de ocurrencia de la amenaza. El impacto mide el costo si se materializa la amenaza mientras que el riesgo mide la exposición en un cierto periodo de tiempo (Lopez Crespo et al., 2006).

En el modelo cuantitativo, se trabaja con valores que son números reales positivos, incluyendo al cero, modelando el grado de dependencia entre activos. Se mide el valor del activo propio o acumulado, el impacto de ocurrencia de una amenaza y el riesgo que pondera ese impacto con la frecuencia estimada de dicha ocurrencia. Si la valoración del activo es económica, el impacto es el costo que produjo la amenaza y el riesgo es la cantidad que hay que predecir como pérdidas anuales (Lopez Crespo et al., 2006).

Categorización aplicada a tipos de información

En la publicación 199 de Normas Federales de procesamiento de Información (FIPS PUB 199, por sus siglas en inglés) se define una categoría de seguridad (CS) de un sistema de información que permite analizar con profundidad las

categorías de seguridad según los tipos de información para medir el impacto potencial en cuanto a los pilares de la seguridad de la información (confidencialidad, integridad, disponibilidad). Una CS según FIPS PUB 199 (2004) es:

> La caracterización de la información o un sistema de información basada en una evaluación del impacto potencial que una pérdida de confidencialidad, integridad o disponibilidad de dicha información o sistema de información tendría en las operaciones de la organización, los activos de la organización o los individuos. (p. 8)

Para cada tipo de información se definen valores aceptables que medirán si el impacto es alto (A), moderado (M), bajo (B) o no aplica (NA). El formato generalizado para expresar la CS, de un tipo de información propuesto es:

> Tipo de información CS = {(confidencialidad, impacto), (integridad, impacto), (disponibilidad, impacto)}.

El valor del impacto potencial NA solo es objeto de seguridad en el pilar de confidencialidad.

1.3. Análisis de marcos de referencia de gestión de riesgos y medición de impacto.

Estos marcos de referencia proporcionan una guía a las organizaciones para fortalecer sus sistemas de seguridad ante la ocurrencia de algún incidente. Un factor muy importante es ayudar a desplegar un sistema para la gestión de riegos a la seguridad de la información (ORTEGA CANDEL, 2021). Algunos de estos se describen a continuación.

1.3.1. NIST (Marco de ciberseguridad del Instituto Nacional de Estándares y Tecnologías de Estados Unidos)

El marco de ciberseguridad de NIST se diseñó con un conjunto de buenas prácticas para mitigar los riesgos asociados a la ciberseguridad en una organización. Promueve la protección y resiliencia de infraestructuras críticas y está diseñado para fomentar la gestión de riesgos y la ciberseguridad (ORTEGA CANDEL, 2021).

El marco está organizado en cinco funciones clave los cuales proporcionan una visión integral del ciclo de vida para la gestión del riesgo de ciberseguridad en el tiempo. La figura 1 muestra las funciones y las actividades que brindan un punto de partida para la mejora de la organización (Mahn, Topper, Quinn, & Marron, 2021):

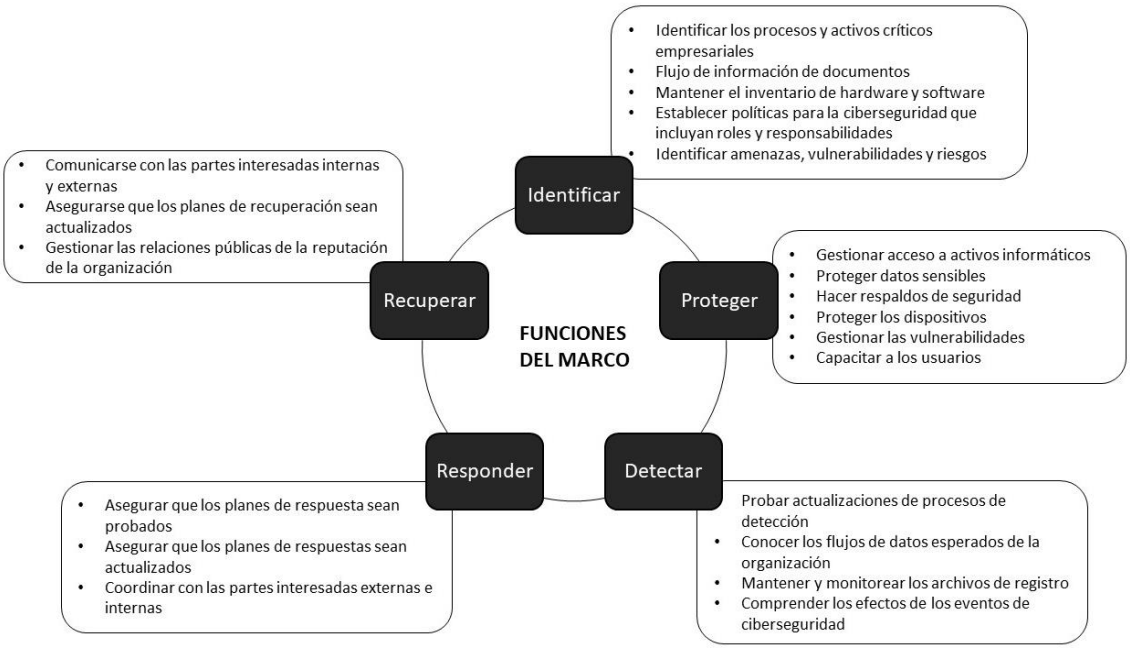


Figura 4: Funciones y actividades de NIST. (Mahn et al., 2021)

NIST provee un Marco de Gestión de Riesgos (RMF, por sus siglas en inglés) con siete pasos completos para que cualquier organización pueda administrar el riesgo de seguridad y privacidad de la información. Este se vincula con un conjunto de estándares y pautas para respaldar la implementación de programas de gestión de riesgos (Computer Security Division, 2016b).

Los pasos para implementar RMF en las organizaciones propuestos por NIST son (Computer Security Division, 2016a):

- **Preparación**: Define las actividades esenciales para preparar a la organización para gestionar los riesgos de seguridad y privacidad.
- **Catalogación**: Categoriza el sistema y la información procesada, almacenada y transmitida en función de un análisis de impacto

- **Selección**: Selecciona el conjunto de controles de NIST SP 800-53 para proteger los sistemas en función de las evaluaciones de riesgos.
- **Implementación**: Implementa y documenta cómo se implementan los controles.
- **Evaluación**: Evalúa para determinar si los controles están en su lugar, funcionando según lo previsto y produciendo los resultados deseados.
- **Autorización**: El directivo toma una decisión basada en el riesgo para autorizar el sistema (para operar).
- **Monitorización**: Monitorea continuamente la implementación del control y los riesgos para el sistema.

Figura 5: Pasos para la implementación de NIST RMF. (Computer Security Division, 2016a)

El RMF de NIST se enfatiza en la importancia de la gestión continua de riesgos y estimula a las organizaciones a integrar procesos de gestión durante todo el ciclo de vida del desarrollo del sistema. Al proporcionar un enfoque estructurado, repetible y mensurable para la gestión de riesgos, el RMF permite a las organizaciones salvaguardar eficazmente sus sistemas de información y mantener el cumplimiento de las regulaciones pertinentes (Ahmet, 2023).

1.3.2. Serie ISO/IEC 27000

La serie ISO/IEC 27000 está compuesto por estándares internacionales que contiene directrices para la seguridad de la información. Presenta buenas

prácticas y procedimientos tanto físicos como de seguridad. Incorpora reglas que permiten reducir el creciente número de amenazas, resolver problemas de seguridad existentes y mejorar los objetivos de seguridad en general (Meriah & Rabai, 2019).

La norma **ISO/IEC 27001** que define un modelo para establecer, implementar, operar, monitorear, revisar y mejorar el Sistema de Gestión de Seguridad de la Información (SGSI). Agrupa once categorías con requisitos de seguridad de la información, estas a su vez están comprendidas por subcategorías, cada una con los correspondientes requisitos de cumplimiento de alto nivel (Meriah & Rabai, 2019).

La norma **ISO/IEC 27002** comprende un código de buenas prácticas para la gestión de la seguridad de la información. Describe cientos de controles que se pueden implementar introducidos por la norma ISO/IEC 27001 (Meriah & Rabai, 2019).

La norma **ISO/IEC 27005** contiene las pautas para la Gestión de Riesgos de Seguridad de la Información (ISRM, por sus siglas en inglés) en una organización. Basa sus conceptos, modelos, procesos y terminologías de conocimiento definidos por la norma ISO/IEC 27001 y ofrece ayuda para su implementación adoptando un enfoque de la gestión de riesgos (Meriah & Rabai, 2019).

La norma **ISO/IEC 27011** ofrece un manual de interpretación de la implementación y gestión de la seguridad de la información en organizaciones de telecomunicaciones basada en ISO/IEC 27002:2005 (Bernal Medina, 2022). Permite establecer políticas, procedimientos y controles para minimizar los riesgos de las organizaciones de telecomunicaciones. Se ha visto la necesidad de implementar esta norma para gestionar adecuadamente los activos de la empresa y continuar con el éxito de las actividades (Avilés Armijos & Uyaguari Guartatanga, 2012).

La norma **ISO/IEC 27033** está dedicado a la seguridad de la red (Bernal Medina, 2022). Ofrece una guía completa para el desarrollo de la seguridad en las redes y los servicios de la red. Se ocupa de la planificación e implementación de la seguridad mediante sus directrices y medidas (Ochoa Palomino, 2019).

La norma **ISO/IEC 27034** se dedica a la seguridad de las aplicaciones informáticas (Bernal Medina, 2022). Proporciona consideraciones para el desarrollo seguro de software, así como factores que pueden afectar a la seguridad general de las aplicaciones (Karakaneva, 2014). Garantiza que los softwares aseguren sus niveles de seguridad para el apoyo a los SGSI.

1.3.3. COBIT 5

COBIT (Objetivos de Control de la Información y Tecnologías Afines) es un marco integral diseñado para que las organizaciones puedan alcanzar sus objetivos estratégicos a través de una gobernanza y gestión efectivas de la TI a nivel empresarial. COBIT 2019 en su última versión se enfatiza en la alineación de los objetivos comerciales de TI y ofrece un enfoque holístico para el gobierno de TI. Contiene diversos aspectos como la gestión de riesgos, el cumplimiento y la medición del desempeño (Ahmet, 2023).

En cuanto a gestión y gobernanza de riesgos, este marco proporciona un enfoque estructurado que permite identificar, evaluar y mitigar sistemáticamente los riesgos de TI. Ofrece una serie de objetivos de control genéricos y una lista completa de procesos de TI, que pueden ser adaptadas a las necesidades específicas y al panorama de riesgos de una organización («COBIT | Control Objectives for Information Technologies», 2023).

El enfoque del marco para el gobierno de riesgos se centra en la importancia de incorporar la gestión de riesgos en la estructura general de gobierno de TI. Los directivos y los profesionales de TI participan en el proceso de gestión de riesgos para garantizar una toma de decisiones y una rendición de cuentas exitosa. El marco también alienta a las organizaciones a adoptar una estrategia de gestión de riesgos (Ahmet, 2023).

COBIT se divide en tres componentes: marco, principios y objetivos de gobernanza y gestión, dedicados a ofrecer un modelo integral que satisfaga a las partes interesadas y vinculado a los objetivos específicos de la organización. Por otra parte, en lo que a gestión de riesgos se refiere COBIT (Risk IT) incluye tres grandes procesos: evaluar, dirigir y monitorear (EDM, por sus siglas en inglés);

alinear, planificar y organizar (APO, por sus siglas en inglés) y monitorear, evaluar y valorar (MEA, por sus siglas en inglés) que ayudan a las organizaciones a gestionar los riesgos (Ahmet, 2023).

1.3.4. Comparación entre los marcos de referencia NIST RMF, COBIT (Risk IT) e ISO/IEC 27005

Cada uno de estos marcos ofrecen una **metodología** única medir el impacto de los riesgos. NIST RMF se centra en los sistemas de TI y la gestión de riesgos de ciberseguridad (Force, 2018). COBIT en cuanto a este aspecto aborda específicamente la gestión de riesgos relacionados con TI dentro del contexto de la gobernanza y la gestión de TI (Ahmet, 2023) y dentro de la serie de ISO/IEC 27000, la ISO/IEC 27005 específicamente trata la conceptualización general de la gestión de riesgo de la seguridad de la información (Torres Hallo, 2020).

En cuanto al **alcance y cobertura** dentro del contexto de gestión de riesgos organizacionales NIST RMF gestiona los riesgos en los sistemas de información federales y aunque está diseñado específicamente para este sector ofrece la posibilidad de ser utilizados en otros sectores para la gestión de riesgos de seguridad de la información (Force, 2018). La extensión de COBIT (Risk IT) proporciona un enfoque estructurado para gestionar los riesgos de TI, teniendo en cuenta las perspectivas y aseguramiento del negocio con TI. Está diseñado solo para abordar los riesgos de las TI, lo que lo hace muy necesario para organizaciones que dependen de la tecnología de la información (Ahmet, 2023). La ISO/IEC 27005 se puede aplicar a todo tipos de organizaciones que pretenden gestionar los riesgos que puede sufrir la seguridad de la información (*NORMA TÉCNICA COLOMBIANA NTC-ISO 27005*).

Estos marcos ofrecen **pasos y etapas de proceso** únicos, para tratar con la gestión de riesgos. NIST RMF está diseñado para gestionar los riesgos de seguridad de la información (Force, 2018). COBIT (Risk IT) se dirige específicamente a la gobernanza y gestión de los riesgos de TI (Ahmet, 2023). ISO/IEC 27005 establece un alineamiento entre los cuatro procesos de un SGSI y

sus propios procesos de ISRM («NORMA TÉCNICA COLOMBIANA NTC-ISO 27005», 2009, p. 27005).

En cuanto a **terminología y conceptos** NIST RMF se centra en los sistemas de información y la ciberseguridad (Force, 2018), COBIT (Risk IT) aborda específicamente los riesgos de TI (Ahmet, 2023). La norma ISO/IEC 27005 aplica los términos y definiciones de las normas ISO/IEC 27001 e ISO/IEC 27002 («NORMA TÉCNICA COLOMBIANA NTC-ISO 27005», 2009, p. 27005).

Los **principios y prácticas clave** tratados por NIST RMF ofrecen un proceso estructurado para gestionar los riesgos de seguridad de la información y ciberseguridad (Force, 2018), mientras que COBIT (Risk IT) es particularmente adecuado para gestionar riesgos relacionados con TI en entornos complejos (Ahmet, 2023). LA norma ISO/IEC 27005 fue concebida para la gestión del riesgo en la seguridad de la información estableciendo el contexto, evaluando y tratando los riesgos a través de un plan de tratamiento para implementar las recomendaciones y decisiones con el fin de reducir el riesgo hasta un nivel aceptable («NORMA TÉCNICA COLOMBIANA NTC-ISO 27005», 2009, p. 27005).

En cuanto a la **integración con otros marcos** NIST RMF adecuado para la integración con otros marcos de ciberseguridad, como el NIST Cybersecurity Framework (CSF) y la serie ISO/IEC 27000, además promueve la interoperabilidad con diversos sistemas de gestión (Force, 2018). Mientras que el componente de riesgo de TI de COBIT se puede integrar con otros marcos de gestión, como ITIL, PMBOK e ISO/IEC 27001 (Ahmet, 2023). La norma ISO/IEC 27005 está se integra perfectamente con ISO/IEC 27001, 27002, 31000 y con la serie de NIST («NORMA TÉCNICA COLOMBIANA NTC-ISO 27005», 2009, p. 27005).

1.4. Estudio del estado del arte

1.4.1. Sistema Informático para la Gestión de Riesgos Empresariales de Operación

El Sistema Informático para la Gestión de Riesgos Empresariales de Operación (SIGREO), gestiona el riesgo y la incertidumbre estructurados en tres etapas: identificación, estimación y control. De esta manera se propicia elevar la calidad

en el tratamiento de estos fenómenos dentro de los procesos administrativos y lograr mayor confiabilidad en la gestión administrativa empresarial (Fonseca Hernández et al., 2022).

