# SAFETY PLACE

CRISTHIAN GUISEPPE ZAMBRANO VIVAS
KEVYN ALEXANDER CHAVES CASTAÑEDA

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
BOGOTA DC
2022

# SAFETYPLACE

CRISTHIAN GUISEPPE ZAMBRANO VIVAS
Cod 430054216
KEVYN ALEXANDER CHAVES CASTAÑEDA
Cod 1720159

TRABAJO DE GRADO PARA OPTAR AL TÍTULO DE INGENIERO DE SISTEMAS

PROFESOR GILBERTO PEDRAZA GARCIA

UNIVERSIDAD PILOTO DE COLOMBIA
FACULTAD DE INGENIERÍA
PROGRAMA INGENIERÍA DE SISTEMAS
BOGOTA DC
2022

Nota de aceptación:

Firma del presidente del jurado

Firma del jurado

Firma del jurado

mBogota, 28 de Febrero de 2022

## Dedicatoria

El presente trabajo investigativo lo dedicamos principalmente a Dios, por ser el inspirador y darnos fuerza para continuar en este proceso de obtener nuestro título universitario.

A nuestros padres, por su amor, trabajo y sacrificio en todos estos años, gracias a ustedes hemos logrado llegar hasta aquí y convertirnos en lo que somos. Ha sido el orgullo y el privilegio de ser sus hijos, son los mejores padres.

A nuestras hermanas por estar siempre presentes, acompañándonos y por el apoyo moral, que nos brindaron a lo largo de esta etapa de nuestras vidas.

A todas las personas que nos han apoyado y han hecho que el trabajo se realice con éxito en especial a aquellos que nos abrieron las puertas y compartieron sus conocimientos.

**Agradecimientos**

A mis profesores y en especial a mis tutores por su ayuda, paciencia y dedicación.

También queremos agradecer a nuestras familias, que nos han apoyado durante todo este proceso.

A nuestros amigos de toda la vida.

# Contenido

Pág

## Lista de Figuras

Pág

## Lista de Tablas

Pág

## GLOSARIO

Amenaza informática: es una viable acción o acontecimiento negativo facilitado por una vulnerabilidad que tiene como consecuencia un efecto no esperado en un sistema o aplicación informática.

Contramedida: medida que se toma para paliar o anular otra; en especial, grupo de sistemas designado a neutralizar los dispositivos del oponente.

DataSet: es un grupo de datos, que comúnmente permanecen estructurados, como ejemplo podríamos mencionar que una tabla de una base de datos de SQL podría ser un dataset, en el cual cada columna de la tabla corresponde a una variable las filas representan los diferentes registros que almacena todas las columnas o cambiantes de la tabla.

Seguridad informática: es una disciplina que se ocupa de defender la totalidad y la privacidad de la información almacenada en un sistema informático.

Sistema operativo: grupo de directivas y programas que controlan los procesos básicos de un dispositivo y permiten el desempeño de otros programas.

Vulnerabilidad Informática: es una extenuación que existe en un sistema que podría ser usada por una persona malintencionada para comprometer su estabilidad.

## Abstract

Nowadays there is a growing problem which is the increase of cyber attacks presented worldwide. Software architects tend to apply the security attribute until the last phase of the software life cycle, this generates a cost overrun to the project causing delays in deliveries to customers. Many of these software architects tend to leave security to the last because searching for information regarding vulnerabilities is a complex process due to the fact that there are thousands of wikis dedicated to this, where the information is usually not clear and in most of the times this information is outdated and not complete. Our Safety Place project aims to develop a knowledge management strategy tool to support informed decision making in architecture design related to the application of countermeasures to block information security threats. The result of the development of this tool is a functional application that helps software architects in decision making by facilitating the implementation of the security attribute to the architect's projects, allowing him not only to consult information about vulnerabilities added by other users, but also to add vulnerabilities that he finds in the development of his application and to consult which of the countermeasures are the most used by the community regarding a specific vulnerability.



KEYWORDS: Software architecture, threats, computer security, software life cycle.



## Resumen

Hoy en día hay una problemática creciente la cuál es el aumento de ciber ataques presentados a nivel mundial. Los arquitectos de software suelen aplicar el atributo de seguridad hasta la última fase del ciclo de vida del software, esto genera un sobrecoste al proyecto causando retrasos en las entregas a los clientes. Muchos de estos arquitectos de software tienden a dejar la seguridad para lo último debido a que buscar información respecto a vulnerabilidades es un proceso bastante complejo debido a que hay miles de wikis dedicadas a esto, donde la información suele no ser clara y en la mayoría de las veces esta información está desactualizada y no está completa. Nuestro proyecto Safety Place pretende desarrollar una herramienta de estrategia de administración de conocimientos para apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de información. El resultado del desarrollo de esta herramienta es una aplicación funcional que ayuda a los arquitectos de software en la toma de decisiones

facilitando la implementación del atributo de seguridad a los proyectos del arquitecto, permitiéndole no solo consultar información sobre vulnerabilidades agregadas por otros usuarios, sino que a su vez puede agregar vulnerabilidades que encuentre en el desarrollo de su aplicación y consultar cuales de las contramedidas son las más usadas por la comunidad referente a una vulnerabilidad específica.

PALABRAS CLAVE: Arquitectura de software, amenazas, seguridad informática, ciclo de vida de un software.

# INTRODUCCIÓN

Como bien es sabido, en la actualidad las tecnologías de la información están jugando un gran papel debido a que estas tecnologías se utilizan en nuestro diario vivir, debido a esto, al papel tan importante que juega puede llegar a ser de suma relevancia, debido a lo anteriormente expuesto, es razonable la conclusión del tema de la seguridad informática debe ser uno de los temas en más auge, por lo que esta misma seguridad es uno factores claves en estas tecnologías.

Debido a que la seguridad es uno de los temas que más relevancia cobra, debido a los innumerables ataques informáticos que se han dado a nivel mundial, por esta razón ha surgido la necesidad de cómo se puede llegar a prevenir o defenderse de dichos ataques.

Este documento esta estructurador por seis capítulos, en donde el primero de ellos aborda la definición del problema en conjunto a los diferentes matices que la seguridad informática conlleva, también está acompañado por definición de los objetivos del proyecto SafetyPlace. El segundo capítulo se trata una ampliación del horizonte de varios aspectos de las tecnologías de la información, esto debido al marco referencial, en donde también se da una perspectiva de una gran parte del trabajo relacionado con el objetivo del proyecto. En el tercer capítulo se plantea una serie de tareas que resultan ser necesarias para llevar a cabo el proyecto SafetyPlace, dichas tareas son acompañadas por unos tiempos establecidos y con subtareas que conforman un conjunto. El cuarto capitulo, llamado desarrollo metodológico se abarca los conceptos que fueron necesarios de su comprensión para la adecuada culminación del proyecto. En el quinto capitulo se abordan temas importantes del proyecto como puede llegar a ser su diseño, objetivo e hipótesis y demás factores de suma relevancia para el proyecto. En el sexto y último capítulo se aborda los resultados obtenidos y el análisis de estos mismos que determinan si el proyecto arroja una hipótesis afirmativa o negativa.

# 1. GENERALIDADES DEL PROBLEMA

## 1.1 DEFINICIÓN DEL PROBLEMA.

Hoy en día la seguridad informática es de suma importancia debido a que la información se vuelve cada vez más valiosa, ya sea datos confidenciales como: Número de tarjeta de crédito o datos “públicos”.

La seguridad informática es uno de los temas más importantes no solo porque la información privada tiene valor, sino también porque cada día se va encontrando nuevas vulnerabilidades que ponen en peligro nuestra información.

Cuando hablamos de nuestra información, sobre la protección de nuestra información, una vulnerabilidad se puede dar en cualquier lugar y en cualquier momento, ninguno está exento de sufrir un ataque hacia la seguridad de nuestra información.

Teniendo en cuenta lo anterior cuando se habla de seguridad no solo esta corresponde a un ambiente cibernético sino que también el ambiente físico se ve envuelto cuando se habla de seguridad informática “La seguridad de la información digital para una organización depende de diferentes frentes: el físico, referente al alojamiento de la información; el social, relacionado con el grado de discrecionalidad del personal que la manipula, y el lógico, que se refiere a la configuración de sus niveles de accesibilidad y disposición”[1].

Las vulnerabilidades no solo afectan de manera “estándar” sino que hay varios niveles de severidad que cada uno de ellos compromete la información de diferente manera, dicho esto también es de suma importancia la gran variedad de “víctimas” que se ven afectadas por dichas vulnerabilidades que llegan a ser desde aplicaciones hasta pueden llegar a afectar el sistema operativo como tal.[2]

## 1.2 PREGUNTA PROBLEMA.

¿De qué manera puede contribuir el desarrollo de una herramienta de administración en la toma de decisiones en cuanto a seguridad informática se refiere?

---

[1] MONSALVE PULIDO, Julian; APONTE, Fredy Andres; CHAVES, David

[2] ESPAÑA, MINISTERIO DE INDUSTRIA, MINERÍA Y TURISMO

## 1.3 JUSTIFICACIÓN.

Hoy en día hay una innumerable cantidad de técnicas que usan los hackers para violentar la seguridad de una persona o una entidad. Esto se debe a que hoy en día la información tiene un valor increíblemente alto, y a medida que pasa el tiempo surgen nuevas vulnerabilidades que le permiten al hacker extraer y posteriormente vender la información obtenida. A continuación, mostraremos algunas de las técnicas más comunes para el robo de información.

- Phishing. Esta es una técnica por medio de la cual una persona suplanta ya sea a otra persona, a una entidad o una empresa, con el fin de obtener información crucial de la víctima. Comúnmente esta técnica se realiza mediante el denominado "Email Spoofing". El "Email Spoofing "consiste en suplantar por ejemplo a una empresa. Para fines explicativos tomaremos a Facebook como ejemplo. Supongamos que Pepito recibe un correo donde el destinatario es Facebook, y este le dice que ha encontrado actividad sospechosa en su cuenta y que necesita corroborar o verificar sus credenciales. Pepito abre el link y se encuentra una página a priori idéntica a la de Facebook, por lo cual le da confianza y procede a escribir sus credenciales, cuando las envía, el hacker obtiene acceso completo a la cuenta de Pepito, permitiéndole recopilar información valiosa que después podrá vender o disponer a su antojo. Este tipo de fishing anteriormente descrito, tiene como base la "vulnerabilidad" existente en el protocolo SMTP, en cual necesitas de dos cosas, el "MAIL FROM" y el "RCPT TO" para poder montar una app de phishing. El problema radica en que el protocolo SMTP no verifica si el correo de origen (mail Fromm), pertenece realmente a la entidad, y es por esto que esta modalidad es tan utilizada. Básicamente esta es una técnica usada desde el 2005 para robar información[3], la cual ha evolucionado en los últimos años, para suplantar incluso a entidades bancarias o gubernamentales. Esta técnica tiene su núcleo en el uso del correo electrónico.

[3] Digital Evidence and Electronic Signature Law Review, 6 (2009), pp. 153-157.

- Pharming. Esta es una técnica que se suele utilizar junto con el ya antes mencionado phishing, dónde te llega un correo, el cual es lo suficientemente atrayente para que hagas click en el enlace adjunto, y de esta manera robar tus datos. Sin embargo, esta técnica a perspectiva nuestra es un poco más profunda, debido a que su núcleo se basa en suplantar páginas web reales y *redireccionar* a una que suplanta por completo esta página. El pharming no solo se da a través de correo electrónico, también se puede dar a través de la descarga de una imagen, de una canción o un video, de “softwares libres “, entre otros. La modalidad de funcionamiento de esta técnica consiste en utilizar un virus, el cual tiene como misión infectar el ordenador en el cual se ha descargado. Para entender la función de dicho virus, primero debemos hablar de cómo nosotros como seres humanos nos comunicamos con un computador, para decirle que nos direccione por ejemplo a la página de Google. En primera instancia debemos saber que el ordenador solo entiende de números, en este caso direcciones IP, las cuales se alojan en una tabla de enrutamiento del sistema, indicando la dirección de un dominio específico. El encargado de en el buscador de nuestro navegador traducir la palabra “Google” por la dirección ip correspondiente es el DNS, de esta manera al escribir dicha palabra el navegador asocia la ip y nos redirecciona a dicho dominio. Teniendo entendido lo anteriormente descrito, el virus que ha infectado el ordenador lo que hace es cambiar la tabla de enrutamiento, y por ejemplo si la ip asociada a Google era 172.217.173.46 , la cambia por la ip del servidor u ordenador donde se encuentra la página web fake, provocando que cada vez que en su buscador ingrese la palabra Google , este lo redireccione a la página de pharming del hacker , de esta manera la persona que no tiene cuidado de los dominios donde ingresa es estafada.[4]

[4] Velasco Núñez, Eloy, Fraudes informáticos en red: del "phishing" al "pharming", en La Ley Penal, 37 (2007), pp. 57-60

- Ransomware. Anteriormente, todo software malicioso o malware, tenía como finalidad el infectar un dispositivo para dañarlo. Sin embargo, la tendencia de estos últimos años es usar estos malwares para la obtención de información crucial, comúnmente de empresas o entidades. El ransomware es todo software el cual se hace pasar por un programa ejecutable de confianza, para de esta manera encriptar la información alojada en el disco duro del computador, para posteriormente pedir un rescate de esta información. Esta ha sido una de las nuevas técnicas más usadas para la obtención de la información. Para situarnos un poco en esta técnica, debemos hablar de un fenómeno mundial el cual fue el ransomware denominado “WannaCry”. Este ransomware tenía como finalidad atacar equipos con el sistema operativo Windows, con la condición de que este estuviera *desactualizado*, para de esta manera entrar por una puerta trasera (exploit) y encriptar la información del disco duro de la máquina. Si tu equipo era infectado con este ransomware, se te desplegará la siguiente ventana:

Figura 1. Aviso de Ransomware


Fuente: tomado de https://www.elsoldeirapuato.com.mx/finanzas/mexico-entre-los-paises-mas-afectados-por-malware-wanna-cry-1675953.html

en esta ventana podías observar la explicación de qué había ocurrido con la información de tu computador, un tiempo límite en el que debías realizar el pago estipulado a cambio de desencriptar tu información, y además la dirección de bitcoin a la cual debías pagar. Este fue un malware que afectó

a empresas como FedEx, Telefónica, Deutsche Bank, LATAM, entre otras. Era increíble ver como en los aeropuertos, en las pantallas donde puedes ver los vuelos programados, salía esta misma ventana. Y es que WannaCry no solo priva de información sino también de servicios, causando que el dispositivo quede inservible. Para comprender el origen de este ransomware, debemos hablar de dos cosas, la primera es del grupo de hackers del cual surge este ransomware. Se trata del grupo denominado “LAZARUS GROUP”, el cual es uno de los grupos de hackers más reconocidos a nivel global. Como segunda instancia debemos hablar de la NSA (Agencia de seguridad nacional de Estados Unidos), la cual tiene como una de sus tareas el recopilar herramientas de hacking para todos los sistemas operativos, a esto se le denomina 0-day exploits. Un 0-day exploit es un hueco de seguridad informática que existe dentro de cualquier sistema operativo. La existencia de estas armas informáticas fue desvelada por el grupo de hackers denominado “ShadowBrokers” demostrando que la NSA busca fervientemente la manera de violentar la seguridad de las personas. LAZARUS encuentra un 0-day exploit para el sistema operativo de Windows y lo informa a la NSA y a Microsoft pidiendo que se les pagara una suma de dinero a cambio de no filtrar el fallo de seguridad, sin embargo, estas entidades no quisieron realizar el pago y el grupo de hackers procedió a publicar este 0-day exploit, teniendo como consecuencia de que en cuestión de solo unas semanas se diera origen al ransomware WannaCry.

Como podemos ver, son muchas las técnicas utilizadas para el robo de la información, y es que vivimos en una sociedad donde toda nuestra información crucial está digitalizada y en caso de no estar apercibidos de todas estas técnicas podemos llegar a ser víctimas de una estafa. Es por esto que como respuesta surge el proyecto Safety Place, buscando que el desarrollador cuente con una herramienta, la cual pueda detectar fallos de seguridad en la arquitectura de software utilizada, de esta manera los desarrolladores se verán beneficiados al contar con una herramienta que les permitirá detectar fallos de seguridad en el software que se esté utilizando, y así, poder brindar un software de mayor calidad y confiabilidad.

## 1.4 OBJETIVOS.

❖ **Objetivo general.**

- Desarrollar una herramienta de estrategia de la administración de conocimiento para apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de la información.

❖ **Objetivos específicos.**

- Reconocer la dinámica de funcionamiento de la arquitectura de software Cliente-Servidor.
- Identificar las mayores amenazas para cada una de las capas de la arquitectura de software Cliente-Servidor.
- Definir la contramedida adecuada para cada una de las potenciales amenazas que se puedan presentar en cada capa de la arquitectura Cliente-Servidor.
- Desarrollo de un prototipo funcional de una herramienta de administración de conocimiento que a partir de la definición de escenarios problemáticos en seguridad responda con alternativas de decisión para solucionar preocupaciones de diseño de arquitectura.

## 1.5 LÍMITES Y ALCANCES.

❖ Alcance.

- La presente herramienta, plantea una ayuda en la toma de decisiones para un arquitecto de software en cuanto temas de seguridad.
- Una amplia recopilación de datos, para la definición de vulnerabilidades y contramedidas informáticas.

❖ Límite.

- La decisión que puede llegar a tomar el arquitecto de software está fuera de nuestro control.
- El cómo abarque el arquitecto de software las vulnerabilidades que pueda presentar su diseño.

# 2. MARCO REFERENCIAL

## 2.1 MARCO CONCEPTUAL.

**2.1.1 ¿Qué es un sistema operativo?.** Un sistema operativo es un programa encargado de controlar la ejecución de distintas aplicaciones, ayudando considerablemente a facilitar el manejo de un dispositivo, como computadores y smartphones. También sirve como interfaz entre el hardware y la persona. Uno de los puntos más relevantes de un sistema operativo es incrementar en gran cantidad la eficiencia con la que se pueden usar los recursos del dispositivo. [5]

**2.1.2 Sistemas operativos para computadores.** En primera instancia tenemos a Microsoft Windows, Este es un sistema operativo muy popular, mayormente utilizados para computadores de hogar. Este sistema operativo cuenta con su última actualización denominada windows 10, el cual tiene versiones home y pro en la gama de hogar. En segunda instancia tenamos a MacOS X. Sistema operativo desarrollado por la empresa Apple para sus computadores. Actualmente la versión más reciente de este sistema operativo es MacOs Catalina. Por último tenemos Linux Ubuntu. Este es un sistema operativo open source, lo cual permite realizar los cambios que el usuario desee ya que es de código abierto, sin embargo, no es muy utilizado en los computadores para el hogar. Comúnmente este SO se utiliza para servidores de empresas dado a su facilidad operativa. Las versiones más "populares" de este sistema operativo son Ubuntu, Red Hat, Debian, Linux Mint, entre otros.

**2.1.3 Sistemas operativos para dispositivos móviles.** Actualmente existen dos sistemas operativos para dispositivos móviles, los cuales son :

- Android. Es el sistema operativo por excelencia para dispositivos móviles, casi todos los teléfonos en el mercado actual usan un sistema operativo android con una capa de personalización propia de cada fabricante. Actualmente la última versión de android es la 11.0.
- IOS. Este es el sistema operativo desarrollado por Apple, el cual es exclusivo para los dispositivos de la propia empresa. La última versión de este sistema operativo es la 14.0.1

**2.1.4 ¿Qué es seguridad informática?.** Según el diccionario de la Real Academia Española, el término seguridad hace referencia a "estar libre y exento de todo peligro, daño o riesgo"[6]. Por lo anterior, hablar de seguridad informática

[5] Stallings, William.Sistemas operativos Aspectos internos y principios de diseño,2005. p.52.

[6] Tomado de Diccionario de la Real Academia Española

hace referencia a un sistema de información en el cual se ha implementado una serie de medidas que buscan dar como resultado un sistema seguro y confiable. Aunque cabe mencionar que todo sistema de información siempre cuenta con un margen de error, y lo que se busca es minimizarlo lo más posible.

**2.1.5 ¿Cuál es el ciclo de vida de un software?.** El SDLC por sus siglas en inglés (Systems Development Life Cycle ) divide el ciclo de vida de un software en 5 etapas, estas son:

- Análisis. En esta etapa se define cuál será el tipo de aplicación o arquitectura que se llevará a cabo, en qué plataformas va a funcionar, qué tipos de datos se van a almacenar o transferir, acceso a los recursos, es decir a quien se le dan privilegios, que perfiles de usuario se van a definir y qué permisos para cada uno y cuáles van a ser los modos de entrar al software (contraseña por pin , biometría , etc)
- Diseño. En esta fase ya se comienza a idear una solución de cómo se va a llevar a cabo el desarrollo de software. Normalmente en esta fase se presentan los diagramas y pseudocódigos.
- Codificación: en esta fase es donde se decide sobre cuál lenguaje de programación se va a desarrollar el proyecto, se empieza a desarrollar el software y por regla general se establecen unas entregas de funcionalidad muy específica.
- Testing. En esta fase como su nombre lo indica, se realizan pruebas al software desarrollado, con la finalidad de verificar que su funcionalidad sea del 100%. Si resulta que su funcionalidad se ve comprometida de alguna manera, entonces se desarrollan los programas necesarios para solventar estos errores.
- Deployment. Esta es la fase donde se implementa el software desarrollado. Lo que se busca es evaluar factores como su portabilidad, adaptabilidad e integración.

**2.1.6 Tipos de seguridad.** Según afirma Aguilera[7] hay dos tipos de seguridad:

- Activa. Comprende el conjunto de defensas o medidas cuyo objetivo es evitar o reducir los riesgos que amenazan el sistema
- Pasiva. Está formada por las medidas que se implementan para, una vez producido el incidente de seguridad, minimizar su repercusión y facilitar la recuperación del sistema.

**2.1.7 Seguridad informática en sistemas operativos móviles.** Android: las amenazas informáticas que son comúnmente conocidas como malware, es una de

[7] Purificacion,Aguilera López. Seguridad Informática. EDITEX,2010

las principales fuentes de robo de datos, debido a esto la cantidad de detecciones de códigos maliciosos que se han diseñado para android es una muestra del avance tecnológico. También un gran avance de la tecnología es que los malware y los botnets son cada vez más elaborados. Es importante resaltar lo siguiente[8]:

- 4500 malware nuevos son generados diariamente.
- El malware para android subió un 600% en 2014.
- 1.5 millones de incidentes de ciberseguridad relacionados con android.

De acuerdo con los datos recolectados por la interpol y Kaspersky Lab 1 de cada 5 dispositivos cuyo sistema operativo es Android es atacado por un malware[9]Los malware más populares son los troyanos SMS, en donde se valen de mecanismos para infectar los dispositivos y poder capturar la información de transacciones electrónicas, por lo cual los atacantes pueden hacer transferencias de dinero a cuentas anónimas sin que la persona se dé cuenta que ha sido atacado.

Cuando se habla sobre latinoamérica se concluye por medio de una encuesta realizada los principales países que han reportado el mayor número de ataques cibernéticos en dispositivos móviles fueron los siguientes países Brasil, México y Colombia**.**

- **Principales amenazas en Latinoamérica.**
  - RiskTool. Generalmente es una herramienta adicional de un programa conocida comúnmente como “Crack / Keygen”, cuando se ejecuta por el usuario genera actividades malintencionadas que después pueden ser aprovechadas por los cibercriminales.

  - Trojan-SMS. Actual de la siguiente forma; el usuario descarga el malware en su equipo inadvertidamente, luego el Troyano envía un SMS a un número Premium sin el consentimiento del usuario, el SMS es transportado a través de la red de la compañía de telefonía prestadora del servicio SMS, posteriormente el código malicioso bloquea los mensajes de confirmación para finalmente permitir que el cibercriminal genere ganancias ilegales mediante el descuido de la víctima.

[8] Milano,Pablo.Op.Cit, p.9

[9] Informe Kaspersky Lab e INTERPOL. 2014

- AdWare. Se ejecuta automáticamente y su objetivo es mostrar publicidad, también conocido como pop-up o ventana emergente, cuando la víctima realiza alguna actividad sobre la publicidad se ejecutan funciones maliciosas que anteriormente el cibercriminal ha configurado[10].

- IOS. Apple es uno de los titanes de la tecnología móvil que ha impuesto varios estándares de calidad en la industria de dispositivos móviles, pero esto no lo deja exento de ser atacado por malware[11].

❖ **Principales amenazas IOS.**

➔ Troyanos de IOS de vigilancia remota y acceso móvil (mRAT)- Remote Access Troyans:Estos ataques se basan en hacer uso de una característica que utilizan los usuarios de esta plataforma conocida como Jailbreak[12], en el cual el usuario puede modificar o tener acceso a características del equipo y aplicaciones que no se encuentran disponibles a través de la App oficial.

➔ Certificados de desarrollador o de Empresas falsos: Usan certificados de distribución para cargar, no de la forma convencional, una aplicación (infectada con malware)[13], lo cual la beneficia por no tener que pasar por el proceso de validación de la tienda Apple y poderse descargar directamente a dispositivo

➔ Perfiles maliciosos de IOS:Los atacantes pueden hacer uso de estos perfiles para eludir el modelo de seguridad de Apple y poner en peligro el dispositivo de una víctima que perfiles infectados[14].

➔ Wi-Fi Man in The Middle(MiTM) :Involucran el uso ilegal de una red para explotar transacciones, conversaciones y transferencias de datos sobre la marcha[15].