Cuenta con un control de acceso para el cual es necesario introducir un usuario y una contraseña para acceder a cada proceso de gestión de riesgos, los cuales son administrados mediante grupos de usuarios de forma individual. Posee una etapa de identificación, donde es posible introducir los riesgos identificados, modificarlos o eliminarlos. Es posible también exportar los resultados en formato PDF. Esta etapa es la base fundamental para la posterior estimación y control de cada uno de los riesgos identificados (Fonseca Hernández et al., 2022).

Posteriormente cuenta con una etapa de estimación, donde es posible evaluar los riesgos por cada una de las rondas y expertos definidos. El proceso de estimación se realiza utilizando las técnicas Fuzzy Delphi. Para ayudar a la determinación de la evaluación por cada uno de los riesgos es posible visualizar los criterios para las variables: frecuencia y consecuencia. Además, posee una etapa de control, donde pueden establecerse para cada uno de los riesgos, elementos como objetivo de control, responsable, alternativas, medidas a aplicar, etcétera. En esta etapa es posible visualizar resultados de la etapa anterior para mejorar el acceso a la información, lo que puede resultar de mucha ayuda en el establecimiento de parámetros para el control de cada riesgo. Una vez finalizada la etapa de control, el software guarda los resultados en la sesión del usuario que inicialmente fue creada y que, además, puede ser modificada en cualquiera de sus etapas de gestión del riesgo (Fonseca Hernández et al., 2022).

1.4.2. Sistema para el Diagnóstico y Seguimiento de Riesgos en Centros de Producción de Software

El sistema está orientado a soportar la gestión sobre el proceso de gestión de riegos en los centros productores de software. Realiza tres actividades básicas: entrada, almacenamiento y salida de información. El sistema tiene como finalidad, implementar de manera ordenada y sistemática los procesos que dan solución a todo tipo de riesgo detectado, asociado a los diferentes proyectos que existen en

el Centro de Representación y Análisis de Datos (DATEC). Además, emplea el principio de seguridad informática, Control de Acceso Basado en Roles (RBAC), además cuenta con un registro de los diferentes riesgos detectados en cada proyecto del centro, así como las diferentes acciones para la mitigación de los mismos. Durante el proceso de entrada de información el sistema toma los datos que requiere para procesar la información (Fonseca, 2016)..

En la propuesta, existen datos gestionados manualmente que son aquellas que se proporcionan de forma directa por el usuario, como es la estructura organizativa del centro, los riesgos identificados, el plan de mitigación y el plan de contingencia, etc.; mientras que otros datos de forma automáticas son informaciones que provienen o son tomados de otros sistemas o módulos como datos de personas, áreas, etc. Para este sistema las informaciones derivadas del proceso de entrada de información son almacenadas en una base de datos. Finalmente, el proceso de salida de información en el sistema cuenta con un módulo de gestión de reportes; mediante dicha interfaz es posible visualizar las informaciones procesadas por el sistema y brindar los elementos necesarios para la toma de decisiones, así como la exportación de dichos reportes en el formato de almacenamiento deseado (Fonseca, 2016).

1.4.3. Riesgos de Seguridad de la Información

Gestiona los riesgos de seguridad de la información que amenazan a una organización. Permite definir los riesgos tecnológicos además de establecer qué áreas se ven involucradas y cuál es su responsabilidad para con estos riesgos. Gestiona el cambio de forma sencilla, teniendo en cuenta todos los aspectos importantes y los nuevos riesgos potenciales que deben controlarse. Posibilita alinear los riesgos tecnológicos al negocio para visualizar el riesgo real frente al apetito de riesgo definido, según se describe en el sitio oficial de ISOTools Excellence[1].

[1] Información tomada de Software Riesgos de Seguridad de la Información - Software ISO (isotools.us).

Tabla 1: Comparación entre los sistemas homólogos estudiados.

Sistemas homólogos estudiados	Inventario de Activos	Registro de Amenazas	Análisis de Riesgo	Medición de Impacto Social
Sistema Informático para la Gestión de Riesgos Empresariales de Operación	No	Si	Si	No
Sistema para el Diagnóstico y Seguimiento de Riesgos en Centros de Producción de Software	No	Si	Si	No
Riesgos de Seguridad de la Información	Si	Si	Si	No

Como resultado del estudio realizado se encontraron tres sistemas similares, el Sistema Informático para la Gestión de Riesgos Empresariales de Operación, el Sistema para el Diagnóstico y Seguimiento de Riesgos en Centros de Producción de Software y el sistema llamado Riesgos de Seguridad de la Información. En el caso de los dos primeros no cuentan con un inventario de activos, si tienen las funcionalidades de registrar amenazas y el análisis de riesgos, pero no cuenta con un método para medir el impacto social, mientras que el sistema Riesgos de Seguridad de la Información si cuenta con un inventario para activos, un registro de amenazas y análisis de riesgos, pero no cuenta con una funcionalidad de medición de impacto social. Atendiendo a los resultados del análisis realizado se determinó que ninguno de estos sistemas cuenta con los requisitos requeridos.

1.5. Metodologías de desarrollo de software

Una metodología es un conjunto de procedimientos que ayudan a los desarrolladores de software a la hora de llevar a cabo sus proyectos. Ofrecen una guía para la toma de decisiones, así como para planificarlo, gestionarlo, controlarlo y evaluarlo. La elección de la metodología a emplear es clave durante el desarrollo de un software por sus implicaciones en lo referente a efectividad, eficiencia y desempeño del producto, costo y el tiempo de desarrollo, métodos de control de calidad y de pruebas, los cuales deben ajustarse a las particularidades de cada metodología (Montero, Cevallos, & Cuesta, 2018).

1.5.1. Metodología ágil

Las metodologías ágiles se caracterizan por su flexibilidad, los proyectos en desarrollo se dividen en proyectos más pequeños, incluye una comunicación constante con el usuario, son altamente colaborativos y se adaptan con más facilidad a los cambios. De hecho, el cambio de requisitos por parte del cliente es una característica especial, así como también las entregas, revisión y retroalimentación constantes (Montero et al., 2018).

Esto quiere decir que han tenido mucho éxito en el desarrollo de sistemas donde la compañía elabora productos pequeño o mediano para sus ventas o para diseñar sistemas a la medida dentro de una organización donde el cliente quiera invertir y no haya regulaciones externas que afecten el desarrollo. Sus principios se basan en (Sommerville, 2005):

- priorización de los requisitos y evaluación de las iteraciones del mismo,
- desarrollo y entrega incremental,
- el trabajo en equipo según sus propias formas y no basado en procesos,
- adopción de requisitos cambiantes y
- mantenimiento de la simplicidad en el software y en el proceso de desarrollo

1.5.2. Metodología a utilizar

La programación extrema (XP), es el enfoque más utilizado para el desarrollo de software ágil. Usa un enfoque orientado a objetos y engloba un conjunto de cuatro reglas y prácticas que ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y prueba.

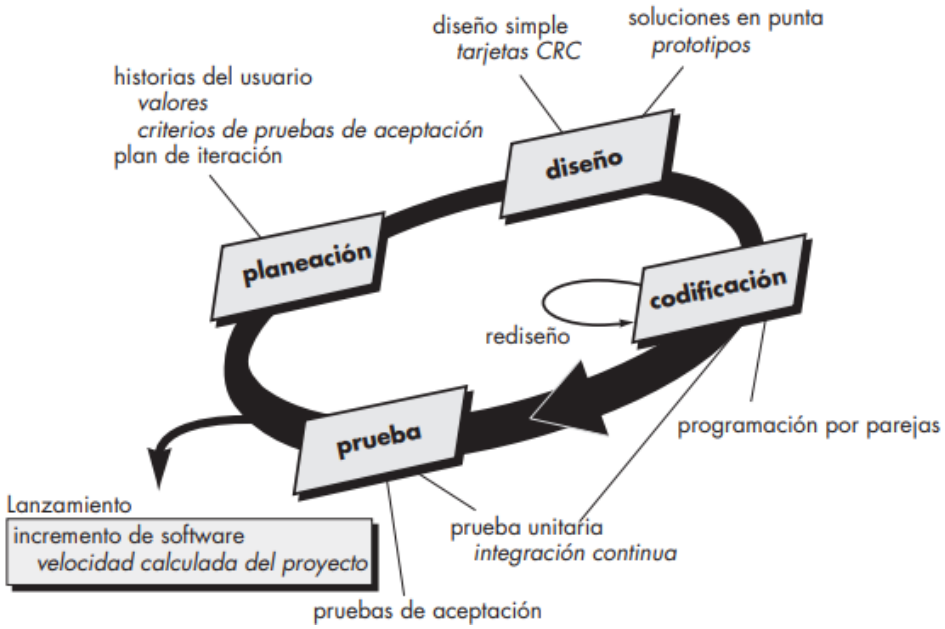


Figura 6: El proceso de la programación extrema (Pressman, 2010b).

Es la más indicada para el desarrollo del sistema porque tiene el objetivo de guiar equipos de trabajo pequeños o medianos, en ambientes de requerimientos imprecisos o cambiantes. En el desarrollo del software se realizará un proceso denominado Planning Game, en el cual se acuerda la fecha de cumplimiento y el alcance de una entrega funcional, el cliente define las historias de usuario y el desarrollador con base en ellas establecerá las características de la entrega, costos de implementación y número de interacciones para terminarla. Se realizarán entregas pequeñas que son el uso de ciclos cortos de desarrollo, llamado iteraciones, que muestra al cliente una funcionalidad del software terminado y para obtener una retroalimentación de él. Para terminar, se realizan las pruebas de aceptación; cada vez que concluya el desarrollo de una funcionalidad, entrará a pruebas por parte del cliente.

1.6. Herramientas, lenguajes y tecnologías

En este epígrafe se describen las herramientas tecnologías y lenguajes seleccionadas por el autor para llevar a cabo la investigación y la propuesta de solución. Se realizó un estudio previo en el que se determinaron las más adecuadas para el desarrollo de aplicaciones web.

1.6.1. Entorno de desarrollo

Visual Studio Code v1.82.2

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es una herramienta que ofrece una amplia gama de características y extensiones que facilitan la escritura y edición de código. Visual Studio Code es multiplataforma, lo que significa que está disponible para Windows, macOS y Linux. Es altamente personalizable y permite a los usuarios adaptar el entorno de desarrollo según sus necesidades y preferencias (Arreaga Manzaba & Chiquito Jaime, 2022).

VS Code posee una buena integración con Git, cuenta con soporte para depuración de código, y además dispone de un sinnúmero de extensiones, que brindan la posibilidad de escribir y ejecutar código en cualquier lenguaje de programación. Inicialmente incluye un mínimo de componentes y funciones básicas de un editor con soporte nativo para JavaScript/TypeScript y Node.js, sin embargo, es personalizable con plugins o extensiones disponibles para escribir código en diferentes lenguajes. VS Code incluye una terminal con todas las funciones, la cual se inicia fácilmente en el directorio de trabajo. La terminal integrada puede utilizar cualquier Shell instalado en el equipo, como PowerShell, Bash o cualquier otro (Alvarez Gómez, 2022).

1.6.2. Framework

Django v4.1.3

Django es un marco de trabajo para el desarrollo de aplicaciones web usando Python. Django considera algunas funcionalidades listas para usar para facilitar el desarrollo de aplicaciones web. Como resultado, no es necesario escribir todo el código ni usar tiempo para buscar errores de código en el marco de trabajo. Mediante Django, el desarrollo de sistemas de información web puede ser rápido, seguro, escalable y también fáciles de mantener. Django representa un marco de trabajo para el desarrollo rápido de sistemas de información web con el lenguaje Python (Vidal-Silva et al., 2021).

1.6.3. Lenguaje de programación

Python v3.9.2

Este lenguaje ofrece muchos beneficios por lo que se ha convertido en el más utilizado para el aprendizaje profundo. Las principales ventajas de Python son: la facilidad de uso y construcción de herramientas de análisis, su versatilidad, el crecimiento de la comunidad de usuarios y ser de gran utilidad (Fernández García & Arévalo Álvarez, 2022). Se integra perfectamente con Django, ya que este marco de trabajo fue creado sobre la base de este lenguaje. Python, además, verifica la exactitud del código y permite probar las líneas sin editar, guardar y ejecutar un archivo fuente e inspecciona estructuras de datos (Forcier, Bissex, & Chun, 2008).

HTML 5

HTML, es un lenguaje de marcado de hipertexto (por sus siglas en inglés HyperText Markup Language), y se emplea para el desarrollo de aplicaciones web. No se trata de un lenguaje de programación puesto que no contiene funciones aritméticas, estructuras de control, entre otras características de un lenguaje de programación. HTML genera aplicaciones web estáticas, aunque en conjunto con diferentes lenguajes de programación se pueden crear aplicaciones dinámicas (Pardo, Tapia, Moreno, & Sánchez, 2018).

CSS 3

CSS es un leguaje de estilos utilizado para definir la presentación, el formato y la apariencia de una página web. Las hojas de estilos se crean por la necesidad de diseñar la información, donde se podrá separar el contenido de la presentación, y, así, por una misma fuente de información, generalmente definida mediante lenguaje de marcaje, ofrecer presentaciones en función de dispositivos, servicios, contextos o aplicativos (Guapi Auquilla, 2018).

JavaScript

JavaScript es un lenguaje de script, creado con el fin de posibilitar páginas web más dinámicas. Hoy en día es uno de los lenguajes de programación más importantes en el mundo de la informática, ya que se utiliza en la gran mayoría de

los sitios web en la actualidad. Una ventaja de JavaScript es que se incrusta fácilmente dentro de HTML, el lenguaje de marcado empleado para codificar páginas web. Además, es uno de los lenguajes de programación más poderosos, extendidos y flexibles que existen actualmente (Krohn, 2019).

1.6.4. Gestor de base de datos

PostgreSQL v11.0.2

PostgreSQL es un gestor de bases de datos orientadas a objetos muy conocido y usado en entornos de software libre porque cumple los estándares SQL92 y SQL99, y también por el conjunto de funcionalidades avanzadas que soporta, lo que lo sitúa al mismo o a un mejor nivel que muchos Sistemas de Gestión de Bases de Datos comerciales. Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario. Es compatible con muchos lenguajes de programación tales como C, C++, Java, Perl, PHP, Python, TCL, entre otros. Es altamente confiable en cuanto a estabilidad. Además, puede extenderse con librerías externas para soportar encriptación, búsquedas por similitud fonética, etc (Ginestà & Mora, 2012).

1.6.5. Servidor web

Apache v2.4

Apache es un extraordinario servidor web (servicio para el protocolo HTTP) que tiene una participación superior al 60% de los servidores en el mundo. Se caracteriza por ser estable, multiplataforma, modular y altamente configurable por lo cual se puede adaptar para satisfacer diferentes necesidades. Registra los diferentes eventos que ocurren cuando está en funcionamiento utilizando archivos log, de esta forma es más fácil la obtención de estadísticas son usadas para la toma de decisiones por parte del administrador. La configuración de Apache se realiza mediante la edición del archivo de texto httpd.conf, en el cual se almacenan todas las instrucciones que debe seguir para su funcionamiento (Montoya, Uribe, & Rodríguez, 2013).

1.6.6. Herramienta CASE

Las herramientas CASE (del inglés Computer Aided Software Engineering, Ingeniería de Software Asistido por Computadora) son aplicaciones informáticas creadas para aumentar la productividad y la calidad en el desarrollo del software reduciendo los costos en términos de tiempo y dinero. Estas herramientas posibilitan una mejor organización y control del desarrollo de un sistema informático, especialmente aquellos sistemas que sean grandes o robustos (Cama, 2021).

Visual Paradigm v16.2

Visual Paradigm es una herramienta profesional que soporta el ciclo de vida completo del desarrollo de software: análisis, diseño, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Agiliza la construcción de aplicaciones con calidad y a un menor coste. Permite la generación de bases de datos, transformación de diagramas de Entidad-Relación en tablas de base de datos, así como ingeniería inversa de bases de datos (Hernández, Peña, Valdés, & Cornelio, 2016).

1.6.7. Herramienta de Prueba

StresStimulus v5.8.8685

Es una herramienta de pruebas de carga, estrés y rendimiento para sitios web, aplicaciones web, API web y aplicaciones móviles de cualquier complejidad. Se utiliza desde el desarrollo inicial hasta la implementación; antes de las versiones, eventos de uso máximo y de forma continua. StresStimulus registra escenarios de aplicación y luego los reproduce para emular el impacto de los usuarios en la infraestructura web mediante la creación de usuarios virtuales (VU). Mientras las VU acceden a las páginas web simultáneamente, StresStimulus monitorea el funcionamiento del servidor y analiza el tiempo de respuesta desde la perspectiva de los usuarios finales. Los análisis de rendimiento recopilados durante las pruebas se utilizan para tomar decisiones técnicas y empresariales correctas que

afectan a la capacidad de respuesta de las aplicaciones, según describe en su página oficial[2].

1.6.8. Gestor Bibliográfico

Zotero v5.0.82

Zotero es un gestor de citas bibliográficas que funciona como un complemento de Firefox o como aplicación independiente. Zotero, cuenta con las opciones básicas de un gestor bibliográfico donde se puede guardar referencias bibliográficas para la elaboración de trabajos, memorias, proyectos de investigación, tesis, etc. tanto de forma local como en internet (Avello Martínez, Martín Lorenzo, Díaz Castañeda, & Clavero Quintana, 2013).

Conclusiones del capítulo

Una vez concluido este capítulo se puede decir que:

- El estudio de los fundamentos teóricos-metodológicos permitió esclarecer los principales conceptos relacionados con el proceso de medición de impacto de la TI en las organizaciones.
- Luego de realizar un estudio del estado del arte se determina que ninguno cumplía todas las necesidades que buscaban satisfacer con el Sistema de Medición de Impacto de Tecnologías de la Información en las Organizaciones.
- El uso de metodologías ágiles, en este caso XP fue la seleccionada y permite disponer de una mayor variación que puede originarse a lo largo del proyecto, una mejora de la calidad y la posibilidad de presentar novedades sobre el producto que se encuentra en desarrollo.
- Además, se expusieron y describieron las principales tecnologías, lenguajes y herramientas que se proponen para desarrollar el sistema.

[2] Tomado de la página oficial de Stress Stimulus, disponible en Herramienta de prueba de carga para sitios web y aplicaciones móviles difíciles (stresstimulus.com).

CAPÍTULO II: Planeación, diseño y codificación del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones

Introducción

En el presente capítulo se realiza la descripción del proceso del negocio, así como la solución propuesta para satisfacer los requerimientos del cliente. Se especificarán los requisitos del sistema, las historias de usuario y se confeccionará el plan de iteraciones, la estimación de esfuerzos por historias de usuario y el plan de entregas. Se expondrán los prototipos de las interfaces del sistema y las tarjetas CRC (Clase-Responsabilidad-Colaboración). Se seleccionará el patrón arquitectónico con el cual se va a desarrollar el software y se identificarán los patrones de diseño utilizados en la codificación. Se determinará cual será el modelo de datos utilizado y se realizará una propuesta de infraestructura y protocolos de red para el despliegue de la aplicación.

2.1. Descripción del proceso de medición de impacto

Para poder medir el impacto de las TI en las organizaciones primero se debe realizar una valoración de los activos de TI de acuerdo con su criticidad y su importancia para la organización, teniendo en cuenta el valor de reemplazo del activo o la consecuencia para el negocio por la pérdida o compromiso de esos activos. Luego se identifican las amenazas y se estima la probabilidad de ocurrencia (riesgo). Esta información se puede obtener de los propietarios o los usuarios del activo, personal de recursos humanos, administrador de las instalaciones o de especialistas en seguridad de la información, etc.

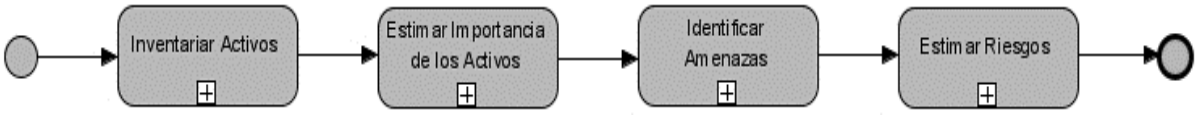


Figura 7: Descripción del proceso de negocio.

2.1.1. Actores del negocio

Los actores que interviene en el proceso de evaluación de impacto de TI dentro de las organizaciones son:

- **Especialista de TI:** Este puede ser dentro de una organización el responsable de la información, servicio, seguridad, sistema u operadores. Entre sus funciones pueden estar las determinar los niveles de seguridad de la información, analizar riesgos, implantar medidas y planes de mejora dentro de la organización.
- **Directivo:** Son los máximos responsables de los activos y servicios de TI dentro de una organización.
- **Trabajadores:** Todas aquellas personas que hacen uso de los activos y servicios de TI dentro de la organización.

2.1.1. Descripción de la solución

Luego de la fundamentación del objeto de estudio y campo de acción; de seleccionar las herramientas y metodología de desarrollo de software se puede decir que se está en condiciones de desarrollar el Sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones. Se propone el desarrollo de una aplicación web que permite realizar un inventario de activos de la infraestructura de TI. Posteriormente se estima la importancia que posee cada activo para la organización a través del procesamiento de un grupo de criterios que debe introducir el usuario. Contará de igual forma con un inventario de amenazas que puedan afectar a los activos de la organización. Tendrá una funcionalidad para estimar el riesgo que corre cada activo dentro de la organización calculando el riesgo total que corre cada activo. Contará con una función para estimar el impacto para la empresa la posibilidad de que los activos se vean afectados por las amenazas a las que está expuesto. Por último, los trabajadores de la organización tendrán la posibilidad de acceder al sistema para valorar su nivel de satisfacción con los servicios de la empresa.

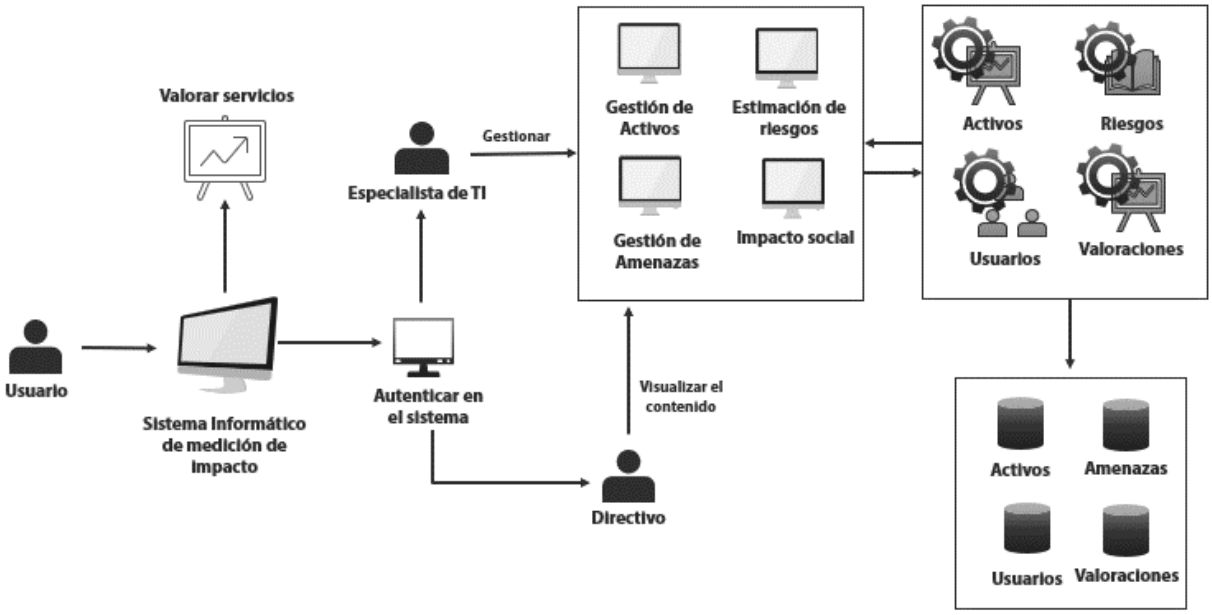


Figura 8: Descripción de la propuesta de solución.

2.2. Planeación de la solución

En este epígrafe se especifican los requisitos funcionales y no funcionales de la propuesta de solución. El desarrollador se basa en las historias de usuario para establecer las características de la entrega, los costos de implementación y la cantidad de iteraciones que son necesarias para terminar el producto. Por último, se definen las fechas de cumplimiento de cada iteración.

2.2.1. Especificación de requisitos

La especificación de requisitos es una de las tareas más importantes en el desarrollo de un software debido a que en ella se determinan los planos de la nueva aplicación. En este documento deben identificarse y detallarse de manera clara los requisitos del sistema. Tiene como principales objetivo ayudar a los clientes a describir lo que se desea obtener mediante un software y ayudar a los desarrolladores a entender qué quiere exactamente el cliente además de definir una base fija en la que trabajar (Agut, 2001).

2.2.1.1. Requisitos funcionales

Los requisitos funcionales (RF) que a continuación se presentan, describen las funciones del sistema informático a desarrollar.

Tabla 2: Tabla de requisitos funcionales.

No. RF	Nombre del RF	Descripción	Prioridad	Complejidad
RF 1	Autenticar Usuario	El usuario podrá iniciar sesión para acceder en el sistema	Alta	Baja
RF 2	Insertar activos	El usuario podrá insertar un activo en el sistema	Alta	Media
RF 3	Modificar activos	El usuario podrá modificar un activo previamente ingresado	Alta	Media
RF 4	Mostrar activos	El usuario podrá visualizar los activos previamente ingresados	Alta	Media
RF 5	Eliminar activos	El usuario podrá eliminar un activo previamente ingresado	Alta	Media
RF 6	Calcular importancia	El usuario podrá estimar la importancia que tiene un activo	Alta	Alta
RF 7	Insertar amenazas	El usuario podrá insertar una amenaza en el sistema	Alta	Media
RF 8	Modificar amenazas	El usuario podrá modificar las amenazas almacenadas	Alta	Media
RF 9	Mostrar amenazas	El usuario podrá visualizar las amenazas almacenadas	Alta	Media
RF 10	Eliminar amenazas	El usuario podrá eliminar las amenazas almacenadas	Alta	Media
RF 11	Estimar Riesgos	El usuario podrá estimar el riesgo de que el activo se vea afectado por las amenazas	Media	Alta
RF 12	Medir impacto	El usuario podrá visualizar el impacto para la entidad de que un activo de la misma sea vulnerado	Alta	Alta
RF 13	Valorar Servicios	El usuario podrá acceder al sistema e insertar una valoración y un comentario acerca de los servicios con los que cuenta la entidad	Alta	Media
RF 14	Medir impacto social	El usuario podrá acceder a las valoraciones y comentarios registrados sobre los servicios con los que cuenta la entidad para medir el impacto de los mismos en los trabajadores	Alta	Media

2.2.1.2. Requisitos no funcionales

Los requisitos no funcionales (RNF) que a continuación se presentan, determinan las restricciones sobre el sistema a desarrollar.

Tabla 3: Tabla de requisitos no funcionales

No. RNF	Tipo	Descripción
RNF 1	Apariencia	El sistema debe tener una interfaz fácil de usar y amigable para el usuario.
RNF 2	Software	Sistema Operativo Windows10 o superior. Sistema Operativo Linux distribución Ubuntu 20.04 o superior.
RNF 3	Seguridad	El sistema debe asegurar que los datos estén protegidos del acceso no autorizado.
RNF 4	Hardware	Capacidad de Disco Duro igual o superior a 80 Gigabytes. Se requiere un mínimo de 1024 MB de RAM y 1.8 GHz de velocidad de procesamiento en la PC del cliente.
RNF 5	Portabilidad	El sistema debe ser adaptado de forma efectiva y eficiente a diferentes entornos.
RNF 6	Rendimiento	Toda funcionalidad del sistema debe responder al usuario en menos de 5 segundos.
RNF 7	Usabilidad	El sistema debe poder ser operado y controlado con facilidad. El sistema debe proteger al usuario de cometer errores.
RNF 8	Mantenibilidad	El sistema debe poder ser modificado de forma eficiente sin introducir defectos o degradar el desempeño.

2.2.2. Historias de usuario

Las historias de usuarios son herramientas que describen las funcionalidades del software dentro de la ingeniería de requisitos ágil. Estas expresan el punto de vista

del usuario y proporcionan la documentación necesaria fomentando el debate entre los interesados y el equipo de desarrollo (Menzinsky et al., 2018).

Tabla 4: Historia de usuario Insertar Activo

Historia de usuario	
Número: 2	**Nombre:** Insertar activos
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: Una vez dentro de la página de Gestionar activos el usuario tendrá la posibilidad de insertar un nuevo activo ingresando los datos solicitados	

Tabla 5: Historia de usuario Calcular importancia

Historia de usuario	
Número: 6	**Nombre:** Calcular importancia
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 5 puntos
Complejidad: Alta	**Puntos reales:** 5 puntos
Descripción: El usuario procederá a insertar valores del 1 al 10 que representen costo, funcionalidad, imagen, confidencialidad, integridad y disponibilidad. Luego con estos valores el sistema calculará la importancia de los activos.	

Tabla 6: Historia de usuario medir impacto

Historia de usuario	
Número: 12	**Nombre:** Medir impacto
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 3 puntos
Complejidad: Alta	**Puntos reales:** 3 puntos
Descripción: El usuario podrá visualizar el impacto para la entidad de que un activo de la misma sea vulnerado	

2.2.2.1 Estimación de esfuerzo por historias de usuario

Para la implementación de la propuesta de solución se realizó una estimación de esfuerzo para cada una de las historias de usuarios descritas. A continuación, se muestran los resultados obtenidos, estableciendo el patrón de medición de que cada punto estimado corresponde a un día de trabajo, la jornada laboral será de 8 horas y se trabajarán 5 días por semana.

Tabla 7: Estimación de esfuerzo con historias de usuarios

Historias de Usuario	Puntos estimados
Autenticar Usuario	1 punto
Insertar activos	2 puntos
Modificar activos	2 puntos
Mostrar activos	2 punto
Eliminar activos	2 punto
Calcular importancia	5 puntos
Insertar amenazas	2 puntos
Modificar amenazas	2 puntos
Mostrar amenazas	2 punto
Eliminar amenazas	2 punto
Medir impacto	3 puntos
Estimar Riesgos	5 puntos
Medir impacto social	2 puntos
Valorar Servicios	2 puntos

2.2.3. Plan de iteraciones

Como parte del proceso de implementación, utilizando una metodología ágil XP se estableció un plan de iteraciones el cual tiene como objetivo mostrar las historias de usuario que serán implementadas, así como el orden y duración de las mismas, y agruparlas en iteraciones. En este proceso se definieron 3 iteraciones sin exceder el tiempo de 3 semanas de duración por iteración, como se muestra a continuación.

Tabla 8: Plan de iteración

Iteración	Historias de Usuario	Duración de la iteración
1	• Autenticar Usuario • Insertar activos • Modificar activos • Mostrar activos • Eliminar activos • Calcular importancia	14 puntos estimados equivalen a 2 semana y 4 días
2	• Insertar amenazas • Modificar amenazas • Mostrar amenazas • Eliminar amenazas • Medir impacto	11 puntos estimados equivalen a 2 semanas y 1 días
3	• Estimar Riesgos	9 puntos estimados equivalen

	• Medir impacto social • Valorar Servicios	a 1 semana y 4 días

2.2.3.1. Plan de entregas

El plan de entregas es un cronograma que establece qué historias de usuario serán agrupadas para conformar una entrega, y el orden de las mismas. Este cronograma será el resultado de una reunión entre todos los actores del proyecto. El plan de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores (Joskowicz, 2008).

El plan de entrega del proyecto quedó definido de la siguiente manera:

Tabla 9: Plan de entregas

Inicio	Fin de la 1ra Iteración	Fin de la 2da Iteración	Fin de la 3ra Iteración
1/09/2023	19/09/2023	4/10/2023	13/10/2023

2.4. Diseño de la solución

2.4.1. Tarjetas CRC

Las tarjetas CRC son un mecanismo utilizado por la metodología XP para pensar en el software en un contexto orientado a objetos. Estas tarjetas identifican y organizan las clases orientadas a objetos que son relevantes para el incremento actual de software (Pressman, 2010b).