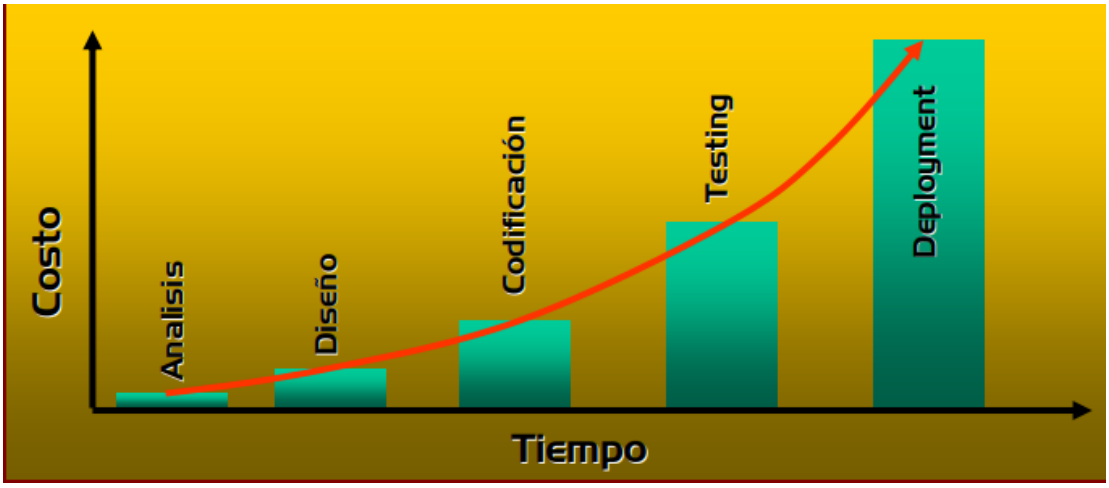
---

[10] Milano,Pablo.Op.Cit, p.9
[11] Milano,Pablo.Op.Cit, p.9
[12] Threats to iOS Mobile Devices. 2014
[13] Ibid.
[14] LA PORTAL.Malicious profiles – one of the most serious threats to IPhones . 2018
[15] Milano,Pablo.Op.Cit, p.9

## 2.2 MARCO TEÓRICO.

Hoy en día el proceso de desarrollo de software se lleva a cabo por medio del SDLC. El común denominador es que se trate la seguridad como un requerimiento de calidad. El verdadero problema radica en que, por regla general, este requerimiento se tiene en cuenta en la fase de testing del SDLC. Lo anterior significa un enorme incremento en el costo del proyecto como lo expresa Milano[16], y para entender un poco más de a que hacemos referencia observemos la siguiente imagen:

Figura 2 .Gráfica Costo vs Tiempo para implementación de seguridad




Fuente: Tomado de http://www.cybsec.com/upload/cybsec_Tendencias2007_Seguridad_SDLC.pdf

Como podemos observar, el hecho de implementar la seguridad incrementa el costo a medida que más tarde se implemente en el SDLC. Es por esto por lo que es de suma importancia implementar este requerimiento en cada una de las fases del ciclo de vida del software. Para asegurar que un software es seguro debe contar con las siguientes propiedades:

- Confidencialidad. Los datos o información debe estar únicamente al alcance de los actores autorizados, por medio de los permisos asignados para cada actor.[17]

[16] Milano, Pablo. Seguridad en el ciclo de vida del desarrollo de software.Buenos Aires: CYBSEC,2007

[17] Purificacion Aguilera López. Op.Cit, p.10

- Integridad. Evita una manipulación no autorizada de los datos. Esto permite garantizar la autenticidad y precisión de la información sin importar en qué momento esta se solicite.[18]
- Disponibilidad. Este principio asegura que el sistema se encuentre disponible para los actores que están previamente autorizados.
- Autenticación: por medio de esta propiedad se busca que antes de realizar cualquier procedimiento en el sistema, se verifique si el autor de verdad es quien dice ser.[19]
- No Repudio. Esta es la capacidad de que en el sistema se detecte la autoría de cualquier procedimiento realizado sobre el sistema.

Después de tener claro lo que significa que un software sea seguro , lo que nos compete es determinar cómo aplicar el requerimiento de seguridad en cada una de las fases del SDLC.

- Análisis. En esta etapa el desarrollador de software puede aplicar seguridad enfocada a cada uno de los requerimientos. Es decir, al tener en claro sobre qué plataforma correrá el software, ya se da una idea del kernel que este maneja, y, por lo tanto, está al tanto de los posibles backdoors que este posee, y que de alguna u otra manera desencadenará un ataque a su futura aplicación. De esta misma manera puede determinar qué tipo de datos va a almacenar el software, son privados o públicos. Tendrá que determinar qué tipo de actores van a existir dentro de la aplicación, y a su vez con que privilegios contará cada uno. Por último, deberá establecer el método de autenticación y cuántos factores de seguridad contendrá.
- Diseño. En esta fase se puede aplicar seguridad mediante la reducción de la superficie de ataque, adoptar el tan útil pero poco utilizado criterio del menor privilegio, es decir, asegurarnos que cada uno de los actores puedan realizar sólo los procedimientos estrictamente necesarios sobre el sistema.

Además, como menciona Milano[20] , en esta fase se pueden utilizar tres métodos que nos permiten implementar seguridad:

- Análisis de Riesgo-Threat Modeling. Por medio del modelado de amenazas se busca descomponer el software de tal manera que se identifiquen cuáles son los recursos clave, después se determina cuáles son las posibles amenazas a cada uno de estos recursos, se asigna un valor de impacto a cada una de las amenazas, se decide

[18] Ibid., p.11

[19] Pedraza García Gilbertro, Astudillo Hernan, Correal Dario. A Methodological Approach to Apply Security Tactics in Software Architecture Design

[20] Milano, Pablo.Op.Cit, p.9

cómo responder a estas amenazas identificando las técnicas y tecnologías que son necesarias para mitigarlas.

- Método STRIDE. Este método ayuda a identificar las amenazas en los componentes de un software, por medio de su acrónimo.
  - ➔ **S**poofing identity: hacerse pasar por otro actor dentro del sistema.
  - ➔ **T**ampering with data: se refiere a modificar la información dentro del sistema con la finalidad de causar daño.
  - ➔ **R**epudiation: Imposibilitar la identificación de el o los autores de algún procedimiento específico.
  - ➔ **I**nformation disclosure**:** se captura información para posteriormente divulgarla a actores no autorizados.
  - ➔ **D**enial of service: Imposibilitar el funcionamiento de un servicio
  - ➔ **E**levation of privilege: conseguir unos privilegios que no corresponden a los definidos para el actor.
- Método DREAD. Ayuda a asignar el valor de impacto a cada una de las amenazas por medio de su acrónimo:
  - ➔ **D**amage potential. Que tanto daño puede causar la amenaza
  - ➔ **R**eproducibility. Que tan factible es que se pueda reproducir la vulnerabilidad.
  - ➔ **E**xploitability. Que tan fácil de explotar es dicha vulnerabilidad
  - ➔ **A**ffected users. Que actores se verán afectados por dicha amenaza
  - ➔ **D**iscoverability. Que tan fácil de descubrir es la vulnerabilidad

- Codificación. En esta fase podemos hablar de aplicar las famosas buenas prácticas para una codificación segura. Esto nos habla de siempre validar los datos de entrada antes de realizar cualquier procedimiento con ellos, siempre desconfiar en que los datos que se reciban sean correctos,evitar generar código con alguna entrada realizada por el usuario,realizar la validación de los datos en cada una de las capas del sistema, etc.

- Testing. Para esta fase se han desarrollado varias técnicas para la implementación de seguridad:
  - Testing Funcional. Como su nombre lo dice, se encarga de testear que cada una de las funcionalidades estén operando correctamente, esto claramente incluye los request de autenticación, mecanismos de registro y login, etc.

- Testing de seguridad basado en riesgo. Se debe aplicar el método STRIDE a cada una de las interfaces
- Test de stress. En esta técnica lo que se busca es generar una carga bastante grande al software durante un tiempo prolongado, para de esta manera determinar si en un ataque de este estilo alguno de los servicios dejaría de estar disponible.
- Test de mutación de datos. Con este test se busca probar que tan seguro es el sistema al enviar datos mutados, es decir, al cambiar caracteres, cambiar el tipo, cambiar su longitud, etc.

- Deployment. Esta fase es tan importante como las anteriores, dado a que, si se da una mala implementación, se puede dañar todo lo que se había conseguido. Para esto hay que evitar poner servicios innecesarios, quitar las contraseñas de tipo default, además también es importante actualizar y liberar parches de seguridad cada cierto tiempo después de implementar el software.

## 2.3 TRABAJO RELACIONADO.

En la siguiente figura se puede apreciar como en el mercado actual existen aplicaciones que están enfocadas en la seguridad informática y cada una de estas herramientas está enfocada en un aspecto diferente que es de suma importancia para poder garantizar la seguridad informática en un equipo. A continuación, se expone una tabla con las herramientas más relevantes y para qué sirve cada una de ellas.

Figura 3. Productos en el mercado

| Nombre | Fecha de Lanzamiento | Peso | Breve Descripcion | Plataforma | Lenguaje | Costo |
|---|---|---|---|---|---|---|
| Nmap | 01-sep-97 | 84.2 MB | Es un programa de código abierto, que sirve para efectuar rastreo de puerto | Multiplataforma | C++, Python, C, Lua, Java | Free |
| Aircrack-ng | feb-06 | 27.1 MB | Es una suite de software de seguridad inalámbrica. | Multiplataforma | C | Free |
| Nikto | 17-dic-12 | 90 MB | Nikto proporciona la capacidad de escanear servidores web en busca de vulnerabilidades. | Unix-like | Perl | free |
| The Burp Suite | 01-07-04 | 908 MB | Entre las funciones básicas se encuentra el servidor proxy que permite inspeccionar y modificar el tráfico haciendo de intermediario entre el navegador y la aplicación destino | Multiplataforma | Java | $ 399.00 |
| Metasploit | 21-oct-09 | 703 MB | Proporciona información acerca de vulnerabilidades de seguridad y ayuda en tests de penetración "Pentesting" y el desarrollo de firmas para sistemas de detección de intrusos. | Multiplataforma | Perl, Ruby | Free |
| Uniscan | 13-mar-13 | 830 MB | Es una sencilla herramienta diseñada para ayudarnos a buscar vulnerabilidades en cualquier aplicación web muy fácilmente | Kali Linux | Perl | Free |
| Seccubus | 18-oct-15 | 814 MB | Realiza escaneos de vulnerabilidades a intervalos regulares, y compara lo que ha descubierto utilizando las diferentes herramientas, con los | Redhat, CentOS | Java | Free |
| Nessus | 15-sep-09 | 680 MB | Es un programa de escaneo de vulnerabilidades en diversos sistemas operativos. | Multiplataforma | Perl | Free |
| OWASP ZAP | 04-sep-10 | 857 MB | Es un escáner de seguridad web de código abierto. | Multiplataforma | Java | Free |
| OpenVAS | 02-may-15 | 750 MB | Es una suite de software, que ofrece un marco de trabajo para integrar servicios y herramientas especializadas en el escaneo y gestión de vulnerabilidades de seguridad de sistemas informáticos. | Multiplataforma | C | Free |
| Wireshark | 1998 | 890 MB | Es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para análisis de datos y protocolos, y como una herramienta didáctica. | Multiplataforma | C, C++ | Free |
| Hashcat | 07-nov-17 | | Es una herramienta para la recuperación de contraseñas | Multiplataforma | C | Free |
| | | | | | | |

Fuente: Elaboración propia

Como se puede apreciar cada herramienta está enfocada en aspectos diferente que es necesario para garantizar la seguridad informática, a diferencia de la herramienta propuesta Safety place cuyo objetivo principal es aconsejar adecuadamente a un arquitecto de software para que su arquitectura propuesta

sea una en la que no tenga, en la medida de lo posible, ninguna vulnerabilidad informática.

En este estado del arte se realiza con el propósito de dar un contexto más amplio de lo que está pasando en la actualidad con la arquitectura de software que es cliente-servidor, en donde a continuación se darán al a conocer la situación de esta arquitectura desde un nivel global a nivel local.

Cuando se trae a colación lo que es la arquitectura cliente servidor que en los últimos años ha avanzado a pasos de gigante desde sistemas centralizados (cliente-servidor no basados en web) a sistemas distribuidos y a sistemas de centralización virtual (Cloud Computing). Al momento de hablar de seguridad en la nube, se entiende que el proveedor tiene que brindar dicha seguridad, debido a los acuerdos de nivel de servicio (SLA), Kandukurii[21] menciona en su artículo que la computación en la nube se está utilizando como infraestructura de servicios, lo cual hace posible una exhaustiva examinación crítica de los problemas de seguridad y confidencialidad. No obstante, el garantizar la seguridad de dichos datos corporativos no llega a ser del todo fácil, esto sucede debido a los diferentes servicios como Software como servicio (SaaS), Plataforma como servicio (PaaS) e Infraestructura como servicio (IaaS), donde cada uno de estos servicios llega con sus propios problemas de seguridad.

A nivel mundial, con más exactitud en Malasia se es expuesto con gran precisión el estado que tiene la arquitectura cliente-servidor, que nos da a conocer que esta arquitectura va ganando una adopción por el sector empresarial, debido a que más empresas se trasladan a la nube, esto generando crecimiento en los mercados de servicios de computación, como bien menciona Shakirat[22], lo anterior expande el panorama del crecimiento y la importancia que ha tomado esta arquitectura con el paso de tiempo, no obstante no se puede dejar de lado la importancia que puede tomar la seguridad en esta arquitectura de software, en donde se han presentado varias propuestas para mejorar dicha seguridad como es la de Rawat[23] en donde se propone que por medio de redes que están definidas por un software (SDN) se proporciona seguridad y eficiencia energética, todo esto debido a la flexibilidad que esta permite, también es de suma

[21] Reddy Kandukuri, Balachandra, Paturi V, Ramakrishna, Rakshit, Atanu, Cloud Security Issues. (2009)
[22] Shakirat Oluwatosin, Haroon, Client-Server Model. (2014)
[23] Rawat, Danda B, Reddy, Swetha R, Software Defined Networking Architecture, Security and Energy Efficiency: A Survey. (2017)

importancia mencionar que debido a la rápida evolución que han tenido los dispositivos móviles, también surge la necesidad de

proteger la información del usuario, como bien lo menciona Kai Xi[24] en su artículo, en este mismo presenta un innovador protocolo de seguridad bio-criptográfico, el cual consiste en la verificación biométrica de las huellas dactilares, el cual se basa en un esquema de infraestructura de clave pública (PKI) computacionalmente eficiente, criptografía de curva elíptica (ECC).