A continuación, se muestran las tarjetas CRC correspondientes a las clases del software, la responsabilidad de cada una en el funcionamiento del sistema y las clases que colaboran con la misma para la realización de sus responsabilidades.

Tabla 10: Tarjeta CRC de la clase Usuario

Clase: Usuario	
Responsabilidad	**Colaboración**
• Almacenar los datos de los usuarios registrados en el sistema	• Activos • Amenazas • Valoraciones

Tabla 11: Tarjeta CRC de la clase Valoraciones

Clase: Valoraciones	
Responsabilidad	**Colaboración**

• Insertar y almacenar las valoraciones y comentarios agregadas por los usuarios	• Activos

Tabla 12: Tarjeta CRC de la clase Activos

Clase: Activos	
Responsabilidad	**Colaboración**
• Gestionar y almacenar los datos de los activos registrados en el sistema • Procesar datos de criterios para determinar la importancia de los activos • Determinar la importancia de los activos registrados	• Amenazas

Tabla 13: Tarjeta CRC de la clase Amenaza

Clase: Amenazas	
Responsabilidad	**Colaboración**
• Gestionar y almacenar los datos de las amenazas registradas en el sistema	• Activos

2.4.2. Prototipos

Los prototipos se utilizan para poner a disposición del usuario una serie de requisitos. Este conocimiento se utiliza para especificar, diseñar y desarrollar la aplicación (Gutierrez, 2011). A continuación, se muestran algunos de los prototipos de las interfaces del sistema:

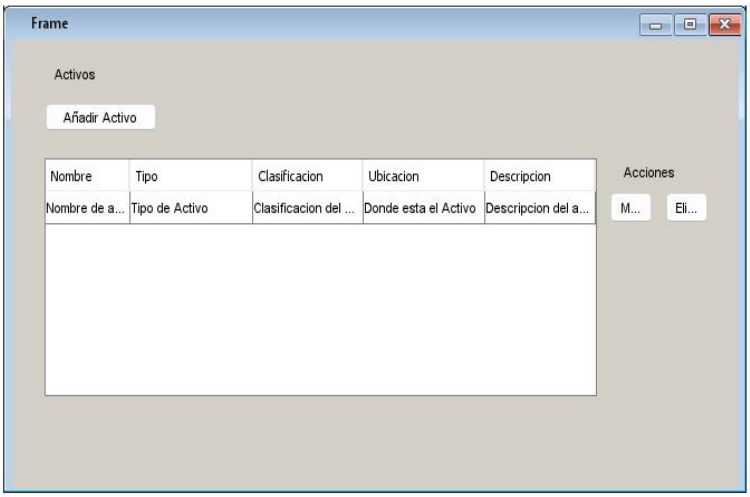


Figura 9: Prototipo de Mostrar Activos

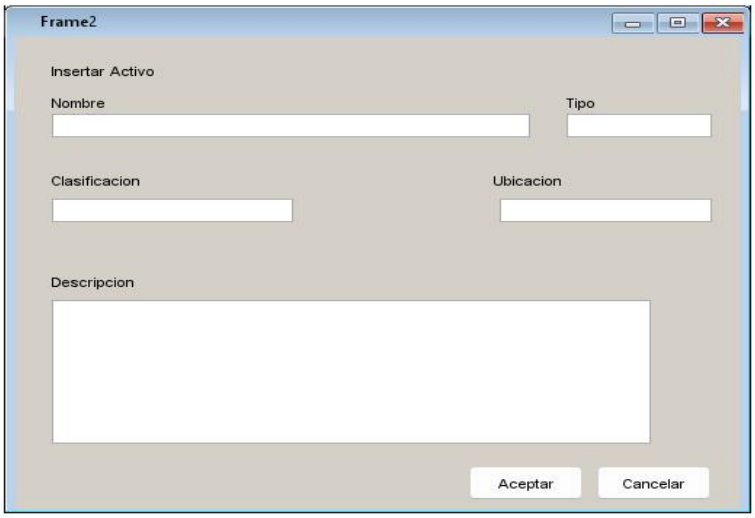


Figura 10: Prototipo de Insertar Activo

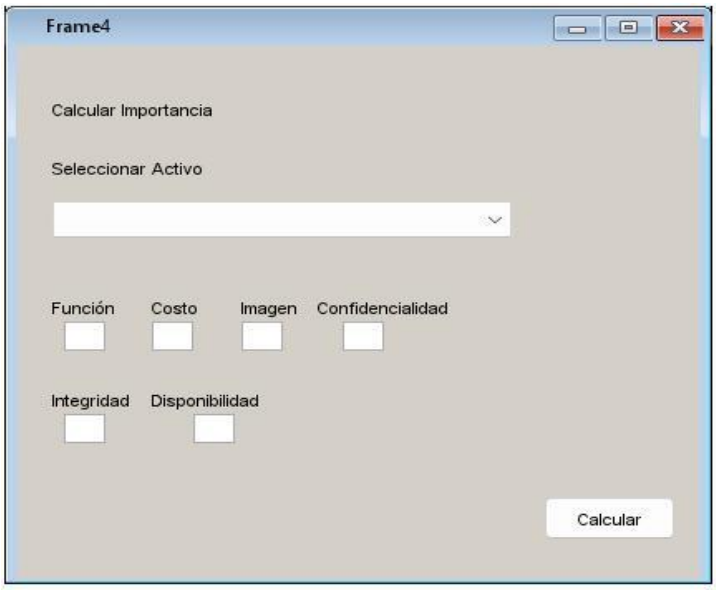


Figura 11: Prototipo de Calcular Importancia

2.5. Patrón arquitectónico

Modelo-Vista-Controlador

El Modelo-Vista-Controlador (MVC) es un patrón arquitectónico que se creó como una solución sencilla y potente para separar el frontend y el backend de una aplicación. Su funcionamiento se fundamenta en la separación del código en tres capas distintas y con una responsabilidad definida; modelos, vistas y controladores. En el caso del marco de trabajo Django, este patrón lo redefine como Modelo-Vista-Plantilla, tratando la Vista como Controlador y los Plantilla como las Vistas (Alonso-Aranda, 2019).

Modelo: Se trata de la parte del sistema que maneja directamente los datos, es decir, el que realiza las operaciones para obtener los resultados, por tanto, contendrá aquellos submódulos/clases necesarios para acceder, mostrar o refrescar dicha información (Alonso-Aranda, 2019).

Vista: Las vistas contienen aquellos módulos que se van a encargar de materializar las interfaces de usuario de nuestra aplicación, de modo que siempre mostrarán la información más actualizada, así el programador puede despreocuparse de ellas, ya que es el módulo quien se encarga automáticamente de dicha actualización (Alonso-Aranda, 2019).

Controlador: El controlador es el enlace imprescindible entre las vistas y los modelos, de modo que sirve para dar respuesta a la comunicación bidireccional entre ambos elementos. El controlador tiene la implementación del código necesario para dar respuesta a las peticiones que llegan desde la aplicación, que normalmente vienen disparadas por el usuario, como pueden ser mostrar u ocultar una barra de tareas, pinchar en cualquier enlace de una aplicación web, dar respuesta a cualquier botón de una aplicación de escritorio, etc. (Alonso-Aranda, 2019).

A continuación, se muestra la representación del patrón Modelo-Vista-Plantilla en el cual se van colocar las clases entidad en el Modelo, las clases controladoras en la Vista y las clases interfaz en la Plantilla. Cada clase tiene sus métodos y/o atributos correspondientes.

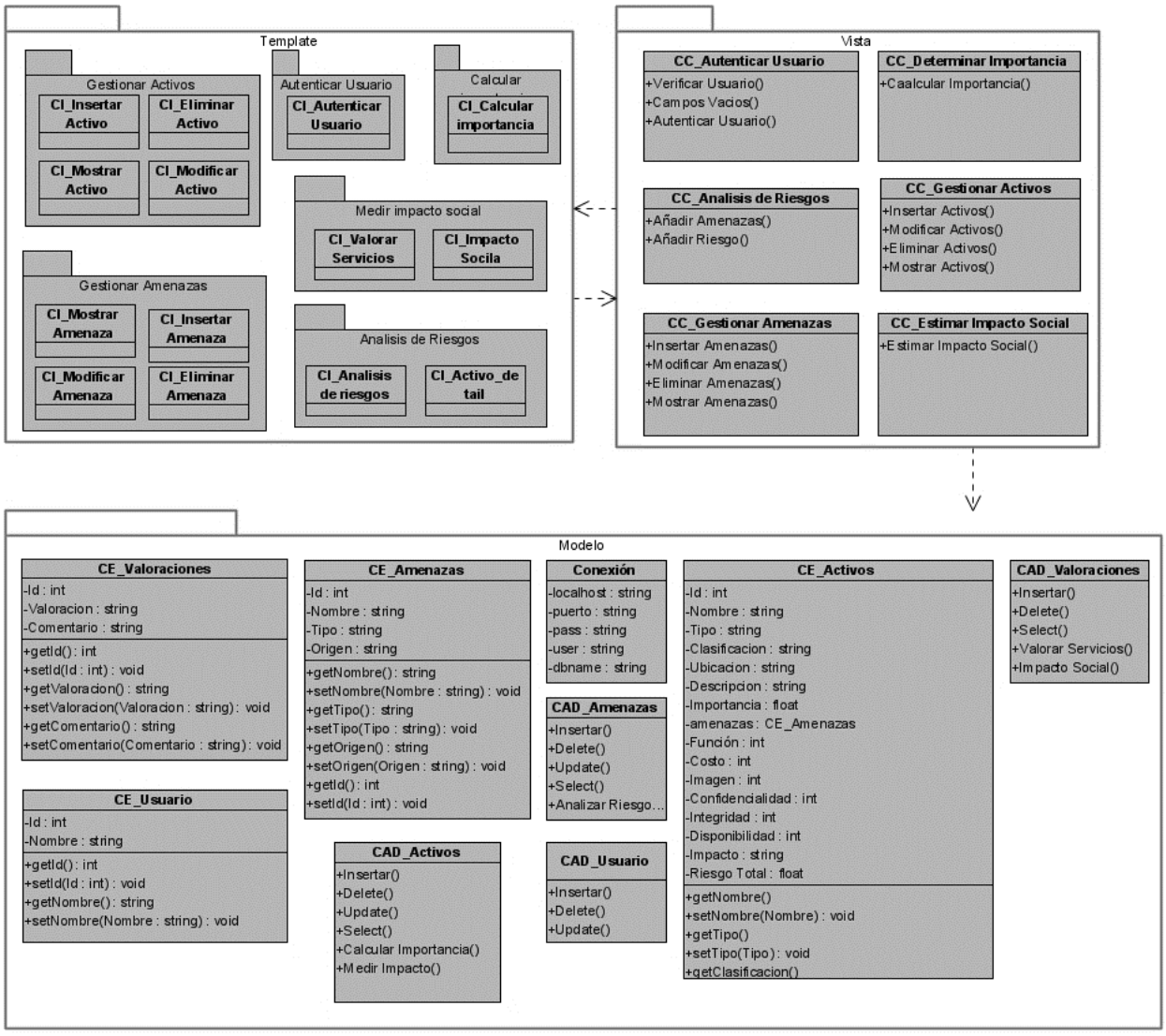


Figura 12: Representación del patrón modelo-vista-controlador de la propuesta de solución.

2.6. Patrones de diseño

Los patrones de diseños son reglas que expresan una relación entre un contexto determinado, un problema y su solución. Para el diseño de software, el contexto posibilita entender el ambiente en el que se encuentra el problema y que solución sería la adecuada para resolverlo. Los patrones de diseño incorporan el conocimiento de diseño pragmático en una forma que permite ser reutilizado muchas veces (Pressman, 2010a).

2.6.1. GRASP

Los patrones GRASP son una serie de buenas prácticas en el diseño de software. Entre ellos se encuentran los patrones Experto, Creador, Bajo acoplamiento, Controlador y Alta cohesión. Los patrones GRASP (Patrones Generales de Software para Asignar Responsabilidades por sus siglas en inglés General Responsibility Assignment Software Patterns) describen los principios fundamentales del diseño de objetos y la asignación de responsabilidades, expresados como patrones.

Los patrones de diseño GRASP son (Pedroso Estrada & Hernández Carmona, 2018):

- **Experto**: El patrón Experto se encarga de asignar una responsabilidad al experto en información: la clase que cuenta con la información necesaria para cumplir la responsabilidad. Mantienen el encapsulamiento de la información, puesto que los objetivos utilizan su propia información para llevar a cabo tareas.

  En la siguiente figura se muestra un ejemplo de patrón experto utilizado para definir el método str de una clase, que se encarga de devolver una representación legible del objeto.

```
class Amenazas(models.Model):
    id=models.AutoField(primary_key=True)
    nombre=models.CharField(max_length=100, verbose_name="Nombre")
    tipo=models.CharField(max_length=500, verbose_name="Tipo", null=True)
    origen=models.CharField(max_length=500, verbose_name="Origen", null=True)

    def __str__(self):
        return self.nombre
```

Figura 13: Ejemplo en el código del patrón experto

  Creador: Guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El propósito fundamental es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento, constituyendo su principal beneficio, lo cual supone menos dependencias respecto al mantenimiento y mejores

oportunidades de reutilización. En la siguiente figura se muestra un ejemplo de patrón creador utilizado para crear una instancia de un modelo si no existe, o modificarla en el caso de que ya exista.

```
if form.is_valid() and request.method == "POST":
    ActivosAmenazas.objects.filter(activos_id=pk).delete()
    for amenaza in form.cleaned_data["amenazas"]:
        riesgo = form.cleaned_data[f"riesgo_{amenaza.nombre}"]
        ActivosAmenazas.objects.update_or_create(
            activos_id=activo,
            amenazas_id=amenaza,
            riesgo=riesgo
        )
```

Figura 14: Ejemplo en el código del patrón creador

Bajo Acoplamiento: El patrón bajo acoplamiento se utiliza para minimizar las dependencias entre las clases, para facilitar el mantenimiento, la reutilización y la extensibilidad del código. En la siguiente figura se muestra un ejemplo de patrón bajo acoplamiento en el cual tenemos un atributo que es una instancia de otra clase, y se establece una relación muchos a muchos para crear una asociación entre ambas clases, sin que una de ellas tenga que conocer la implementación de la otra.

```
class Activos(models.Model):
    id=models.AutoField(primary_key=True)
    nombre=models.CharField(max_length=100, verbose_name="Nombre")
    tipo=models.CharField(max_length=100, verbose_name="Tipo")
    clasificación=models.CharField(max_length=100, verbose_name="Clasificación")
    ubicacion=models.CharField(max_length=100, verbose_name="Ubicación")
    descripcion=models.CharField(max_length=500, verbose_name="Descripción", null=True)
    amenazas=models.ManyToManyField(Amenazas, related_name="activos", through='ActivosAmenazas' )
```

Figura 15: Ejemplo en el código del patrón bajo acoplamiento

Controlador: El patrón controlador sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa, de tal forma que es la que recibe los datos del usuario y la que los envía a las distintas clases según el método llamado, sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto para aumentar la reutilización de código y a la vez tener un mayor control y es un objeto de interfaz no destinada al usuario que se encarga de manejar un evento del sistema. En la siguiente figura se muestra un ejemplo de patrón controlador utilizado para recibir la entrada del

usuario, acceder al modelo si es necesario, y seleccionar la plantilla adecuada para mostrar los datos.

```python
def activo_detail(request, pk):
    activo = Activos.objects.get(pk=pk)
    riesgos = ActivosAmenazas.objects.filter(activos_id=activo)
    return render(request, "AnalisisRiesgo/activo_detail.html", {"activo": activo, "riesgos": riesgos})
```

Figura 16: Ejemplo en el código del patrón controlador

- **Alta Cohesión**: En la perspectiva del diseño orientado a objetos, la cohesión funcional es una medida de cuanto están relacionadas y enfocadas las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. En la siguiente figura se muestra un ejemplo de patrón alta cohesión utilizado para hacer que una clase de un formulario sea altamente cohesiva, usando solo los atributos y métodos que sean relevantes para la representación y el procesamiento de dicho formulario.

```python
class Activos_AmenazasForm(forms.ModelForm):
    class Meta:
        model = Activos
        fields = 'amenazas',
        widgets={
            'amenazas':forms.CheckboxSelectMultiple()
        }
```

Figura 17: Ejemplo en el código del patrón alta cohesión

2.6.2. GOF

Los Patrones Gang of Four (GOF) describen las formas comunes en que diferentes tipos de objetos pueden ser organizados para trabajar unos con otros. Se clasifican en 3 grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento (Páez Lapchinskiy & Marrero González, 2019).

- **Creacionales**: describen las formas de crear instancias de objetos. El objetivo de estos patrones es abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados. En este caso se utilizó el patrón Builder, que separa la construcción de un objeto complejo de su representación, permitiendo que el mismo proceso de construcción pueda

crear diferentes representaciones. Este patrón se puede usar en Django para crear objetos de formularios personalizados, usando el método init de la clase base Form.

```python
class Activos_AmenazasForm(forms.ModelForm):
    class Meta:
        model = Activos
        fields = 'amenazas',
        widgets={
            'amenazas':forms.CheckboxSelectMultiple()
        }

    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        # Recorrer todas las opciones de amenazas disponibles
        for amenaza in Amenazas.objects.all():
            # Añadir un campo de riesgo con el nombre de la amenaza como prefijo
            self.fields[f"riesgo_{amenaza.nombre}"] = forms.DecimalField( max_value=1, min_value=0,
                label=f"Riesgo de {amenaza.nombre}",
                required=False # Hacer el campo opcional
            )
```

Figura 18: Ejemplo en el código del patrón GOF builder

- **Estructurales**: describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades. Estos objetos adicionados pueden ser incluso objetos simples u objetos. En este caso se utilizó el patrón Decorator, que es un patrón que permite añadir funcionalidad a una función o clase sin modificar su código original. Este patrón se puede usar en Django para crear decoradores, que son funciones que modifican el comportamiento de otras funciones

```python
@login_required
def principal(request):
    return render(request, 'principal.html')
```

Figura 19: Ejemplo en el código del patrón GOF decorator

2.7. Modelo de datos

Un modelo de datos se puede interpretar como un esquema que especifica las expresiones permitidas por el propio modelo, comunica las reglas y definiciones esenciales de los datos a los usuarios; además es un esquema lógico que describe la semántica a través de tablas representada por una tecnología de

manipulación de datos tal como el lenguaje SQL. El modelo de datos está formado por tres elementos fundamentales (Toledo Rodríguez, 2019):

- Entidades: objetos con existencia física o conceptual.
- Atributos: características que identifican o definen una entidad.
- Relaciones: dependencias o asociaciones entre las entidades.

La mayoría de los modelos de datos se pueden manifestar a través de un diagrama entidad-relación, ya que representa las relaciones entre los objetos de datos (Toledo Rodríguez, 2019). A continuación, se muestra el modelo de datos en el que están representadas las cuatro entidades con las que cuenta el sistema, además de la relación muchos a muchos de ActivosAmenazas la cual guarda los identificadores de las entidades Activos y Amenazas, y establece la asociación entre dichas clases para determinar la probabilidad de ocurrencia de ciertas amenazas sobre un activo, la cual se almacena en el atributo riesgo. Cada entidad muestra sus atributos correspondientes.

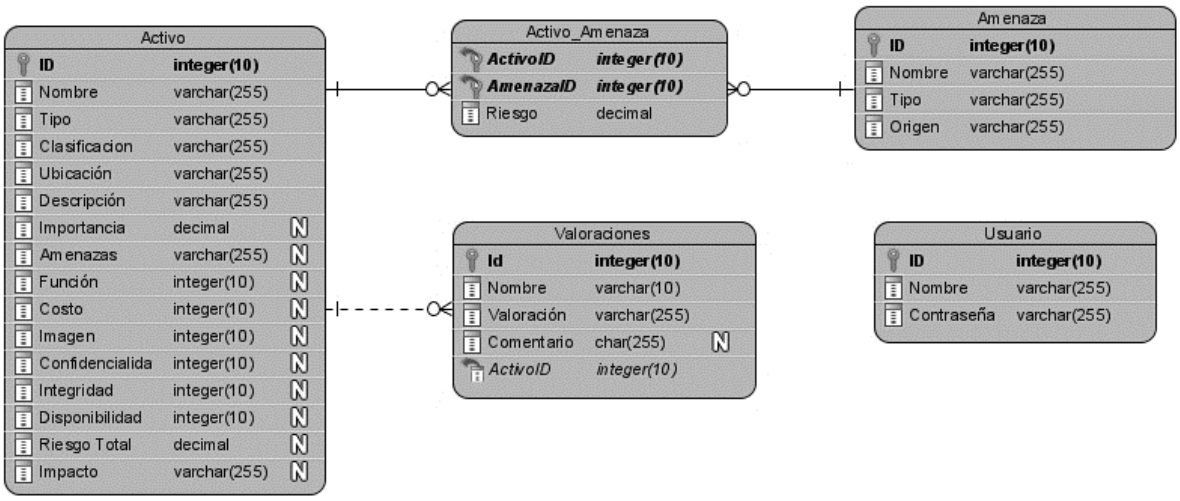


Figura 20: Modelo de datos de la propuesta de solución

2.8. Arquitectura del sistema en el despliegue del software

Al ser una aplicación web se sustenta bajo la arquitectura cliente-servidor, esta requiere de un servidor dedicado a alojar la información gestionada por el sistema. El cliente debe acceder a través de un navegador, se propone como servidor web

Apache y la conexión se debe establecer a través del protocolo HTTPS (Protocolo de transferencia de hipertexto sobre SSL/TLS) por el puerto 443, el protocolo de trasporte que se requiere para asegurar la entrega fiable y segura de la información es TCP (Protocolo de control de transmisión). En la **Figura 21** se representan los componentes y especificaciones necesarias para el despliegue de la solución.

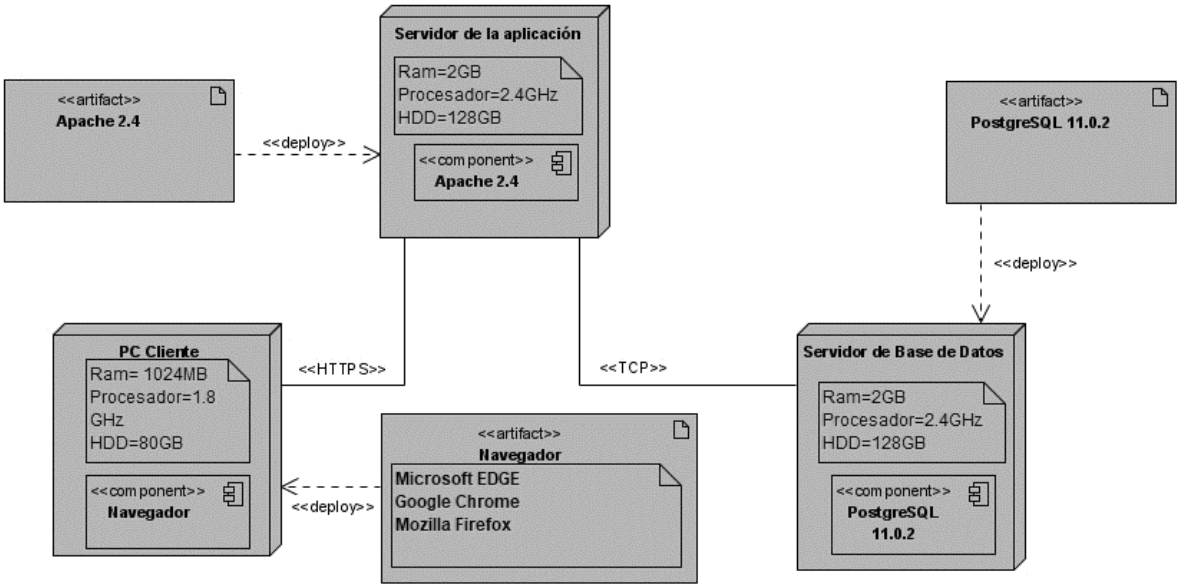


Figura 21: Diagrama de despliegue del sistema.

Conclusiones del capítulo

Una vez desarrollada la planificación, análisis y codificación del Sistema Informático para la evaluación de impacto de la infraestructura de TI en las organizaciones se llega a la conclusión de que:

- la descripción del proceso del negocio y de la solución permitió sentar las bases para describir con claridad los artefactos resultantes de la metodología;
- como parte del proceso de planificación de la solución se definieron catorce requisitos funcionales, divididos en igual número de historias de usuario y ocho requisitos no funcionales; los RF se dividieron en tres iteraciones estableciendo un tiempo estimado de desarrollo de casi siete semanas;

- como parte del proceso de diseño se identificaron cuatro tarjetas CRC y se plantearon los prototipos de interfaz de usuario para llegar tener una idea más clara de lo que solicita el cliente;
- el patrón arquitectónico modelo-vista-plantilla de Django brindó la posibilidad de optimizar la organización y la comunicación entre los diferentes componentes de la de solución propuesta;
- el uso de patrones GRASP Y GOF aumentó en gran medida la calidad del software desarrollado;
- en el modelo de datos se modelaron las relaciones entre las entidades con sus atributos;
- se propuso la infraestructura de red y los protocolos para dar soporte a la aplicación web desarrollada.

CAPÍTULO III: Validación del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones

Introducción

En este capítulo se realizan las validaciones a la solución fundamentada en el capítulo anterior. La estrategia de prueba se basa en realizar pruebas de aceptación, unitarias y rendimiento acorde con la metodología implementada y el tipo de aplicación desarrollada.

3.1. Estrategia de prueba

Una Estrategia de Prueba es un guía que describe los pasos y procesos que deben realizarse como parte de las pruebas, cuando se planean y se llevan a cabo dichos pasos, y cuánto esfuerzo, tiempo y recursos se requieren. Cualquier estrategia de prueba debe incluir la planificación de pruebas, el diseño de casos de prueba, la ejecución de la prueba, la recolección y evaluación de los resultados (Pressman, 2010a).

Objetivos de las pruebas: Las pruebas tienen como objetivo validar que el software cumple con los requisitos funcionales y no funcionales especificados, verificar que el software es fácil de usar, seguro, compatible y eficiente, detectar y corregir los defectos antes de la entrega al cliente.

Alcance de las pruebas: Las pruebas se les realizarán a todas las funcionalidades del software, como el registro de activos y amenazas, la estimación de la importancia, el análisis de riesgo y la valoración de los servicios.

Métodos de las pruebas. Los métodos de las pruebas utilizadas son los siguientes:

- **Pruebas manuales:** Son las pruebas que se realizan de forma manual siguiendo los pasos y los criterios definidos en los casos de prueba. Se utilizan para probar las funcionalidades, la usabilidad y la compatibilidad del software.

- **Pruebas automatizadas:** Son las pruebas que se realizan de forma automática por las herramientas de pruebas, siguiendo los scripts y las validaciones definidos.
- **Pruebas de Rendimiento:** Son las pruebas realizadas sobre computadoras, redes, software u otros dispositivos, son utilizados para determinar la velocidad y eficiencia de los mismos
- **Pruebas de caja negra:** Son las pruebas que se realizan sin conocer la estructura interna o el código fuente del software. Se basan en los requisitos, las especificaciones y los escenarios del software. Se realizan tanto de forma manual como automática.
- **Pruebas de caja blanca:** Son las pruebas que se realizan conociendo la estructura interna o el código fuente del software. Se basan en los flujos de control, los flujos de datos y las condiciones lógicas. Se realizan de forma automática.

3.1.1. Pruebas de aceptación

Las pruebas de aceptación de software pertenecen a las pruebas de caja negra y se utiliza para verificar si el software cumple con las expectativas y requisitos de los clientes y usuarios finales. Estas tienen como objetivo asegurar que el software sea funcional, usable y satisfactorio para los usuarios. Para ello, se utilizan criterios de aceptación que definen las condiciones que debe cumplir el software para ser aceptado (Paz, 2016).

En la **Tabla 14** se muestran las descripciones de las tareas a realizar, los pasos a seguir para comprobar el funcionamiento de cada funcionalidad y el resultado esperado después de realizar dicha prueba. Finalmente se va a documentar el resultado obtenido luego de ejecutar la prueba.

Tabla 14: Casos de prueba de aceptación.

Nombre del proyecto	
Sistema de Medición de Impacto	
Fecha de inicio de las pruebas	**Hora de inicio de la prueba**
19/9/2023	8:00 am
Fecha de finalización de la prueba	**Hora de finalización de las pruebas**
25/10/2023	5:00 pm
Nombre del probador	
Ernesto Elias Elzaurdín Valdés	

No.	**Descripción de las tareas**	**Pasos para ejecutar**	**Resultados esperados**	**Defecto / comentarios / adiciones**
Primera iteración:				
1	Prueba de la funcionalidad Autenticar Usuario	En la página inicial se selecciona el botón Iniciar Sesión y posteriormente se ingresa el usuario y la contraseña requeridos para acceder al sistema	Luego de pulsar el botón Iniciar Sesión se accederá a una página donde se introducirá el usuario y la contraseña. Si el usuario esta registrado accederá al sistema mientras que si se dejan los campos en blanco o se ponen datos incorrectos no se accederá y volverá a solicitar los datos	La prueba se superó de manera satisfactoria.
2	Prueba de la funcionalidad Insertar activos	Estando en la página principal se pulsará el botón Gestionar Activos en el menú del sidebar, dentro de esta página se pulsará el botón Agregar Activo, luego se ingresarán los datos solicitados por el sistema y se pulsará el botón aceptar.	El usuario accederá a la página Gestionar Activos y luego accederá a la funcionalidad de Agregar Activo. Posteriormente introducirá los datos requeridos para almacenar el activo. Si se dejan los campos en blanco el sistema no guardará el activo y lanzará una alerta de campos requeridos.	La prueba se superó de manera satisfactoria.
3	Prueba de la funcionalidad Modificar activos	Estando en la página principal se pulsará el botón Gestionar Activos en el menú del sidebar, dentro de esta página se pulsará el botón Editar que le corresponde a cada activo, luego se ingresarán los datos que se desean modificar y se pulsará el botón aceptar.	El usuario accederá a la página Gestionar Activos y luego accederá a la funcionalidad de Editar que le corresponde a cada activo. Posteriormente introducirán los datos que se desean modificar y se pulsará el botón Aceptar. Si se dejan los campos en blanco el sistema no guardará el activo y lanzará una alerta de campos requeridos.	La prueba se superó de manera satisfactoria.