En España, de la mano de Torres, se presenta la virtualización como una tecnología que brinda la posibilidad de abstraer una parte del software de una computadora para posteriormente ser desplegada de manera sumamente sencilla en otra donde pueden llegar a ser albergadas más máquinas virtuales. Este mecanismo permite alojar más de una computadora virtual en una computadora física, generando una intranet que está forjada por un conjunto de computadoras que ejercen la función de servidores. Se presenta la virtualización como una herramienta que permite tener una intranet mucho más eficiente con las otras, al reducir el número de computadoras en ella, que a su vez es más segura debido a que esta permite la realización de copias de seguridad de forma automatizada sin necesidad de un usuario presente y también facilitando la administración ya que es sistema centralizado de servicios en unos pocos servidores. Todo esto permitiendo ahorrar en recursos como lo son hardware, energía eléctrica y mantenimiento.

Dejando de lado los avances más relevantes que ha tenido la seguridad informática, es importante resaltar que este es un tema sumamente crucial, pero el enfoque en el que está pensado conduce de manera recurrente a vulnerabilidades informáticas, como bien resalta Mouratidis[25] en su artículo, es importante el cambiar el desarrollo de un sistema, que no deje el aspecto de la seguridad en un segundo plano si no que en cada paso esté presente.

A nivel Latinoamérica, en Argentina se desarrolló una investigación que se vio enfocada en la creación de una herramienta web, en la que sea posible un razonamiento automático que permita la integración del soporte gráfico en un ambiente modelado, como menciona Gimenez[26],teniendo presente lo anterior nos

[24]Xi, Kai, Ahmad, Tohari, Han, Fengling, Hu, Jiankun, A fingerprint-based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. (2010)
[25] Mouratidis, Haralambos, Giorgini, Paolo, Manson, Gordon, When security meets software engineering: a case of modelling secure information systems. (2005)
[26] Giménez, Christian, Braun, Germán, Cecchi, Laura, Fillottrani, Pablo Rubén, Una arquitectura cliente-servidor para modelado conceptual asistido por razonamiento automático. (2016)

da un contexto a lo que se puede llegar a lograr, o expresado de mejor manera la innovación a la que se puede llegar, a la vez en México se analizó la problemática actual que es la acumulación mundial de información, Voutssas[27] resalta en su artículo que la información en forma de documentos electrónicos o digitales, podrían implicar un riesgo , amenazas y vulnerabilidades que podrían llegar a afectar a dicha información, y a su vez se generan diversas estrategias de seguridad informática y la relación de ésta con la preservación y cuidado de esa información. También en México Roque[28], se planteó explorar las deficiencias que se pueden encontrar en la seguridad informática que se presentan en los estudiantes universitarios, para poder realizar una evaluación preliminar sobre el efecto que podría tener un programa que tenga como objetivo la capacitación y concientización diseñado para cubrir tales deficiencias. Se trabajó con un solo grupo de participantes, a quienes se encuestó antes y después de un evento formativo en modalidad de conferencia. Para analizar los datos se utilizó el software SPSS, donde se realizaron pruebas no paramétricas de Wilcoxon para buscar diferencias entre las respuestas recabadas antes y después del evento. Lo anterior arrojó como resultado que los participantes podrán obtener un nivel mucho mayor al que actualmente poseen en lo que se refiere a conocimientos y seguridad en actividades cotidianas.

Por otro lado, en Cuba, Díaz[29] en su artículo, en donde se realizó un trabajo en el que se propone un modelo para la gestión automatizada e integrada de controles de seguridad informática, basado en sistemas de gestión de información y eventos de seguridad (SIEM), que posibilita aumentar la efectividad de los controles implementados y disminuir la complejidad de la gestión de la seguridad de la información. Se define el concepto de automatización en el contexto de la seguridad informática y se determinan los controles que pueden ser automatizados. Como parte de la investigación se seleccionan un grupo de indicadores que permiten medir de forma automatizada la efectividad de los controles, se propone además una guía para la aplicación del modelo propuesto y se describe una posible implementación de este utilizando el sistema SIEM de software libre OSSIM. En paralelo en la ciudad de Quito, Jáuregui[30] realizo una tesis cuyo enfoque es la implementación de un servidor FTP Autenticado, que

[27] Voutssas M, Juan, Preservación documental digital y seguridad informática. (2010)
[28] Roque Hernández, Ramón Ventura, Juárez Ibarra, Carlos Manuel, Concientización y capacitación para incrementar la seguridad informática en estudiantes universitarios. (2018)
[29] Montesino Perurena, Raydel, Baluja García, Walter, Porvén Rubier, Joelsy, Gestión automatizada e integrada de controles de seguridad informática. (2013)
[30] Torres, Jáuregui, Pilar, Diana, Implementación de un servidor ftp autentificado con cliente-servidor filezilla. (2014)

sirve para el intercambio seguro y veloz de archivos para diversos clientes a través de un cliente servidor. Esto con el objetivo principal de acceder a dicha información sin la necesidad de estar en la empresa, contar con una contraseña aplicando métodos de seguridad informática, un análisis de datos que podrán transmitir los diversos elementos para que dicha implementación sea un completo éxito. Como fundamento principal se basará en un servidor FTP, que proporciona seguridad de la información, así como también realizar autenticación de los diversos clientes anteriormente mencionados, distintas pruebas de conexión, se encontraron una gran variedad de vulnerabilidades que se resolvieron, con una implementación de seguridades para fortalecer dicho servidor.

A nivel Colombia, en la ciudad de Medellín se propuso de la mano de Bonilla[31], la implementación de un modelo de seguridad en las organizaciones, en los cuales se deben seguir un paso a paso que son esenciales para una adecuada toma de decisiones que a su vez cumple la función de elevar el nivel de seguridad en la información, en este artículo se hace un recorrido por las diferentes y extensas configuraciones para el diseño de la seguridad, esto empieza por la identificación de amenazas informáticas, el cual es el primer paso que resulta esencial para una buena protección de los flujos de datos que se pueden llegar a manejar en diferentes organizaciones, el siguiente paso es el análisis e identificación de los riesgos, para que así poder mitigar el impacto que podría llegar a tener en caso que dicho riesgo se materialice, generar controles y políticas de seguridad las cuales pueden llegar a ser fácilmente cuantificadas y diseñadas de acuerdo a lo que se estableció en el paso anterior, para así poder llegar a tener un adecuado control sobre los impactos en la seguridad. Este modelo finaliza, donde se pueden identificar los requerimientos y se desarrolla una arquitectura de red en donde el tener una vulnerabilidad sea poco probable.

En la universidad UNAD (Universidad Nacional Abierta y a Distancia), se realizó un artículo de la mano de Solarte[32], en donde tienen como objetivo el desarrollar habilidades en los ingenieros de sistemas, que permitan conducir proyectos de diagnóstico, para la implementación e implantación de sistemas de seguridad de la información, todo esto teniendo como pilar los estándares ISO/IEC 27001 y el sistema de control propuesto en la norma ISO/IEC 27002.

En la Universidad Santo Tomás se realizó un estudio en el que se realiza un test de aprendizaje para determinar qué aspectos son más influyentes en el

[31] Bonilla, Sandra M, Gonzalez, Jaime A, Modelo de seguridad de la información. (2012)
[32] Solarte, Nicolas, Enriquez, Edgar, Benavides, Mirian, Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 2700. (2015)

aprendizaje, tales como es el aspecto audiovisual y medios de comunicación síncronos y asíncronos, donde la respuesta se da por medio de una herramienta que emplea la arquitectura cliente servidor, que daba respuestas gráficas, dando como resultado algo más comprensible para los estudiantes, como bien resalta Roa[33] en su artículo.

Teniendo presente lo anteriormente mencionado, se da un amplio panorama de cómo y para que se está utilizando la arquitectura cliente servidor a lo largo del mundo, no dejando de lado la seguridad que se implementa al momento de la creación de una arquitectura de este tipo. (como se maneja la seguridad física, más a nivel generalizado). En la herramienta propuesta Safety Place, se abarca la arquitectura de software Cliente-servidor, en donde buscan que el desarrollador cuente con una herramienta, la cual pueda detectar fallos de seguridad en la arquitectura de software utilizada, de esta manera los desarrolladores se verán beneficiados al contar con una herramienta que les permitirá detectar fallos de seguridad en el software que se esté utilizando, y así, poder brindar un software de mayor calidad y confiabilidad.

[33] Roa , Katherine, Martinez, Crisman, Cabrera, Carlos, Experiencias en el aula virtual como mediación pedagógica para el apoyo al aprendizaje en el espacio académico de lenguaje cliente servidor. (2019)

# 3. DISEÑO METODOLÓGICO

## 3.1 TIPO DE ESTUDIO.

La presente investigación se desarrolla utilizando el método de investigación cuantitativa. Se toma esta decisión debido a que utilizaremos recursos informáticos, recopilando información de artículos y libros de carácter científico, además de que para esta investigación hay variables claves como el tiempo, la calidad y la seguridad. Además, se adoptará el marco de trabajo SCRUM debido a que nos potencia enormemente el trabajo en equipo, además de tener entregas previamente planificadas, permitiéndonos tener una perspectiva amplia del panorama.

Tabla 1. Desarrollo de objetivos

| OBJETIVOS | DESARROLLO METODOLÓGICO |
|---|---|
| Comprender la dinámica de funcionamiento de la arquitectura de software Cliente-Servidor. | ● Reconocer la estructura de la arquitectura cliente-servidor que se encuentra divida por capas y identificar el funcionamiento individual y en conjunto de cada una de las capas. |
| Identificar las mayores amenazas para cada una de las capas de la arquitectura de software Cliente-Servidor. | ● Reconocer los recursos que pueden llegar a ser esenciales en cada una de las capas de la arquitectura de software y las vulnerabilidades que pueden llegar a tener. |
| Definir la contramedida adecuada para cada una de las potenciales amenazas que se puedan presentar en cada capa de la arquitectura Cliente-Servidor | ● Identificar las mayores amenazas que pueden llegar a afectar la arquitectura cliente-servidor.<br>● Comprender como cada potencial amenaza afecta a la arquitectura de software.<br>● |
| Desarrollo de un prototipo funcional de una herramienta de administración de conocimiento que a partir de la | ● Realizar el back-end.<br>● Realizar en Front-end.<br>● Realizar una seria de diagramas que contextualicen, la |

| | |
|---|---|
| definición de escenarios problemáticos en seguridad responda con alternativas de decisión para solucionar preocupaciones de diseño de arquitectura. | infraestructura, casos de uso, despliegue de la aplicación y componentes. |

*Fuente: Elaboración propia*

## 3.2 HIPÓTESIS.

El desarrollo de una herramienta de administración, cuyo enfoque principal es la ayuda en la toma de decisiones con respecto a temas de seguridad en una arquitectura cliente servidor, resultará benéfica para los desarrolladores de software.

## 3.3 PREGUNTA DE INVESTIGACION.

¿De qué manera puede contribuir el desarrollo de una herramienta de administración en la toma de decisiones en cuanto a seguridad informática se refiere?

## 3.4 METODOLOGÍA.

Para el desarrollo de la herramienta Safety Place, se lleva a cabo por la metodología de trabajo SCRUM, la cual consiste en: un marco de trabajo o framework que se usa en grupos que manejan proyectos complicados. O sea, hablamos de una metodología de trabajo adaptable que tiene como finalidad la entrega de costo en períodos cortos de tiempo. La metodología SCRUM se basa en aspectos que resultan ser fundamentales en la mayoría de las metodologías de trabajo, tales aspectos son:

- ➔ **Transparencia**. Con el Método Scrum todos los implicados poseen entendimiento de qué pasa en el plan y cómo pasa. Esto provoca que haya un conocimiento “común” del plan, una perspectiva universal
- ➔ **Inspección**. Los miembros del equipo Scrum muchas veces examinan el desarrollo para identificar probables inconvenientes. La inspección no es un examen diario, sino una forma de saber que el trabajo fluye y que los accesorios funcionan de forma autoorganizada.

- ➔ **Adaptación**. Una vez que hay algo que modificar, los equipamientos se acomodan para lograr la finalidad del sprint. Esta es la clave para lograr el triunfo en proyectos complicados, donde los requisitos son variables o poco definidos y en donde la habituación, la innovación, la dificultad y flexibilidad son primordiales.

**3.4.1 Tareas.** Para el efectivo desarrollo de la herramienta de software propuesta en este proyecto, se plantearon ciertos tipos de actividades, que contribuyeron a un completo desarrollo de la herramienta, dichas actividades se plantearon por objetivo como se podrá ver a continuación.