No.	Descripción de las tareas	Pasos para ejecutar	Resultados esperados	Defecto / comentarios / adiciones
4	Prueba de la funcionalidad Mostrar activos	Estando en la página principal se pulsará el botón Gestionar Activos en el menú del sidebar.	El usuario accederá a la página Gestionar Activos donde se mostrarán todos los activos registrados en el sistema e información referente a ellos	La prueba se superó de manera satisfactoria.
5	Prueba de la funcionalidad Eliminar activos	Estando en la página principal se pulsará el botón Gestionar Activos en el menú del sidebar, dentro de esta página se pulsará el botón Borrar que le corresponde a cada activo	El usuario accederá a la página Gestionar Activos y luego accederá a la funcionalidad de Borrar que le corresponde a cada activo. Posteriormente el activo es eliminado del sistema	La prueba se superó de manera satisfactoria.
6	Prueba de la funcionalidad Calcular importancia	Estando en la página principal se pulsará el botón Calcular Importancia en el menú del sidebar, dentro de esta página se pulsará el botón Calcular Importancia que le corresponde a cada activo, una vez dentro de la nueva página el usuario ingresará los valores solicitados y pulsará el botón Aceptar	El usuario accederá a la página Calcular Importancia y luego accederá a la funcionalidad de Calcular Importancia que le corresponde a cada activo. Posteriormente el usuario ingresará los valores solicitados y pulsará el botón Aceptar. Si se dejan los campos en blanco el sistema no guardará y lanzará una alerta de campos requeridos.	Si el usuario ingresa los valores el sistema calcula la importancia y la almacena, pero en caso de dejar los campos en blanco ocurre un error que no permite que se realice la operación y debe ser corregido
Segunda iteración:				
1	Prueba de la funcionalidad Insertar amenazas	Estando en la página principal se pulsará el botón Gestionar Amenazas en el menú del sidebar, dentro de esta página se pulsará el botón Agregar Amenaza, luego se ingresarán los datos solicitados por el sistema y se pulsará el botón aceptar.	El usuario accederá a la página Gestionar Amenazas y luego accederá a la funcionalidad de Agregar Amenaza. Posteriormente introducirá los datos requeridos para registrar la amenaza. Si se dejan los campos en blanco el sistema no guardará la amenaza y lanzará una alerta de campos requeridos.	La prueba se superó de manera satisfactoria.
2	Prueba de la funcionalidad Modificar amenazas	Estando en la página principal se pulsará el botón Gestionar Amenazas en el menú del sidebar, dentro de esta página se pulsará el botón Editar que le corresponde a cada amenaza, luego se ingresarán los datos que se desean modificar y se pulsará el botón aceptar.	El usuario accederá a la página Gestionar Amenazas y luego accederá a la funcionalidad de Editar que le corresponde a cada amenaza. Posteriormente introducirán los datos que se desean modificar y se pulsará el botón Aceptar. Si se dejan los campos en blanco el sistema no guardará la amenaza y lanzará una alerta de campos requeridos.	La prueba se superó de manera satisfactoria.

No.	Descripción de las tareas	Pasos para ejecutar	Resultados esperados	Defecto / comentarios / adiciones
3	Prueba de la funcionalidad Mostrar amenazas	Estando en la página principal se pulsará el botón Gestionar Amenazas en el menú del sidebar.	El usuario accederá a la página Gestionar Amenazas donde se mostrarán todas las amenazas registradas en el sistema e información referente a ellas	La prueba se superó de manera satisfactoria.
4	Prueba de la funcionalidad Eliminar amenazas	Estando en la página principal se pulsará el botón Gestionar Amenazas en el menú del sidebar, dentro de esta página se pulsará el botón Borrar que le corresponde a cada amenaza	El usuario accederá a la página Gestionar Amenazas y luego accederá a la funcionalidad de Borrar que le corresponde a cada amenaza. Posteriormente la amenaza es eliminada del sistema	La prueba se superó de manera satisfactoria.
5	Prueba de la funcionalidad Estimar Riesgos	Estando en la página principal se pulsará el botón Análisis de Riesgos en el menú del sidebar, dentro de esta página se mostrarán todos los activos y el riesgo ser afectados, se pulsará el botón Añadir Amenazas que le corresponde a cada activo, se accederá a una página que mostrará las amenazas que afectan a dicho activo y el riesgo de que cada una lo afecten, se pulsará el botón Añadir Amenazas donde se seleccionarán las amenazas, se introducirá el riesgo de cada una y se pulsará el botón aceptar.	El usuario accederá a la página Análisis de Riesgo y visualizará cada activo con el riesgo de ser afectado, cuando se acceda a la funcionalidad de Añadir Amenazas que le corresponde a cada activo el usuario visualizará todas las amenazas que afectan al activo y sus respectivos riesgos. Posteriormente se pulsará el botón Añadir Amenazas donde se seleccionarán las amenazas y se introducirán los riesgos correspondientes a cada una y se pulsará el botón Aceptar. Si se dejan los campos en blanco el sistema no guardará los datos y lanzará una alerta de campos requeridos.	Si el usuario ingresa los valores el sistema almacena los datos y estima el riesgo total del activo, pero en caso de dejar los campos en blanco ocurre un error que no permite que se realice la operación y debe ser corregido
Tercera iteración:				
1	Prueba de la funcionalidad Valorar Servicios	El usuario desde la página inicial pulsará el botón Valorar Servicios en el menú del sidebar, dentro de esta página pulsará el botón Valorar correspondiente a cada servicio que se muestra en pantalla, luego se seleccionará la valoración que el usuario le otorga a dicho servicio y podrá dejar un comentario y se pulsará el botón aceptar.	El usuario accederá a la página de Valorar Servicios, dentro de esta página podrá visualizar todos los servicios que estén inventariados, luego al pulsar el botón Valorar correspondiente a cada servicio se mostrará en pantalla un campo donde se seleccionará la valoración que el usuario le otorga a dicho servicio y podrá dejar un comentario y se pulsará el botón aceptar.	La prueba se superó de manera satisfactoria.
2	Prueba de la funcionalidad Impacto social	Estando en la página principal se pulsará el botón Impacto Social en el menú del sidebar, dentro de esta página se mostrarán todas las valoraciones y los comentarios realizados por los usuarios.	El usuario accederá a la página Impacto Social, dentro de esta página podrá visualizar todas las valoraciones y comentarios introducidos por los usuarios además de poder borrarlos.	La prueba se superó de manera satisfactoria.

No.	Descripción de las tareas	Pasos para ejecutar	Resultados esperados	Defecto / comentarios / adiciones
3	Prueba de la funcionalidad Mostrar impacto	Estando en la página principal se pulsará el botón Análisis de Riesgos en el menú del sidebar, dentro de esta página se mostrarán todos los activos y el impacto para la entidad la afectación de dichos activos	El usuario accederá a la página Análisis de Riesgo y visualizará cada activo con el impacto que tiene para la entidad en caso de ser afectado	La prueba se superó de manera satisfactoria.

3.1.2. Pruebas unitarias

Las pruebas unitarias pertenecen a las pruebas de caja blanca y se utilizan comprobar el correcto funcionamiento de una unidad de código y hoy en día son una parte fundamental del desarrollo de software, ayudando a elevar la calidad del producto con beneficios cómo el de prevenir errores en etapas tempranas del desarrollo, legibilidad de código, confiabilidad en el producto, entre otros, ahorrando tiempo y dinero en los proyectos. (Bedoya Alzate, 2021).

Para la realización de estas pruebas de forma automatizada se utilizó Pytest, herramienta de prueba de Python. Se determinaron un conjunto de casos de pruebas para que la herramienta ejecutara y comprobara el funcionamiento del código con el objetivo de detectar errores. Como resultado de la prueba se obtuvo que la misma fue superada satisfactoriamente.

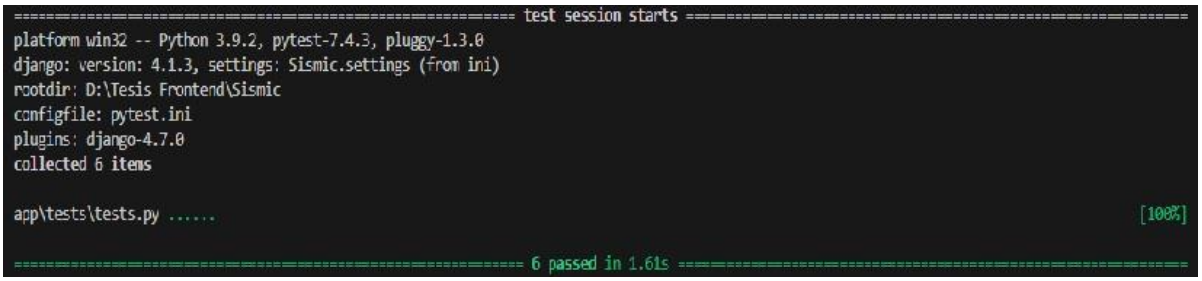


Figura 22: Resultados de las pruebas unitarias con la herramienta Pytest.

3.1.3. Prueba de rendimiento

La herramienta utilizada para realizar la prueba de rendimiento al sistema es Strees Simulator la cual graba el comportamiento de cada funcionalidad. Posteriormente realiza un análisis en el cual simula el uso del sistema por tres usuarios virtuales de manera simultánea para comprobar cómo se comporta el software en condiciones de mucho tráfico.

▼ Overall Result

Completion Status	Completed
Pass/Fail Status	Passed
Max User Load	3
Max Concurrent User Load	3
Total sent (KB)	633.549
Total received (KB)	104.052
KB sent/sec	3.016
KB received/sec	0.495

Figura 23: Resultado general de la prueba.

▼ Transactions

Avg. response time (s)	10.188
Transactions/sec	0.157
Number of transactions	14
Requested transaction iterations	33
Transactions with error(s)	0
Requested transaction iterations with error(s)	0
Transactions with timeout(s)	0
Requested transaction iterations with timeout(s)	0
Transactions with missed goal(s)	0
Requested transaction iterations with the missed goal	0

Figura 24: Número de transacciones en la prueba

El resultado general de prueba arrojó que el total de tráfico de carga enviado es de 633.5 KB/seg, mientras que el de descarga es de 104.05 KB/seg. El tiempo promedio de respuesta exitosa fue de 10.18 transacciones/seg. Es por ello que se puede afirmar que el sistema presenta tiempos de respuesta aceptable para los usuarios. En el **Anexo 3** se muestran más detalles del procedimiento.

▼ Transactions (slow to fast)

SISMIC: Insertar Amenazas	12.172
SISMIC: Editar Amenazas	12.095
SISMIC: ImpactoSocial	11.599
SISMIC: Mostrar Activos	11.378
SISMIC: Mostrar Amenazas	11.203
SISMIC: Editar Activos	11.028
SISMIC: Login	10.488
SISMIC: Eliminar Activos	10.349
SISMIC: Calcular Importancia	8.726
SISMIC: Insertar Valoracion	8.461
SISMIC: Insertar Activos	8.059
SISMIC: Analisis de Riesgos	0.015
SISMIC: Logout	0
SISMIC: Eliminar Amenazas	0

Figura 25: Transacciones de lentas a rápidas por funcionalidad.

Conclusiones del capítulo

Luego de realizadas las validaciones al sistema se puede concluir que:

- la revisión de los requisitos permitió obtener la aprobación y validación de los requisitos por parte del cliente, permitiendo también corregir inconformidades que arrojó dicho proceso de revisión,
- se llevaron a cabo pruebas unitarias mediante el método de caja blanca automatizadas y con la técnica del camino básico para comprobar la calidad estructural y la lógica interna del sistema, obteniendo resultados satisfactorios,
- se realizaron pruebas de aceptación, a través del método de caja negra con el cual se pudo comprobar el estado de las funcionalidades del sistema y
- se realizaron pruebas de rendimiento para comprobar el comportamiento del sistema en situaciones de mucho tráfico obteniéndose buenos resultados en la respuesta.

CONCLUSIONES

Una vez desarrollado el sistema informático para la medición de impacto de la infraestructura de Tecnologías de Información en las organizaciones se puede concluir que:

- la sistematización de los referentes teóricos-metodológicos que sustentan a la evaluación de impacto de las TI en las organizaciones permitió un mejor entendimiento de los términos relacionados relacionados con el objeto de estudio y el campo de acción,
- la planificación y diseño del proceso de desarrollo del sistema fue fundamental para lograr un mejor entendimiento con el cliente y encontrar puntos de mejora en el proceso de desarrollo,
- la implementación del sistema informático para la evaluación de impacto de la infraestructura de TI en las organizaciones hizo que se cumplieran los requisitos pactados con el cliente para contribuir a la reducción de la complejidad en la medición y
- la validación del sistema informático a través de una estrategia de pruebas hizo que se comprobara la calidad del producto desarrollado.

Es por ello que se puede afirmar que los objetivos propuestos para la presente investigación han sido cumplidos. El sistema desarrollado contribuirá a minimizar la complejidad en cuanto a la medición de impacto de las TI en las organizaciones.

RECOMENDACIONES

Como parte de la continuidad de la investigación, se recomienda:

- establecer una conexión entre la aplicación y algún otro sistema/servicio que envíe un reporte automáticamente en caso de que algún activo sea vulnerado,
- implementar inteligencia artificial para que el sistema realice análisis autonómico más exhaustivo y detallado,
- divulgar los resultados de la investigación mediante su presentación en eventos y su publicación en revistas científicas especializadas.
- implementar una funcionalidad que posibilite exportar datos a formato PDF,

REFERENCIAS BIBLIOGRÁFICAS

Agut, R. M. (2001). Especificación de Requisitos Software según el estándar de IEEE 830. *Universidad Jaume I. Departamento de Informática. Paper*.

Ahmet, E. F. E. (2023). A comparison of key risk management frameworks: COSO-ERM, NIST RMF, ISO 31.000, COBIT. *Denetim ve Güvence Hizmetleri Dergisi*, *3*(2), 185-205.

Alonso-Aranda, C. (2019). *MODELO-VISTA-CONTROLADOR. LENGUAJE UML*.

Alvarez Gómez, G. (2022). *Módulo de agenda virtual para un asistente personal virtual*. Universidad de las Ciencias Informáticas. Facultad 4.

Arreaga Manzaba, R. A., & Chiquito Jaime, K. E. (2022). *Desarrollo de un prototipo de pulsera de seguridad conectada a la red utilizando tecnología IOT de bajo costo y un aplicativo web para el registro de datos y envío de notificaciones con la ubicación en tiempo real mediante mensajería instantánea*. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas ….

Avilés Armijos, J. M., & Uyaguari Guartatanga, M. E. (2012). *Diseño de una política de seguridad para la empresa de Telecomunicaciones PUNTONET en la ciudad de Cuenca, en base a las normas de seguridad ISO 27001 y 27011 como líneas base para las buenas prácticas de tratamiento y seguridad de la información*.

Bernal Medina, H. C. (2022). *Análisis de vulnerabilidad en dispositivos móviles con sistema operativo Android*.

Cama, Z. Y. (2021). Desarrollo de una Herramienta CASE para el Diseño de Diagramas Entidad–Relación Extendido y su mapeo al Modelo Relacional Orientado a Estudiantes en el contexto. *INF-FCPN-PGI Revista PGI*, 210-213.

Casanova, M. P., & Calderón, C. A. (2020). Modelo para la gestión de infraestructuras de tecnologías de la información. *TecnoLógicas*, *23*(48), 32-54. Recuperado de http://www.scielo.org.co/scielo.php?pid=S0123-77992020000200032&script=sci_arttext

COBIT | Control Objectives for Information Technologies. (2023). Recuperado 29 de agosto de 2023, de ISACA website: https://www.isaca.org/resources/cobit

Computer Security Division, I. T. L. (2016a, noviembre 30). About the RMF - NIST Risk Management Framework | CSRC | CSRC. Recuperado 28 de agosto de 2023, de CSRC | NIST website: https://csrc.nist.gov/Projects/risk-management/about-rmf

Computer Security Division, I. T. L. (2016b, noviembre 30). NIST Risk Management Framework | CSRC | CSRC. Recuperado 28 de agosto de 2023, de CSRC | NIST website: https://csrc.nist.gov/Projects/risk-management

Estándares para la categorización de seguridad de Información federal y sistemas de información. (2004).

Fernández García, M. A., & Arévalo Álvarez, J. C. (2022). *Analítica de datos para hurtos a personas en la ciudad de Medellín a través de modelos de Machine Learning y Deep Learning*.

Fonseca, B. B. (2016). Sistema para el diagnóstico y seguimiento de riesgos en centros de producción de software. *Serie Científica de la Universidad de las Ciencias Informáticas*, *9*(6). Recuperado de https://publicaciones.uci.cu/index.php/serie/article/view/415

Fonseca Hernández, A. A., Hernández García, L., Núñez Torres, E., de la Oliva de Con, F., Fonseca Hernández, A. A., Hernández García, L., … de la Oliva de Con, F. (2022). Sistema Informático para la Gestión de Riesgos Empresariales de Operación (SIGREO). *Cofin Habana*, *16*(2). Recuperado de http://scielo.sld.cu/scielo.php?script=sci_abstract&pid=S2073-60612022000200005&lng=es&nrm=iso&tlng=es

Force, J. T. (2018). Risk management framework for information systems and organizations. *NIST Special Publication*, *800*, 37.

Forcier, J., Bissex, P., & Chun, W. J. (2008). *Python web development with Django*. Addison-Wesley Professional.