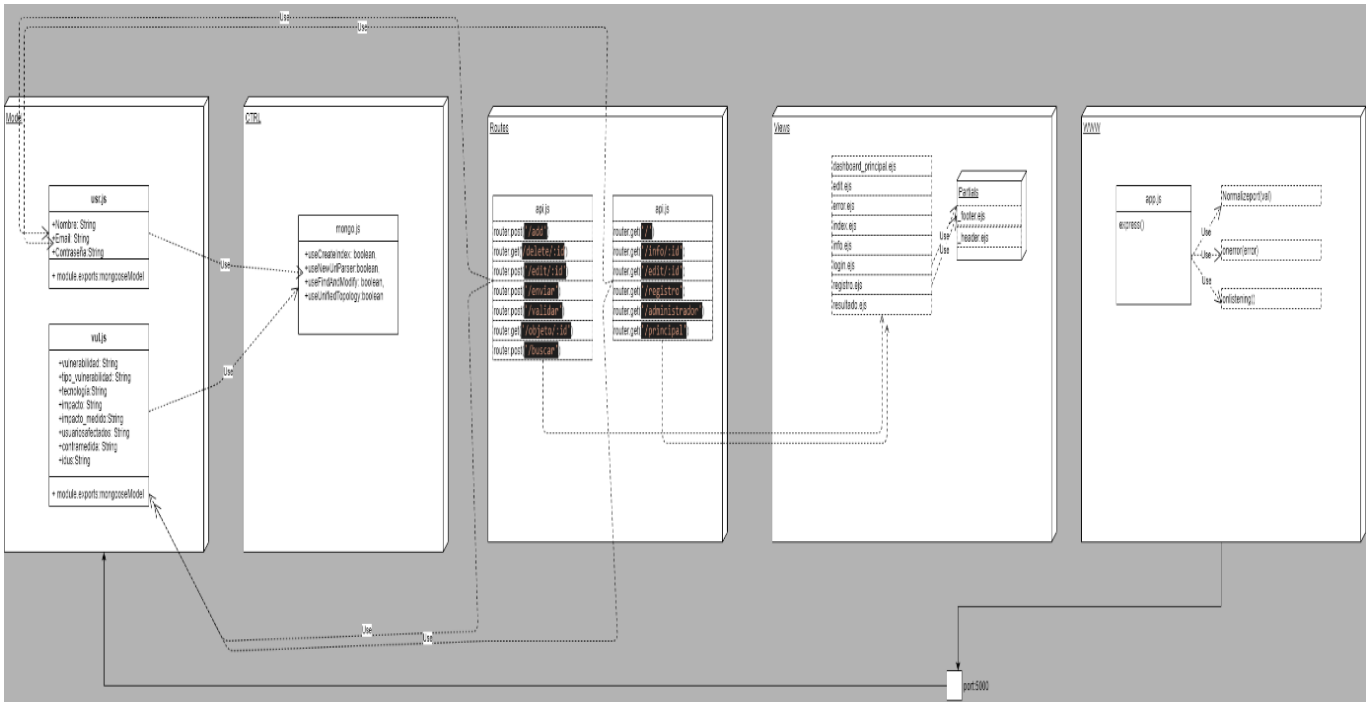
- ➔ Comprender la dinámica de funcionamiento de la arquitectura de software Cliente-Servidor.
  - ◆ Comprender el funcionamiento de la capa de presentación.
  - ◆ Comprender el funcionamiento de la capa de aplicación.
  - ◆ Comprender el funcionamiento de la comunicación entre los componentes del cliente y del servidor.
  - ◆ Comprender el funcionamiento de la capa de datos.
- ➔ Identificar las mayores amenazas para cada una de las capas de la arquitectura de software cliente-servidor.
  - ◆ Identificar cada uno de los recursos esenciales en cada una de las capas.
  - ◆ Identificar las vulnerabilidades más recientes que pueden llegar a afectar a cada uno de ellos.
  - ◆ Clasificar las amenazas según su impacto por el método DREAD.
  - ◆ Clasificar el tipo de amenaza por el método STRIDE.
- ➔ Definir la contramedida adecuada para cada una de las potenciales amenazas que se puedan presentar en cada capa de la arquitectura Cliente-Servidor.
  - ◆ Investigar los dataset gratuitos que contienen contramedidas adecuadas para las vulnerabilidades más conocidas.
  - ◆ Investigar sobre las posibles soluciones a ataques de SQL injection.
  - ◆ Investigar sobre las posibles soluciones a ataques de tipo “Elevation of privilege”.
  - ◆ Investigar cómo contrarrestar ataques de tipo “Denial of Service”

- ➔ Desarrollo de un prototipo funcional de una herramienta de administración de conocimiento que a partir de la definición de escenarios problemáticos en seguridad responda con alternativas de decisión para solucionar preocupaciones de diseño de arquitectura.
  - ◆ Desarrollar el back-end.
  - ◆ Desarrollar el front-end
  - ◆ Asociar datasets donde se encuentre información para las vulnerabilidades específicas para el recurso.
  - ◆ Desarrollar un sistema de recomendación para la toma de decisiones.

Las anteriores actividades fueron propuestas para su desarrollo en un lapso de catorce semanas, en la siguiente figura se puede apreciar cómo se planteó el desarrollo de dichas actividades en ese lapso, también se tiene planteado el realizar historias de usuario junto con un diagrama de clases, en el que se pueda evidenciar la estructura que tiene la aplicación Safety Place.

**3.4.2 Entregables.** Para poder llegar a tener una clara comprensión de la estructura de la aplicación Safety Place, y como esta desarrollada en el lenguaje de programación JavaScript, se tiene contemplado el desarrollo de un diagrama, que refleje la estructura, que tiene el aplicativo.

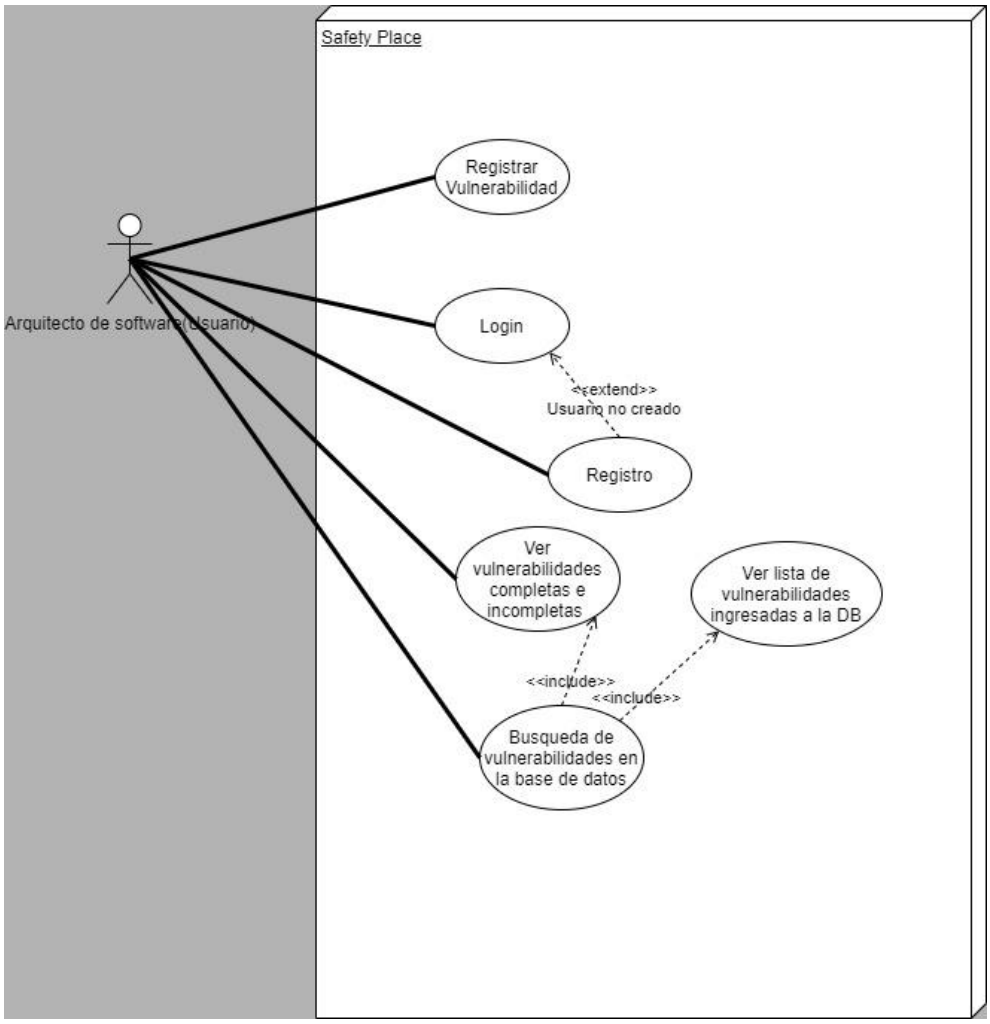
Figura 4 .Diagrama de SafetyPlace



*Fuente: Elaboración propia*

En el siguiente diagrama se puede apreciar de mejor manera, el cómo pueden estar relacionados los distintos casos de uso, anteriormente planteados, no solo eso, sino que también las acciones que se pueden llegar a realizar de dos diferentes agentes, que interactúan con el aplicativo Safety Place, uno de ellos es un usuario normal y el otro es el administrador de la aplicación.

Figura 5 .Diagrama de Casos de uso




Fuente: Elaboración propia

**3.4.3 Guía de validación de producto.** La herramienta de administración Safety Place, tiene como objetivo el ayudar a un arquitecto de software en la toma de decisiones en cuanto a temas de seguridad informática se trata, pero surge una pregunta, ¿Como se puede medir el impacto y la eficiencia de dicha herramienta? para poder dar respuesta a esta pregunta, se tienen a consideración tres factores de suma importancia, los cuales son: Facilidad de uso percibida por el usuario, Utilidad percibida por el usuario y intención de uso. se planteó tomar un grupo de muestra de un tamaño aproximado de diez integrantes, en donde se les plantee un ejercicio de planificar un diseño de arquitectura de software cliente-servidor, en donde a la primera mitad se le brindara la herramienta de administración y a la otra mitad no, para el diseño se les resalta que el tema de la seguridad debe estar presente, para así poder tener un diseño más completo, posteriormente se les dará una encuesta en donde se califique de uno a cinco los aspectos en los que pondría la dificultad del mismo desarrollo del diseño, los recursos utilizados en este diseño, la dificultad de la implementación en cuanto temas de seguridad, esto será de gran ayuda para poder definir el impacto y la efectividad que puede llegar a tener la herramienta de administración Safety Place.

# 4. DESARROLLO METODOLÓGICO

## 4.1 RECONOCER LA ARQUITECTURA CLIENTE SERVIDOR.

Para poder llegar a comprender la dinámica de funcionamiento de la arquitectura de software Cliente-Servidor, se tiene que plantear en un principio la pregunta ¿Que es una arquitectura de software? Según el estándar IEEE[34], La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Pero más allá de una simple definición es de suma importancia tener la clara función de una arquitectura de software Cliente- servidor, este modelo de arquitectura es un modelo de aplicación distribuida en las que las tareas son delegadas entre los diferentes proveedores de servicios que pueden llegar a existir, a estos se les es conocidos como servidores, y los que realizan las peticiones, son llamados clientes, las aplicaciones que son basadas en esta arquitectura de software pueden llegar a realizar una o más peticiones a los servidores, que deben responder a dichas peticiones, tanto el Cliente como el Servidor son entidades abstractas que pueden residir en la misma máquina o en máquinas diferentes, como bien resalta Marini[35] en su artículo.

La arquitectura de software cliente servidor está compuesta por 3 capas que son de suma importancia para el adecuado funcionamiento de la misma arquitectura, dichas capas son: capa de presentación, capa de aplicación y capa de datos.

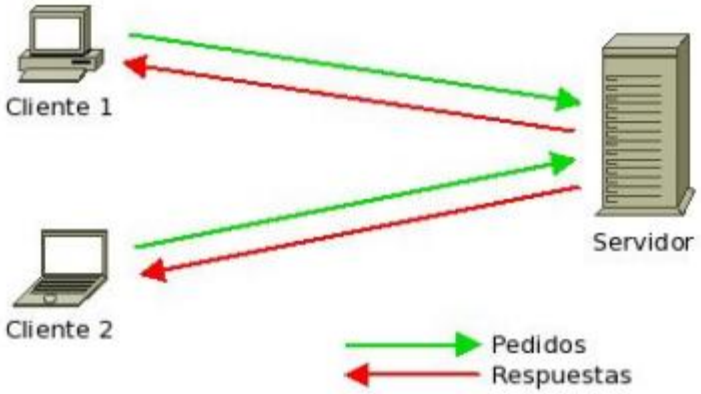
**4.1.1 Reconocer el funcionamiento capa de presentación.** Esta capa es la responsable con todo lo que tiene que ver con la interacción que puede llegar a ocurrir entre el usuario y una aplicación. Para poder realizar una tarea a cabalidad, es necesario conocer qué tipos de usuarios pueden llegar a utilizar la aplicación, las actividades que estos usuarios pueden llegar a realizar, teniendo esto presente se puede llegar a determinar cuál es el estilo óptimo para que los usuarios realicen dichas tareas. En esta capa se abarcan las tareas que se deben realizar por parte del cliente.

[34] Marini, Emiliano, El modelo cliente-servidor. (2012)

**4.1.2 Reconocer el funcionamiento capa de aplicación.** Esta capa es la encargada de gestionar y controlar la sucesión de acciones y la fuerza de cumplimiento de las reglas de negocio que rigen en las diferentes empresas, además también asegura la integridad de las transacciones de las operaciones que se llegaran a ser requeridas para que se cumplan dichas reglas. El objetivo que debe cumplir esta lógica es el de aislar las reglas del negocio, así como las transformaciones de datos de los consumidores (usuarios y otros componentes de esta misma capa) y de los sistemas de gestión de datos.

**4.1.3 Reconocer la comunicación existente entre cliente y servidor.** La comunicación que existe en la arquitectura cliente-servidor se da por medio de una red de comunicaciones en la cual los consumidores permanecen conectados a un servidor, en el cual se centralizan los múltiples recursos y aplicaciones con que se cuenta; y que los pone a disposición de los consumidores cada vez que dichos son solicitados. Esto quiere decir que cada una de las gestiones que se hacen se concentran en el servidor, de forma que en él se disponen los requerimientos provenientes de los consumidores que poseen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de solamente lectura y los que, por otro lado, tienen la posibilidad de ser modificados, etcétera. Esta clase de red puede utilizarse conjuntamente en caso de que se encuentre usando en una red mixta.

Figura 6. Figura arquitectura Cliente-Servidor




**Fuente: Tomado de https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf**

En la anterior figura se puede evidenciar a grandes rasgos cómo es que funciona la arquitectura cliente-servidor.

**4.1.4 Reconocer el funcionamiento capa de datos.** En esta capa se encuentran los procesos que están encargados de la gestión de los datos, es decir, los procesos que pueden llegar a ser requeridos para el mantenimiento de los datos. Estas tareas son realizadas, generalmente, por un Sistema de Gestión de Bases de Datos Relacionales, como SQL Server, Oracle, MySQL, Informix, etc.

**4.1.5 Conclusión.** Este objetivo tuvo como meta identificar y reconocer el funcionamiento de cada una de las capas de la arquitectura cliente servidor y como cada una de estas capaz interactúan. Con base en las múltiples tareas que fueron propuestas para poder llegar a este objetivo que dentro de ellas la más relevante fue el reconocer cada uno de estas capaz junto con su funcionamiento sin dejar esto de lado también se debe resaltar que el reconocer dicho funcionamiento es solo el primer paso para poder completar este objetivo, para poder llegar a la meta deseada se debe reconocer y comprender el cómo cada una de estas capaz interactúan entre sí, para poder actuar como un solo organismo. Como conclusión de este objetivo, fue posible el obtener una mayor comprensión de no solo el funcionamiento de una arquitectura cliente servidor, sino que también, se comprendió el cómo esta interactúa con sus componentes y recursos, lo cual, fue de suma relevancia al momento del desarrollo de la herramienta propuesta.

Para poder llegar a identificar las mayores amenazas para cada una de las capas de la arquitectura de software Cliente-Servidor, se debe tener claro que una arquitectura de software está divida en 3 capas, las cuales son capa de presentación, capa de lógica de negocio y por ultima la capa de datos.

## 4.2 LA IDENTIFICACIÓN DE RECURSOS ESENCIALES.

- Capa de presentación. Esta capa es la responsable con todo lo cual trata sobre la relación que puede llegar a suceder entre el cliente y una aplicación. Para lograr hacer una labor a cabalidad, se necesita conocer qué tipos de usuarios tienen la posibilidad de llegar a usar la aplicación, las ocupaciones que dichos usuarios tienen la posibilidad de llegar a hacer, teniendo esto presente se puede llegar a establecer cuál es el estilo óptimo para que los usuarios realicen dichas labores. Los recursos que pueden llegar a ser de suma importancia para esta capa son:
    - Java Server Faces (JSF).
    - ICE Faces.
    - RichFaces.
    - Apache Tomahawk.
    - Apache Trinidad.
    - Tobago.
    - AJAX.
    - DWR.
    - GWT.
- Capa de negocio. Esta capa es la delegada de gestionar y mantener el control de la sucesión de actividades y la fuerza de cumplimiento de las normas de negocio que rigen en las distintas organizaciones, además garantiza la totalidad de las transacciones de las operaciones que se llegaran a ser requeridas para que se cumplan dichas normas. La finalidad que debería consumar esta lógica es la de aislar las normas del comercio, así como las transformaciones de datos de los clientes (usuarios y otros elementos de esta misma capa) y de los sistemas de administración de datos.
    - Jboss.
    - Spring Framework.
    - Matrixssl.

- Capa de datos. En esta capa se hallan los procesos que permanecen delegados a la administración de los datos, o sea, los procesos que tienen la posibilidad de llegar a ser requeridos para el mantenimiento de los datos. Estas labores son llevadas a cabo, principalmente, por un Sistema de Administración de Bases de Datos Relacionales. Los recursos que pueden llegar a ser de suma importancia para esta capa son:
    - SQL Server.
    - Oracle.
    - MySQL.
    - Informix.

**4.2.1 La identificación de las amenazas en recursos esenciales.** Para poder llegar a identificar las vulnerabilidades existentes que se presentaron de manera más reciente en cada uno de los recursos, de mayor importancia en cada una de las capas que están presenten en la arquitectura cliente-servidor se utilizó como mayor herramienta el recopilatorio de información CVE , la cual tiene un recopilatoria de las vulnerabilidades más recientes que presentaron dichos recursos. Entre dichas vulnerabilidades que se encontraron las que cobran más relevancia en la arquitectura cliente-servidor son:

Tabla 2. Cuadro de amenazas

| Nombre de amenazas | Capa afectada |
|---|---|
| Denial Of Service | Capa de presentación, negocio y de datos |
| XSS | Capa de presentación |
| Directory Traversal | Capa de datos y de negocio |
| Code Execution | Capa de negocios |
| Sql Injection | Capa de datos |
| Elevation Of Privileges | Capa de negocio |

*Fuente: Elaboración propia*

**4.2.2 Clasificación de amenazas por el método dread.** La clasificación de amenazas se realiza por el método DREAD consiste en ayudar a asignar el valor de impacto a cada una de las amenazas por medio de su acrónimo, en este sistema de clasificación de amenazas se debe dar asignar cierto grado de prioridad dependiendo de los valores a evaluar en los siguientes aspectos:

- **D**amage potential. Que tanto daño puede causar la amenaza

- **R**eproducibility. Que tan factible es que se pueda reproducir la vulnerabilidad.
- **E**xploitability. Que tan fácil de explotar es dicha vulnerabilidad
- **A**ffected users. Que actores se verán afectados por dicha amenaza
- **D**iscoverability. Que tan fácil de descubrir es la vulnerabilidad

**4.2.3 Clasificación de amenazas por el método stride.** La clasificación de amenazas se realiza por el método STRIDE que consiste en ayudar a identificar las amenazas en los componentes de un software, por medio de su acrónimo.

- **S**poofing identity. Hacerse pasar por otro actor dentro del sistema.
- **T**ampering with data. Se refiere a modificar la información dentro del sistema con la finalidad de causar daño.
- **R**epudiation. Imposibilitar la identificación de el o los autores de algún procedimiento específico.
- **I**nformation disclosure**. S**e captura información para posteriormente divulgar a actores no autorizados.
- **D**enial of service. Imposibilitar el funcionamiento de un servicio
- **E**levation of privilege. Conseguir unos privilegios que no corresponden a los definidos para el actor.

**4.2.4 Conclusión.** Este objetivo tuvo como meta identificar no solo los recursos esenciales en cada una de las capas de la arquitectura cliente servidor, sino que también es el reconocer las amenazas que pueden llegar a afectar dichos recursos. Con base en las tareas que fueron planteadas para poder llegar a la meta propuesta de este objetivo, dentro de dichas tareas una de las que más cobro relevancia fue la de identificarlas y clasificar las diversas amenazas que se pueden llegar a presentar, todo esto debido a además de presentar ser una tarea muy laboriosa se debieron clasificar las amenazas identificadas se debieron clasificar por dos métodos diferente, los cuales son los métodos STRIDE y DREAD, donde el primero de ellos se enfoca en clasificar dependiendo tipo de amenaza que representa y el otro se enfoca en clasificar cinco aspectos que se pueden interpretar como el impacto que puede llagar a tener dicha amenaza. Como conclusión de este objetivo, gracias a las actividades propuestas que se plantearon con antelación para el desarrollo de este objetivo, fue posible que al identificar los recursos de cada una de las capas que conforman la arquitectura cliente -servidor, se hallaran las vulnerabilidades que pueden llegar a afectar dichos recursos y también se permitió que gracias a la identificación de las amenazas una adecuada clasificación de estas por los dos métodos plantados,

que permite determinar el grado de importancia que puede llegar a tener cada vulnerabilidad encontrada.

### 4.3 DEFINICIÓN DE CONTRAMEDIDAS.

Para poder llegar a definir la contramedida adecuada para cada una de las potenciales amenazas que se puedan presentar en cada capa de la arquitectura Cliente-Servidor, en una primera instancia se debe poder definir una amenaza y cómo afecta a la arquitectura de software.

**4.3.1 Consulta de datasets.** Para poder realizar una adecuada definición de contramedidas, para las vulnerabilidades más relevantes, se debió acudir a recopilatorios de información en los cuales se encontraban alojados una basta cantidad de información, dentro de las cuales se encontraban lo que se refería a la vulnerabilidad acompañada de su respectiva contramedida. dichos recopilatorios de información que se utilizaron para llevar a cabo esta tarea fueron CVE y IT Security Database.

**4.3.2 Consulta sql injection.** Una inyección SQL se da una vez que, de alguna forma, se inserta o "inyecta" código SQL invasor dentro del código SQL programado, con el objetivo de alterar el manejo usual del programa y conseguir de esta forma que se ejecute el fragmento de código "invasor" incrustado, en la base de datos. Esta clase de intrusión comúnmente es de carácter maligno, nocivo o espía, por consiguiente, es un inconveniente de estabilidad informática, y debería ser tomado presente por el programador de la aplicación para lograr prevenirlo. Un programa producido con descuido, displicencia o con ignorancia del problema, va a poder ser vulnerable, y la estabilidad del sistema (base de datos) va a poder permanecer ocasionalmente comprometida.

De las prácticas más comunes para poder evitar este tipo de vulnerabilidades están:

- Usar declaraciones preparadas con consultas parametrizadas
- Usar procedimientos almacenados
- Validación de entrada de la lista blanca
- Hacer cumplir el principio de menor privilegio

**4.3.3 Consulta elevation of privilege.** El incremento de privilegios es una manera común para que los atacantes obtengan ingreso no autorizado a los sistemas en un perímetro de estabilidad. Los atacantes empiezan por descubrir puntos de vista débiles en las defensas de una organización y obtener ingreso a un sistema. En varios casos, aquel primer punto de penetración no otorgará a los atacantes el grado de ingreso o datos que requieren. Después intentarán escalar los privilegios para obtener más roles u obtener ingresos a sistemas extras más propensos. En algunas ocasiones, los atacantes que tratan de escalar los privilegios encuentran que las «puertas permanecen abiertas de par en par»: controles de estabilidad inadecuados o incumplimiento del inicio de menor privilegio, y los usuarios poseen más privilegios de los que realmente requieren. En otras ocasiones, los atacantes explotan las vulnerabilidades del programa o usan técnicas concretas para superar el mecanismo de roles de un sistema operativo.

De las prácticas más comunes para poder evitar este tipo de vulnerabilidades están:

- Aplicar políticas de contraseña
- Crear usuarios y grupos especializados con privilegios mínimos necesarios y acceso a archivos
- Mantener los sistemas y aplicaciones parcheados y actualizados
- Cambiar las credenciales predeterminadas en todos los dispositivos, incluidos los enrutadores e impresoras

**4.3.4 Consulta denial of service.** La denegación de servicio hace lo cual su nombre indica, que un portal web no se encuentre disponible para los usuarios, empero uno famoso perjudica a toda una base de datos de usuarios online. En un ataque DoS, el agresor puede utilizar una exclusiva conexión a internet para explotar una vulnerabilidad de programa o inundar el propósito con demandas equivocadas y al final hacer que el lugar no se encuentre disponible y evadir que responda a las demandas de los usuarios legítimos. O bien, el ataque puede iniciarse a partir de diversos dispositivos conectados que se distribuyen por medio de Internet, en un ataque de Denegación de Servicio Compartido orquestado, o DoS.

De las prácticas más comunes para poder evitar este tipo de vulnerabilidades están:

- Garantizar suficiente ancho de banda
- Configuración segura y actualización de aplicaciones
- Hardware en las instalaciones

- Servicios basados en la nube

**4.3.5 Conclusión.** Este objetivo tuvo como meta el definir contramedidas adecuadas a las amenazas que se identificaron y clasificaron con anterioridad. Con base en las tareas que fueron planteadas para poder llegar a la meta propuesta de este objetivo, las diversas taras consistían en que partiendo de una amenaza en concreto se consultarían datasets en dónde se obtendrían las contramedidas adecuadas para dicha amenaza, en este objetivo se definieron contramedidas para tres amenazas en concreto que resultan ser las más comunes que afectan a la arquitectura cliente servidor. Como conclusión de este objetivo, gracias a las actividades propuestas, fue posible el determinar las contramedidas adecuadas para las distintas vulnerabilidades que se pueden llegar a encontrar en una arquitectura cliente-servidor, sino que también fue posible la indagación de contramedidas específicas para entender el cómo funcionan y en cómo podrían llegar a afectar un aplicativo con una arquitectura cliente-servidor.