Ginestà, M. G., & Mora, O. P. (2012). Bases de datos en PostgreSQL. *Sl]:[sn]*.

Guapi Auquilla, M. J. (2018). *Diseño metodológico para el desarrollo de interfaces gráficas en páginas web utilizando los lenguajes HTML 5 y CSS 3*. Riobamba.

Gutierrez, D. (2011). Métodos de desarrollo de software. *Caracas: Universidad de los Andes*.

Hernández, L. R. B., Peña, D. M., Valdés, O. R., & Cornelio, O. M. (2016). Extensión de la herramienta Visual Paradigm for UML para la evaluación y corrección de Diagramas de Casos de Uso. *Serie Científica de la Universidad de las Ciencias Informáticas*, *9*(7), 7-20.

Joskowicz, J. (2008). Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*, *22*.

Karakaneva, J. (2014). Software applications security. *Trakia Journal of Sciences*, *12*(4), 419.

Krohn, H. S. (2019). Programación de buscadores en JavaScript para diccionarios digitales. *Cuadernos de Lingüística Hispánica*, (34), 109-130.

Liendo Afonso, L. C. (2023). *Optimización del proceso de reporting del análisis de impacto en el negocio en una consultora de ciberseguridad*. Recuperado de http://titula.universidadeuropea.com/handle/20.500.12880/5414

Lopez Crespo, F., Amutio Gomez, M. Á., & Candau, J. (2006). *MAGERIT – versión 2 Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información*.

Mahn, A., Topper, D., Quinn, S., & Marron, J. (2021). *Getting Started with the NIST Cybersecurity Framework: A Quick Start Guide* (N.º NIST Special Publication (SP) 1271). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.1271

Martínez Landrove, N. (2019). *Ciberseguridad y riesgo operacional en las organizaciones*. Recuperado de https://repositorio.comillas.edu/xmlui/handle/11531/42317

Menzinsky, A., López, G., Palacio, J., Sobrino, M., Álvarez, R., & Rivas, V. (2018). Historias de usuario. *Ingeniería de requisitos ágil*.

Meriah, I., & Rabai, L. B. A. (2019). Comparative Study of Ontologies Based ISO 27000 Series Security Standards. *Procedia Computer Science*, *160*, 85-92. https://doi.org/10.1016/j.procs.2019.09.447

Montero, B. M., Cevallos, H. V., & Cuesta, J. D. (2018). Metodologías ágiles frente a las tradicionales en el proceso de desarrollo de software. *Espirales revista multidisciplinaria de investigación*, *2*(17), 114-121.

Montoya, C. E. G., Uribe, C. A. C., & Rodríguez, L. E. S. (2013). Seguridad en la configuración del servidor web Apache. *INGE CUC*, *9*(2), 31-38. Recuperado de https://revistascientificas.cuc.edu.co/ingecuc/article/view/3

NORMA TÉCNICA COLOMBIANA NTC-ISO 27005. (2009). Recuperado 30 de agosto de 2023, de Dokumen.tips website: https://dokumen.tips/documents/iso-27005-espanol.html

Ochoa Palomino, A. (2019). Diseño de una Red de Seguridad Informática para la Protección del Sistema Web de un Call Center ante Ataques Informáticos Aplicando la Norma ISO 27033. *Universidad Peruana de Ciencias Aplicadas (UPC)*. https://doi.org/10.19083/tesis/625726

ORTEGA CANDEL, J. M. (2021). *Ciberseguridad. Manual práctico*. Ediciones Paraninfo, S.A.

Páez Lapchinskiy, P. E., & Marrero González, A. (2019). *Módulo Gestión de Acuerdo en la tecnología de Alfresco Community 5.2*. Universidad de las Ciencias Informáticas. Facultad 2.

Pardo, M. R. V., Tapia, J. A. H., Moreno, A. S. G., & Sánchez, L. F. V. (2018). Comparación de tendencias tecnológicas en aplicaciones web. *3c Tecnología: glosas de innovación aplicadas a la pyme*, *7*(3), 28-49.

Pedroso Estrada, J., & Hernández Carmona, I. (2018). *Sistema de reportes estadísticos para el seguimiento al egresado en la Universidad de las Ciencias Informáticas*. Universidad de las Ciencias Informáticas. Facultad 4.

Pérez Bejerano, Y. (2019). *Sistema de Gestión de Licencias del Personal Aeronáutico del Instituto de la Aeronáutica Civil de Cuba versión 2.0* (BachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1.). Universidad de las Ciencias Informáticas. Facultad 1. Recuperado de https://repositorio.uci.cu/jspui/handle/123456789/10288

Pérez Lorences, P. (2014). *Procedimiento para mejorar la gestión de tecnologías de la información en el sector empresarial cubano.* (Doctorado). Universidad Central “Marta Abreu” de Las Villas.

Pressman, R. S. (2010a). (15) Ingenieria del Software. Un Enfoque Practico 7ma edición | Leonardo Alvarado—Academia.edu. Recuperado 27 de septiembre de 2022, de https://www.academia.edu/24308956/Ingenieria_del_Software_Un_Enfoque_Practico_7ma_edici%C3%B3n

Pressman, R. S. (2010b). *Ingeniería del software. Un enfoque práctico* (7a ed.).

Quinn, S., Ivy, N., Chua, J., Barrett, M., Feldman, L., Topper, D., … Gardner, R. K. (2022). *Using business impact analysis to inform risk prioritization and response* (N.º NIST IR 8286D; p. NIST IR 8286D). Gaithersburg, MD: National Institute of Standards and Technology (U.S.). https://doi.org/10.6028/NIST.IR.8286D

Software Riesgos de Seguridad de la Información. (s. f.). Recuperado 14 de octubre de 2023, de Software ISO website: https://www.isotools.us/software/riesgos-seguridad-informacion/

Sommerville, I. (2005). *Ingeniería del software*. Pearson educación.

Stine, K., Quinn, S., Witte, G., & Gardner, R. (2020). *Integrating Cybersecurity and Enterprise Risk Management (ERM)* (N.º NIST Internal or Interagency Report (NISTIR) 8286). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.IR.8286

Toledo Rodríguez, M. (2019). *Módulo para la gestión de copias de seguridad en Nova 360* (BachelorThesis, Universidad de las Ciencias Informáticas. Facultad 1.). Universidad de las Ciencias Informáticas. Facultad 1. Recuperado de https://repositorio.uci.cu/jspui/handle/123456789/10098

Toro, R. (2015, abril 13). ISO 27001: El impacto en los Sistemas de Gestión de Seguridad de la Información. Recuperado 24 de septiembre de 2023, de PMG SSI - ISO 27001 website: https://www.pmg-ssi.com/2015/04/iso-27001-el-impacto-en-los-sistemas-de-gestion-de-seguridad-de-la-informacion/

Torres Hallo, M. (2020). *MODELO DE GESTIÓN DE RIESGOS DE PROCESOS DE TECNOLOGÍA DE INFORMACIÓN BAJO LA NORMA ISO/IEC 27000 EN EMPRESAS AÉREAS DEL ECUADOR*.

Vidal-Silva, C. L., Sánchez-Ortiz, A., Serrano, J., Rubio, J. M., Vidal-Silva, C. L., Sánchez-Ortiz, A., … Rubio, J. M. (2021). Experiencia académica en desarrollo rápido de sistemas de información web con Python y Django. *Formación universitaria*, *14*(5), 85-94. https://doi.org/10.4067/S0718-50062021000500085