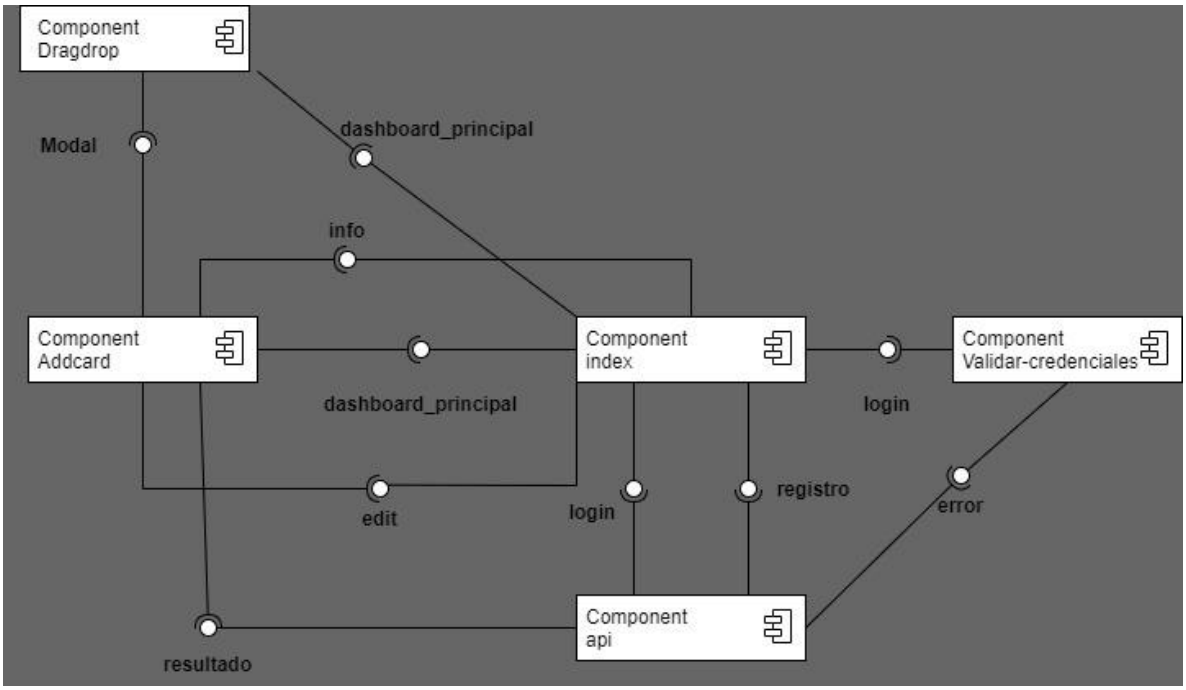
## 4.4 DESARROLLO DE LA HERRAMIENTA.

Para poder desarrollar una herramienta de administración de conocimiento que a partir de la definición de escenarios problémicos en seguridad, es de tener en cuenta la forma en la que fue construida dicha herramienta.

**4.4.1 Desarrollo del back-end.** Para el desarrollo del Back-end, de la herramienta de administración, se realizó, en el lenguaje de programación JavaScript (JS), gracias a Node JS, tecnología la cual permite ejecutar lenguaje JS desde el lado del servidor. A su vez por medio de Node se creó un servidor de express con el cual se conecta la parte front y back del proyecto, permitiéndonos manejar peticiones http y crear las rutas para las mismas. Esta herramienta se pudo realizar en gran parte a las APIs existentes, que permitió un ágil y fácil desarrollo del back-end de dicha herramienta, por otro lado, como base de datos se utilizó fue Atlas de MongoDB que nos permite tener una base de datos no relacional que siempre esté al alcance. Los componentes que constituyen la herramienta Safety Place, son los que se pueden apreciar en la siguiente figura, en donde se puede apreciar aspectos que resultaron fundamentales al desarrollo de la hermanita,

entre estos aspectos se puede encontrar las APIs utilizadas, las validaciones de las credenciales, que son necearías para el uso de la aplicación.

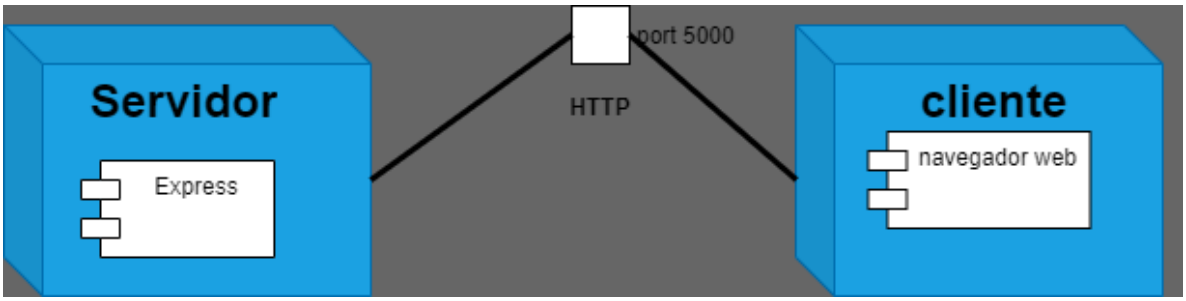
Figura 7. Diagrama de componentes




Fuente: Elaboración propia

También, es importante resaltar que la aplicación se encuentra alojada en un host de Heroku que permite la accesibilidad de la herramienta desde cualquier parte, sin ningún problema. Para esto se despliega la herramienta, desde este host que es un servidor Express que se enlaza desde el puerto 5000 y permite el acceso a la aplicación desde un navegador, esto se hace por medio del protocolo HTTP, esto se ilustra de mejor manera en la siguiente figura.

Figura 8. Despliegue




Fuente: Elaboración propia

**4.4.2 Desarrollo del front-end.** Para desarrollo del Front-end, de la herramienta de administración, se realizó por medio de dos recursos que resultaron de suma importancia para un adecuado desarrollo, estos recursos fueron CSS y EJS, los cuales van enfocados en la parte gráfica en la que interactúa el usuario. Además, se utilizó el framework llamado Bootstrap en su versión 5, facilitándonos la creación de elementos y contenedores.

**4.4.3 Asociación de datasets.** Para poder llegar a realizar una adecuada asociación entre los recopilatorios de información que fueron utilizados en el desarrollo y la herramienta de administración, se realizó con ayuda de la base de datos que fue utilizada en el Back-end, se creó una columna especial para poder llegar a realizar dicha asociación, entre la información que se puede llegar a encontrar.

**4.4.4 Desarrollo del back-end.** Para poder llegar a realizar un sistema de recomendación que funcione de manera óptima, se realizó por medio del lenguaje de programación JavaScript (JS), que un factor importante para dicha recomendación es el uso de varios tipos de filtros que permitan que el usuario llegue de manera fácil y ágil, a la información deseada

**4.4.5 Conclusión.** Este objetivo tuvo como meta el desarrollo de la herramienta Safety Place, esta herramienta se enfoca en ser una ayuda en la toma de decisiones en cuestiones de seguridad informática en arquitecturas cliente servidor. Con base en las tareas que fueron propuestas para poder llegar a la meta propuesta de este objetivo, las diversas actividades que fueron necesarias para la culminación de este objetivo fue necesario desde el desarrollo del back-

end en donde se utilizaron diversos recursos para poder realizarlo, este back-end consiste en armar toda la infraestructura necesaria para que el aplicativo funcione, en otro aspecto también se debe desarrollar en front-end, el cual consiste en el funcionamiento del aspecto grafico del aplicativo. Como conclusión de este objetivo gracias a las herramientas propuestas en este capítulo, fue posible el adecuado desarrollo de una herramienta de administración de conocimiento, que tiene un enfoque en la ayuda en la toma de decisiones en cuanto a temas de seguridad en arquitecturas cliente-servidor se trate.

# 5. VALIDACIÓN DEL PRODUCTO

## 5.1 DISEÑO.

La seguridad informática, con el pasar del tiempo ha tomado mucha más relevancia, debido a que cada vez la tecnología juega un papel fundamental, en no solo el diario vivir de las personas, sino que también es imposible el solo hecho de imaginar la era actual sin la tecnología.

Debido a que la tecnología desempeña un papel tan indispensable en el contexto de la actualidad, la seguridad que esta la acompaña a de ser primordial. Pero lo anterior lleva a la pregunta ¿Actualmente se goza de total protección? En cuanto a temas de seguridad informática se trata, lamentablemente la respuesta a la anterior pregunta es un no, debido a que por la misma razón que la tecnología juega un papel tan importante en el diario vivir, que se van presentando amenazas a la seguridad informática, que pueden llegar a afectar de distintas maneras, en distintos niveles de severidad que puede llegar a comprometer la información que tanto se desea resguardar.

**5.1.1 Objetivo.** Debido al atenuante panorama que se experimenta en la actualidad en cuanto a temas de seguridad informática se trata, se plantea como ayuda la herramienta de administración de conocimiento Safety Place, la cual tiene como objetivo, ser un apoyo en la toma de decisiones en cuanto a seguridad se trata para los arquitectos de software, que utilicen la arquitectura cliente-servidor.

**5.1.2 Hipótesis.** El desarrollo de una herramienta de administración, cuyo enfoque principal es la ayuda en la toma de decisiones con respecto a temas de seguridad en una arquitectura cliente servidor, resultará benéfica para los desarrolladores de software.

**5.1.3 Variables.** Para poder realizar una adecuada corroboración de la hipótesis, anteriormente planteada, se tienen planteados tres variables dependientes, las cuales son: Facilidad de uso, utilidad percibida e intención de uso, además de estas variables, también se debe tener en cuenta que también se toma como variable independiente la herramienta de estrategia de la administración de conocimiento, cuyo objetivo es apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de la información. Estas tres variables dependientes

resultan ser claves debido a que con ellas es posible llegar a la conclusión de que si la herramienta propuesta Safety Place puede llegar a cumplir con la hipótesis propuesta.

## 5.2 DISEÑO DEL INSTRUMENTO.

El instrumento de medición, que se tiene planteado para recolectar la información necesaria, es una encuesta de Google Forms, esta herramienta fue seleccionada dentro de muchas otras debido a su versatilidad y adaptabilidad lo cual jugo un papel sumamente relevante al momento de distribuir la encuesta creada entre los distintos participantes.

**5.2.1 Objetivo.** La meta que se tiene propuesta con el desarrollo del instrumento de medición es poder concluir el grado de efectividad, impacto, usabilidad, utilidad y demás factores que determinan si la herramienta de administración de conocimiento Safety Place, cumple con la hipótesis planteada.

**5.2.2 Preguntas.** Con el fin de determinar el grado de efectividad, usabilidad, utilidad e impacto que se puede llegar a generar con el uso de la herramienta de estrategia de la administración de conocimiento Safety Place, se generaron seis preguntas las cuales son:

- ¿Como arquitecto de software, la búsqueda de vulnerabilidades y sus respectivas contramedidas te pareció más sencilla con nuestra aplicación?
- ¿Qué tan molesto es el tener que buscar información respecto a una vulnerabilidad específica en la infinidad de wikis existentes y foros dedicados a esta temática?
- ¿Qué tan difícil fue el uso de la aplicación safety place para usted?
- ¿Con que frecuencia utilizarías la aplicación?
- ¿Qué sugiere usted como desarrollador de software para mejorar la experiencia de usuario de nuestra aplicación Safety place?

Como se es posible apreciar en las anteriores preguntas, cada una de ellas evalúa un aspecto en específico de la aplicación, tal como puede ser la utilidad que llego a percibir el usuario con el uso de la aplicación, también se permite evidenciar el como se le plantea al usuario una calificación a la utilidad que pudo percibir con el uso y objetivo de la aplicación, no solo eso sino que también fue posible cuestionar la facilidad que el usuario llego a experimentar en la interacción que tuvo con la aplicación.

**5.2.3 Escala de medición.** Para la escala de medición, del instrumento realizado en la plataforma de Google Forms, se tiene planteado utilizar una escala de Likert, este tipo de escala consiste en plantear un cuestionamiento al usuario sobre el grado en el que el usuario puede estar de acuerdo o en desacuerdo a una declaración, pero la escala de Likert, no solo se limita a un sí o aun no, sino que plantea que el nivel en el usuario pueda estar de acuerdo o en desacuerdo con una declaración planteada, pueda ser calificado con una escala del uno al cinco. Esta escala toma su nombre por el psicólogo Rensis Likert. Likert diferenció entre una escala conveniente, la cual surge de las respuestas colectivas a un conjunto de ítems, y el formato en el que las respuestas son puntuadas en un rango de valores[35].

La escala de Likert pertenece a los tipos de escalas de medición usados esencialmente en la indagación de mercados para la comprensión de las opiniones y reacciones de un consumidor hacia una marca, producto o mercado meta. Esto resulta de mucha utilidad dado que primordialmente para hacer mediciones y conocer sobre el nivel de conformidad de una persona o encuestado hacia definida sentencia afirmativa o negativa.

Figura 9. Escala de Likert

[35] Maldonado, Sandra, Manual Practico para el diseño de la escala Likert. (2007)


**Fuente: Tomado de https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/**

Las respuestas que se pueden encontrar en la escala de Likert pueden llegar a ser ofrecidas en distintos niveles de medición o escalas, está escala la mayoría del tiempo debe de tener un elemento neutral, para que aquellos usuarios a los que se les plantee algún tipo de encuesta que presente la escala ya mencionada puedan adoptar una posición de no estar desacuerdo ni en desacuerdo.

Como es posible apreciar en la anterior imagen, se está tomando un ejemplo de respuestas con la escala de Likert en donde se le da al usuario la posibilidad de adoptar una posición neutral, pero no solo eso, sino que también se nos presenta el cómo se toman los resultados, cundo en las dos menores opciones por así decirlo, se considera que se obtuvo un resultado negativo, o al expresarlo de mejor manera, el usuario no está en de acuerdo con la declaración propuesta, al contrario en la otra parte de la imagen, cuando se obtiene una respuesta neutral o posición en desacuerdo con la declaración, se podría decir que se obtiene un resultado positivo.