ANEXOS

Anexo 1: Prototipo de interfaz de usuarios

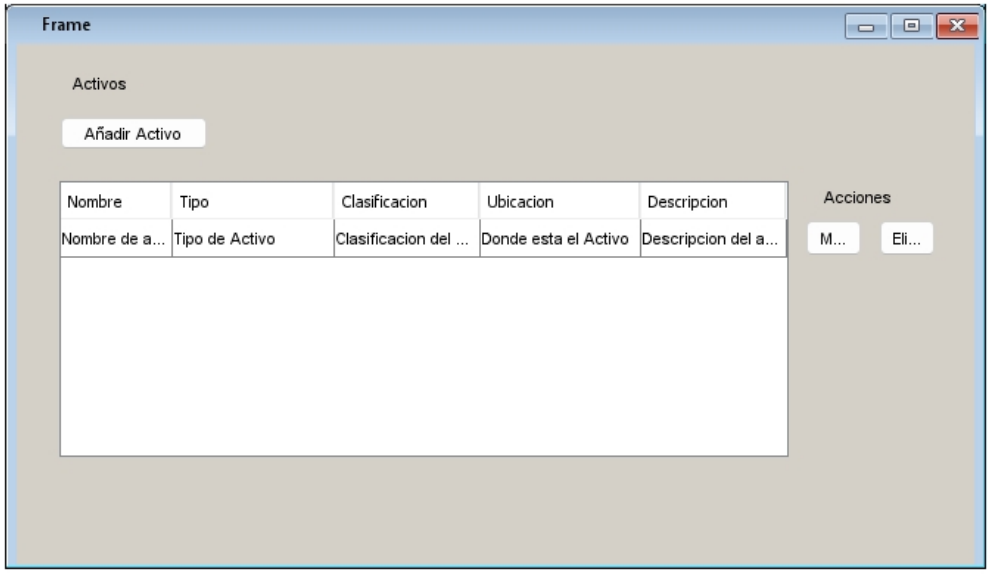


Figura 26: Prototipo mostrar activo

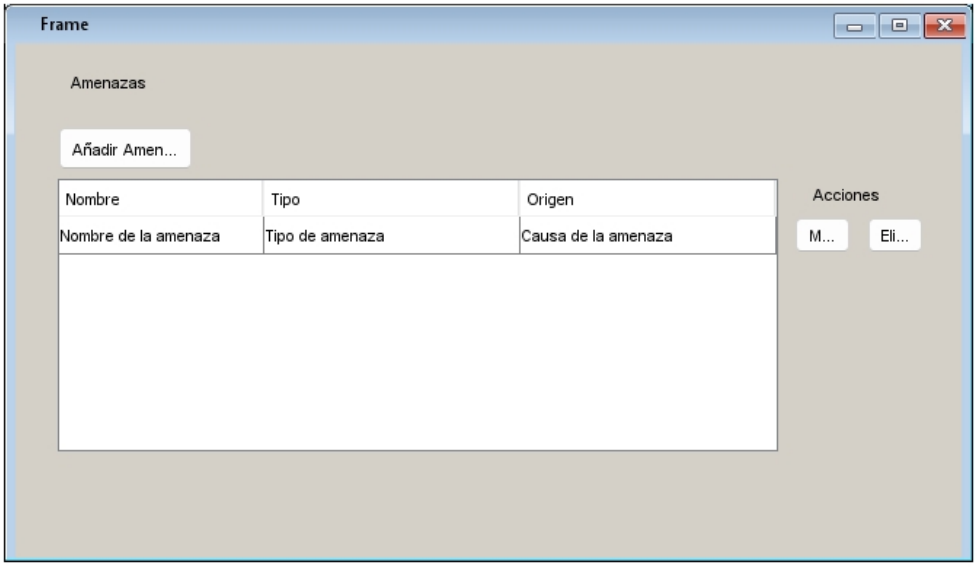


Figura 27: Prototipo mostrar amenazas

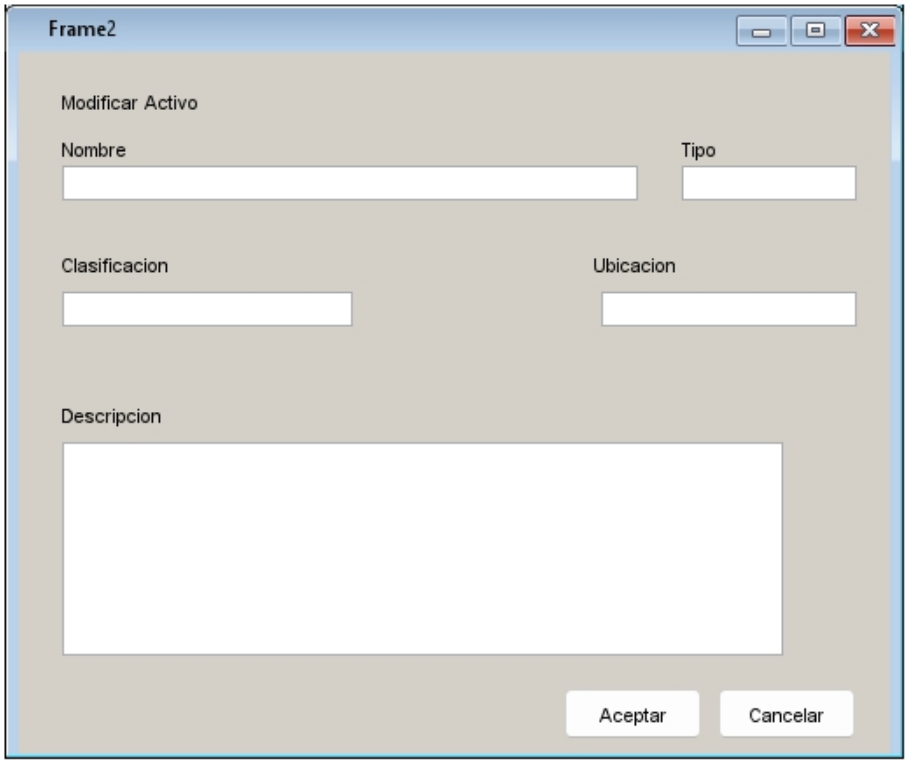


Figura 28: Prototipo modificar activo

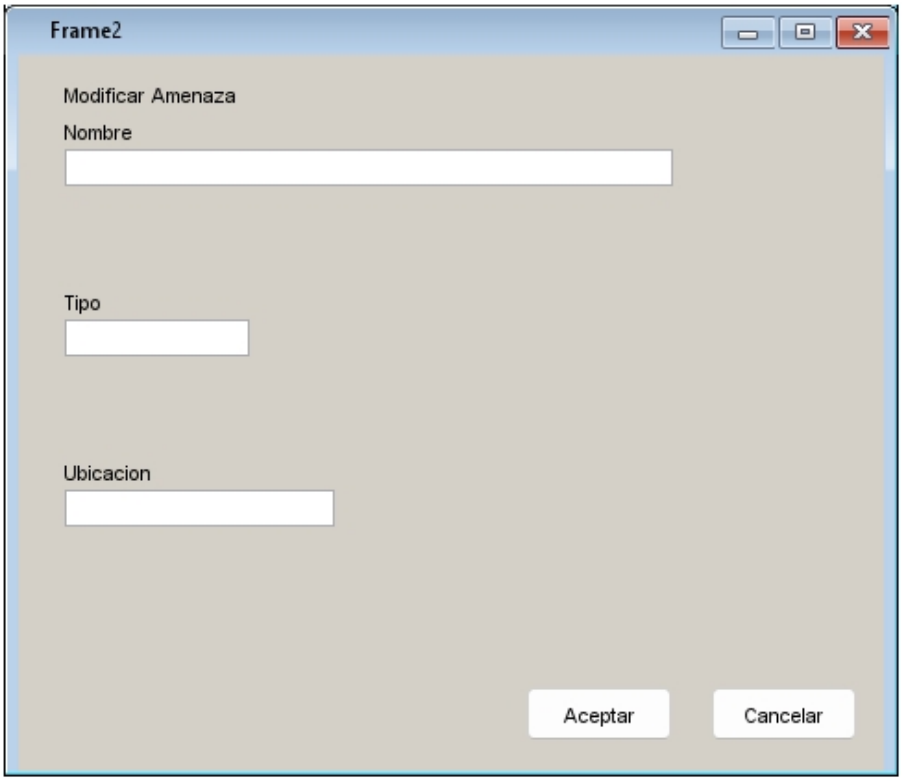


Figura 29: Prototipo modificar amenaza

Figura 30: Prototipo autenticar usuario

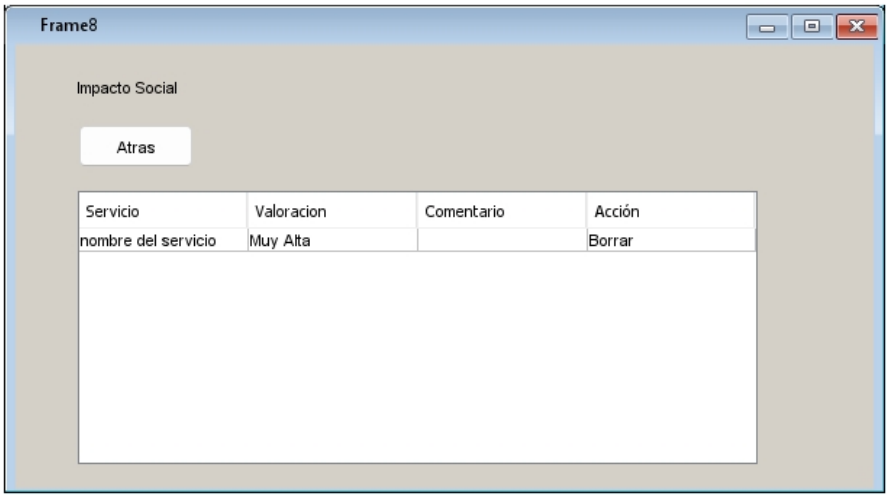


Figura 31: Prototipo impacto social

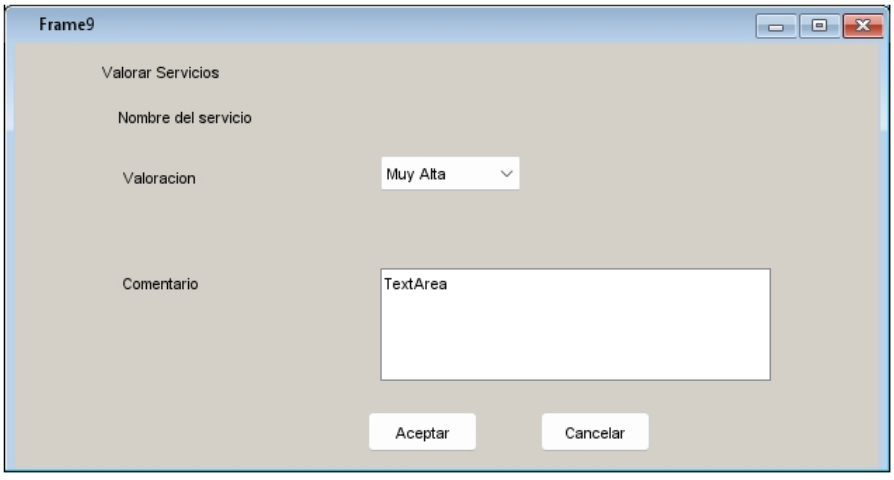


Figura 32: Prototipo valorar servicio

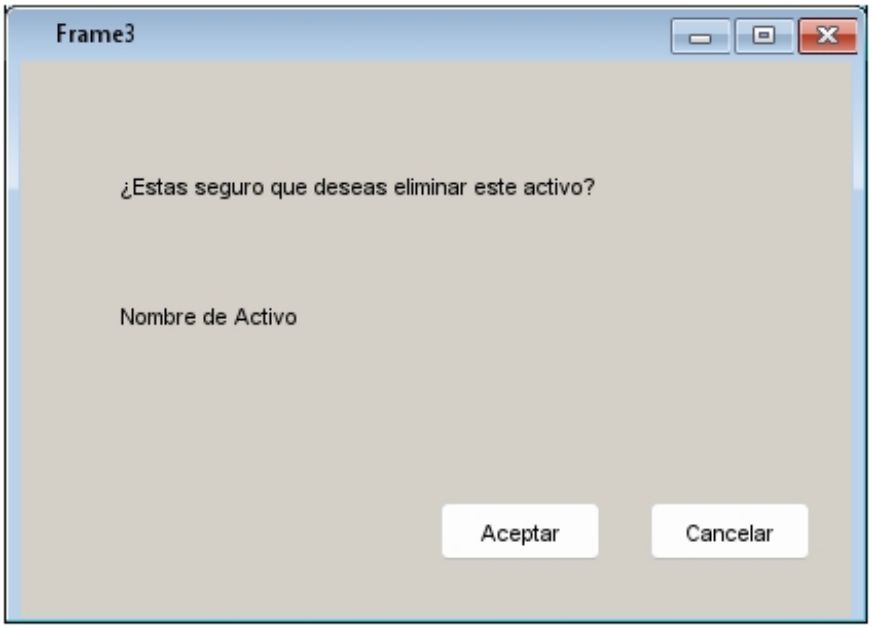


Figura 33: Prototipo eliminar activo

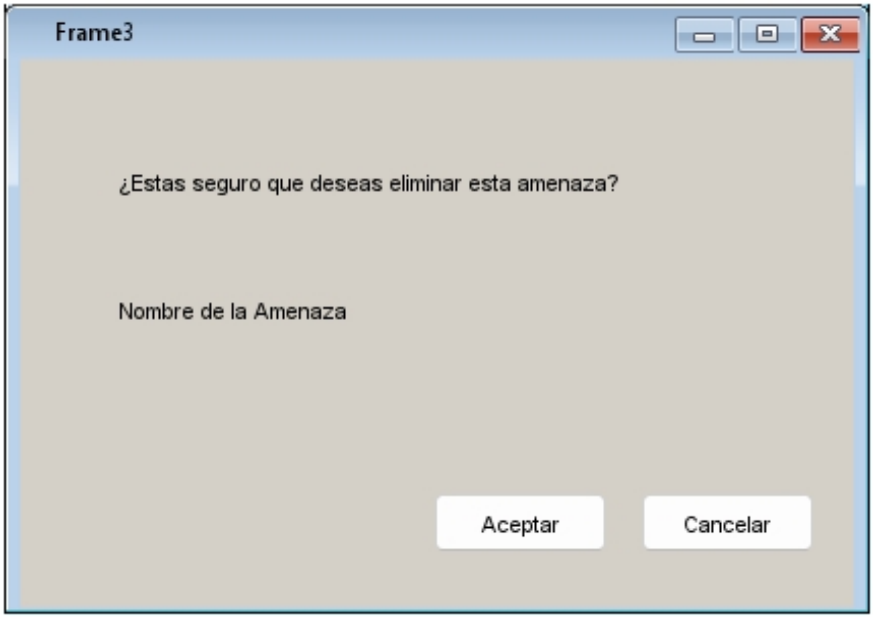


Figura 34: Prototipo eliminar amenaza

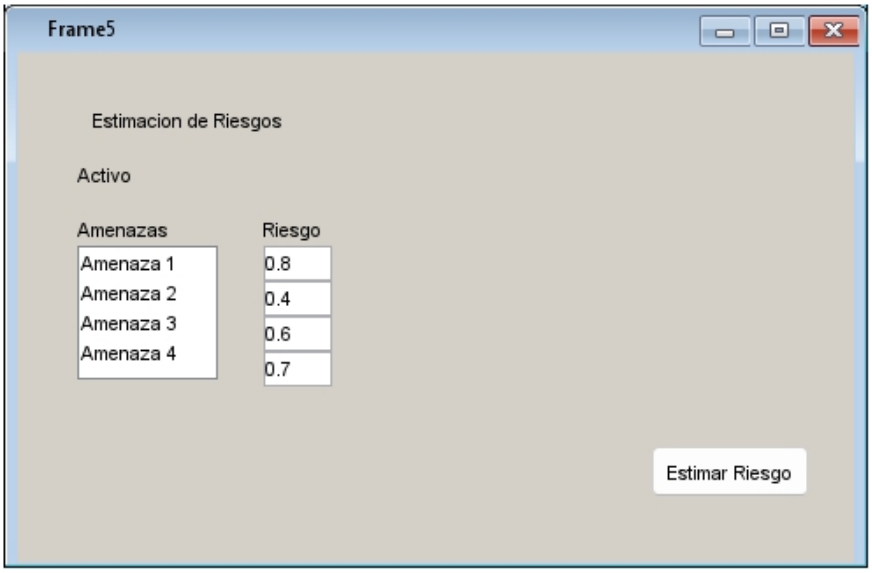


Figura 35: Prototipo estimación de riesgo

Anexo 2: Historias de usuario

Tabla 15: Historia de usuario Insertar amenaza

Historia de usuario	
Número: 7	**Nombre:** Insertar amenaza
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: El usuario tendrá la posibilidad de insertar en el sistema las amenazas que puedan ocurrir sobre los activos.	

Tabla 16: Autenticar usuario

Historia de usuario	
Número: 1	**Nombre:** Autenticar usuario
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 1 punto
Complejidad: Baja	**Puntos reales:** 1 punto
Descripción: El usuario podrá iniciar sesión para acceder en el sistema ingresando su usuario y contraseña	

Tabla 17: Modificar activos

Historia de usuario	
Número: 3	**Nombre:** Modificar activos
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: Una vez dentro de la página de Gestionar activos el usuario tendrá la posibilidad de insertar un nuevo activo ingresando su nombre, tipo y ubicación.	

Tabla 18: Mostrar activos

Historia de usuario	
Número: 4	**Nombre:** Mostrar activos
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 1 punto
Complejidad: Media	**Puntos reales:** 1 punto
Descripción: Una vez dentro de la página de Gestionar activos el usuario tendrá la posibilidad de visualizar los activos previamente ingresados.	

Tabla 19: Eliminar activos

Historia de usuario	
Número: 5	**Nombre:** Eliminar activos
Iteración asignada: 1	
Prioridad en negocio: Alta	**Puntos estimados:** 1 punto
Complejidad: Media	**Puntos reales:** 1 punto
Descripción: Una vez dentro de la página de Gestionar activos el usuario tendrá la posibilidad de eliminar un activo previamente ingresado.	

Tabla 20: Modificar amenaza

Historia de usuario	
Número: 8	**Nombre:** Modificar amenaza
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: Una vez dentro de la página de Gestionar amenazas el usuario tendrá la posibilidad de modificar los datos de la amenaza	

Tabla 21: Mostrar amenazas

Historia de usuario	
Número: 9	**Nombre:** Mostrar amenazas
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 1 punto
Complejidad: Media	**Puntos reales:** 1 punto
Descripción: Una vez dentro de la página de Gestionar amenazas el usuario tendrá la posibilidad de visualizar las amenazas previamente ingresadas.	

Tabla 22: Eliminar amenazas

Historia de usuario	
Número: 10	**Nombre:** Eliminar amenazas
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 1 punto
Complejidad: Media	**Puntos reales:** 1 punto
Descripción: Una vez dentro de la página de Gestionar activos el usuario tendrá la posibilidad de eliminar un activo previamente ingresado.	

Tabla 23: Estimar Riesgos

Historia de usuario	
Número: 11	**Nombre:** Estimar Riesgos
Iteración asignada: 3	
Prioridad en negocio: Media	**Puntos estimados:** 5 puntos
Complejidad: Alta	**Puntos reales:** 5 puntos
Descripción: El usuario podrá estimar el riesgo de que el activo se vea afectado por las amenazas	

Tabla 24: Medir impacto

Historia de usuario	
Número: 12	**Nombre:** Medir impacto
Iteración asignada: 2	
Prioridad en negocio: Alta	**Puntos estimados:** 3 puntos
Complejidad: Alta	**Puntos reales:** 3 puntos
Descripción: El usuario podrá visualizar el impacto para la entidad de que un activo de la misma sea vulnerado	

Tabla 25: Valorar Servicios

Historia de usuario	
Número: 13	**Nombre:** Valorar Servicios
Iteración asignada: 3	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: El usuario podrá acceder al sistema e insertar una valoración y un comentario acerca de los servicios con los que cuenta la entidad	

Tabla 26: Medir impacto social

Historia de usuario	
Número: 14	**Nombre:** Medir impacto social
Iteración asignada: 3	
Prioridad en negocio: Alta	**Puntos estimados:** 2 puntos
Complejidad: Media	**Puntos reales:** 2 puntos
Descripción: El usuario podrá acceder a las valoraciones y comentarios registrados sobre los servicios con los que cuenta la entidad para medir el impacto de la satisfacción de uso de los trabajadores	

Anexo 3: Resultados de las pruebas de rendimiento

▼ Test Iterations

Avg. time of successful iterations	190.529
Iterations started	3
Successful iterations	3

Figura 36: Iteraciones de la prueba.

▼ Requests

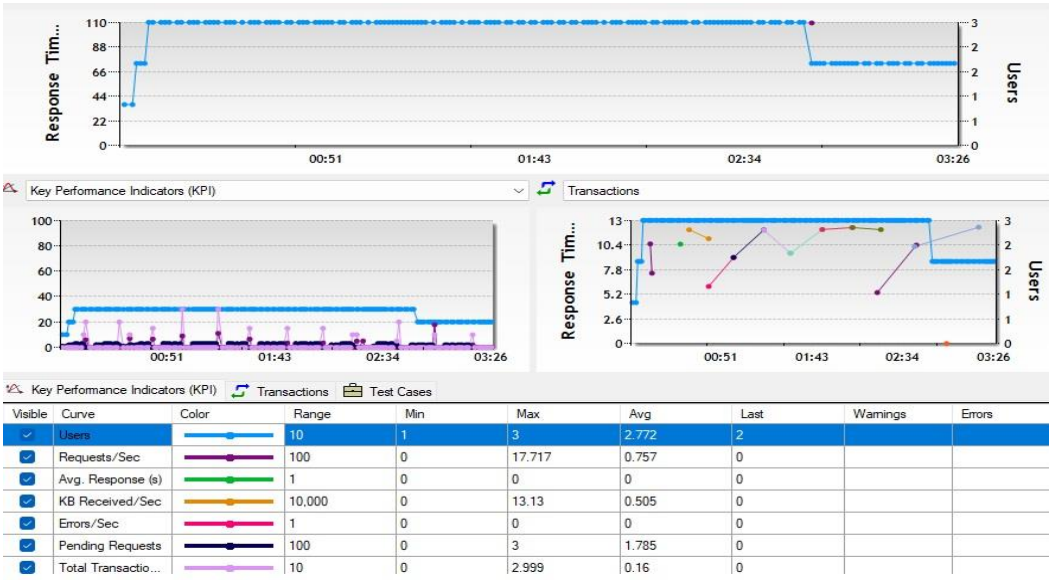
Avg. response time (s)	0
Requests/sec	0.743
Number of URLs	52
Total requests issued	156
Request errors	0
Request timeouts	0

Figura 37: Número de solicitudes.

Transaction details — Enter search text to filter by Transaction or Test Case

	Transaction	Requests	Iterations started	Successful Iterations	Avg.(s)	Min. (s)	Median (s)	Max. (s)	Goal (s)	Missed Goals	Errors	Timeouts
details	Calcular Importancia	1	3	3	8.726	5.392	10.393	10.393	15			
details	Editar Activos	4	3	3	11.028	9.096	11.987	12.003	15			
details	Editar Amenazas	6	3	3	12.095	12.022	12.022	12.24	15			
details	Eliminar Activos	6	3	3	10.349	9.522	9.537	11.987	15			
details	ImpactoSocial	3	3	3	11.599	10.238	12.28	12.28	15			
details	Insertar Activos	4	3	3	8.059	6.032	9.065	9.08	15			
details	Insertar Amenazas	2	3	3	12.172	12.036	12.24	12.24	15			
details	Insertar Valoracion	4	3	3	8.461	5.924	8.941	10.519	15			
details	Mostrar Activos	6	3	3	11.378	11.064	11.064	12.007	15			
details	Mostrar Amenazas	4	3	3	11.203	9.537	12.036	12.036	15			

Figura 38: Resultados de las pruebas de rendimiento I.



Visible	Curve	Color	Range	Min	Max	Avg	Last	Warnings	Errors
✓	Users		10	1	3	2.772	2		
✓	Requests/Sec		100	0	17.717	0.757	0		
✓	Avg. Response (s)		1	0	0	0	0		
✓	KB Received/Sec		10,000	0	13.13	0.505	0		
✓	Errors/Sec		1	0	0	0	0		
✓	Pending Requests		100	0	3	1.785	0		
✓	Total Transactio...		10	0	2.999	0.16	0		

Figura 39: Resultados de las pruebas de rendimiento II.