## 5.3 EJECUCIÓN.

Se tiene planificado que, para llevar a cabo la implementación y uso de la herramienta de medición planteada, se tome en una primera instancia un grupo de muestra de un tamaño de diez personas, estas personas no pueden ser personas con habilidades básicas, al expresarlo de mejor manera las diez personas que deben de participar, deben de tener una serie de habilidades afines arquitectura

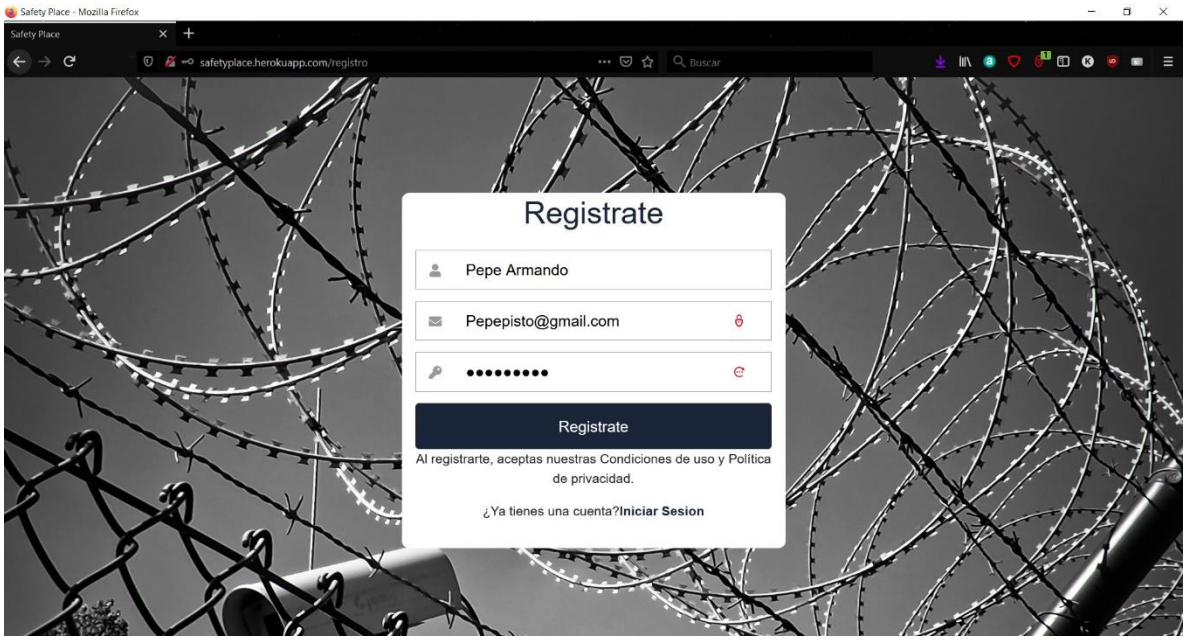
de software, esto debido a que el enfoque que se le da a la herramienta es netamente para el apoyo en la toma de decisiones en cuanto a temas de seguridad informática se trata, en específico en la arquitectura de software cliente-servidor. Debido al objetivo de la herramienta Safety Place es tan especifico, es muy difícil que personas que no estén instruidas en temas como arquitectura de software, seguridad informática y desarrollo de aplicaciones les sea supremamente difícil el entender la funcionalidad, impacto y usabilidad que tiene la herramienta.

**5.3.1 Guía de uso de la aplicación.** Para dar un correcto uso a la aplicación se dará una orientación, de cómo se utiliza la herramienta Safety Place, para así poder hacer uso de todo el potencial que esta herramienta puede llegar a dar. Como requisitos mínimos para aplicación se tienen parámetros que se pueden considerar básicos, pero igual tienen cierto grado de relevancia por lo cual serán mencionados, dichos parámetros son:

- Conexión a internet. Este parámetro es de suma importancia debido a que la herramienta de estrategia de la administración de conocimiento para apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de la información, esta herramienta Safety Place se encuentra aloja en un host llamado Heroku, lo cual permite el fácil acceso a la aplicación desde cualquier parte.
- Uso de ciertos navegadores web. Este parámetro tiene cierto grado de importancia debido a que el aplicativo ha presentado inconvenientes en navegadores web poco reconocidos tales como: Brave y Dolphin, debido a ello se recomienda el uso de navegador web tales como los son: Google Chrome, Opera, Microsoft Edge, Firefox Mozilla y Safari.

Como primer paso para hacer un uso adecuado de la aplicación Safety Place, el usuario se debe registrar, en la pagina de registro se le pedirán al usuario datos como: nombre completo, correo electrónico y una contraseña.

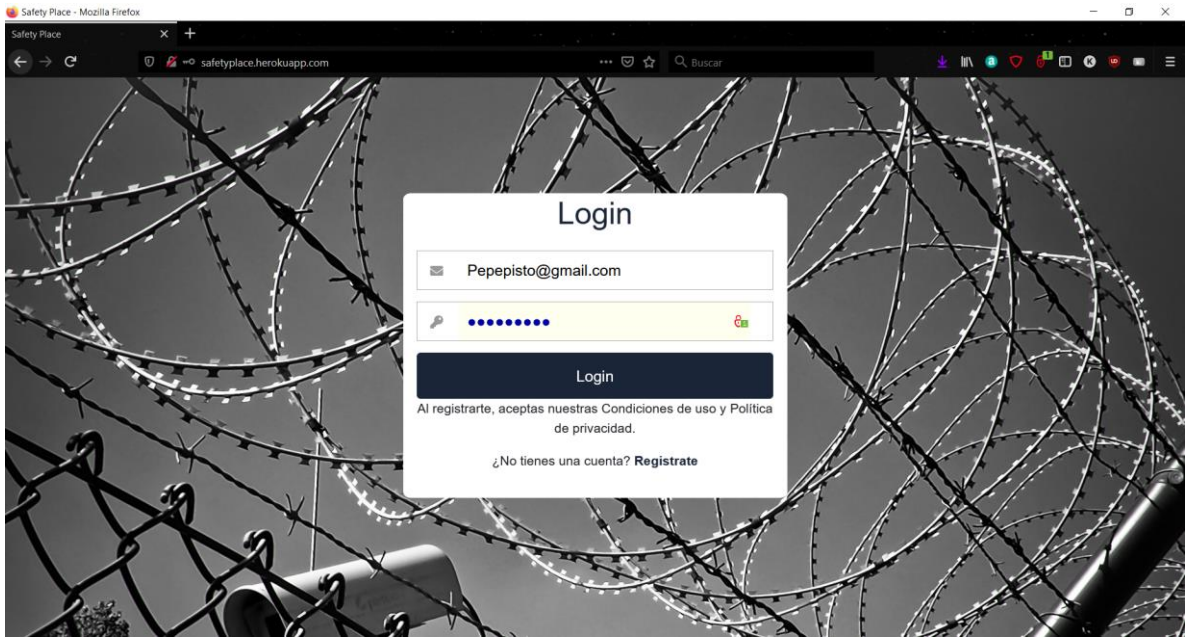
Figura 10. Registro en SafetyPlace




Fuente: Tomado de http://safetyplace.herokuapp.com

Ya una vez el usuario llene los datos que son requeridos para el registro de un usuario nuevo, será regresado a la página principal, en donde será requerido que inicie sesión con los datos que se le pidieron anteriormente.

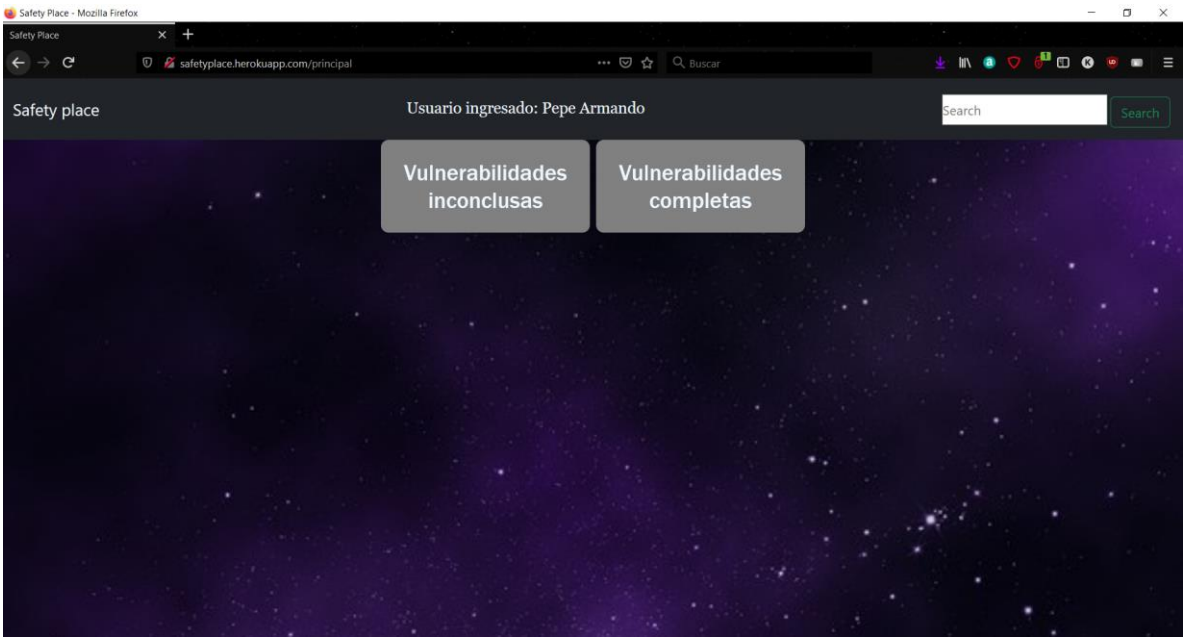
Figura 11. Inicio de Sesión



Fuente: Tomado de http://safetyplace.herokuapp.com

Una vez digitados toda la información correspondiente al inicio de sesión, se da un click en el botón “login” en dado caso que algún dato ingresado sea erróneo disparará un anuncio que será posible verlo en arte inferior, en el dirá que las credenciales ingresadas son erróneas, este aviso pasado unos cuantos segundos desaparecerá. En caso de que alguna de las credenciales que se ingresaron para el inicio de sesión sean erróneas, basta con corregirlos he intentar nuevamente con el botón “login”. Ya iniciada la sesión, se redireccionará a la página de inicio que es única para cada usuario.

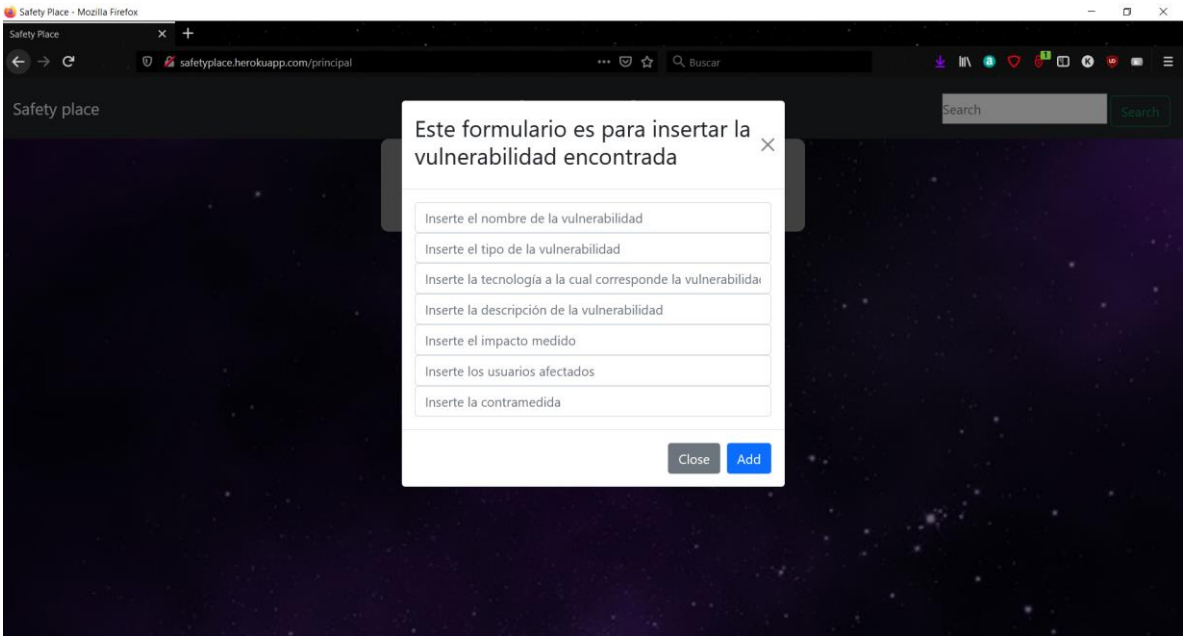
Figura 12. Página de Inicio




Fuente: Tomado de http://safetyplace.herokuapp.com

En esta página de inicio, se podrá apreciar que existen dos columnas en las que dicen: Vulnerabilidades inconclusas y Vulnerabilidades Completas, esto haciendo referencia, a que la aplicación no solo se limita a presentar las vulnerabilidades las cuales ya se tiene gran parte o toda la información recopilada, sino que también permite agregar vulnerabilidades de las cuales les hace falta información. En dado caso que se desee agregar una vulnerabilidad ya sea que se quiera agregar completa o incompleta, solo basta con hacer doble click en cualquiera de las columnas.

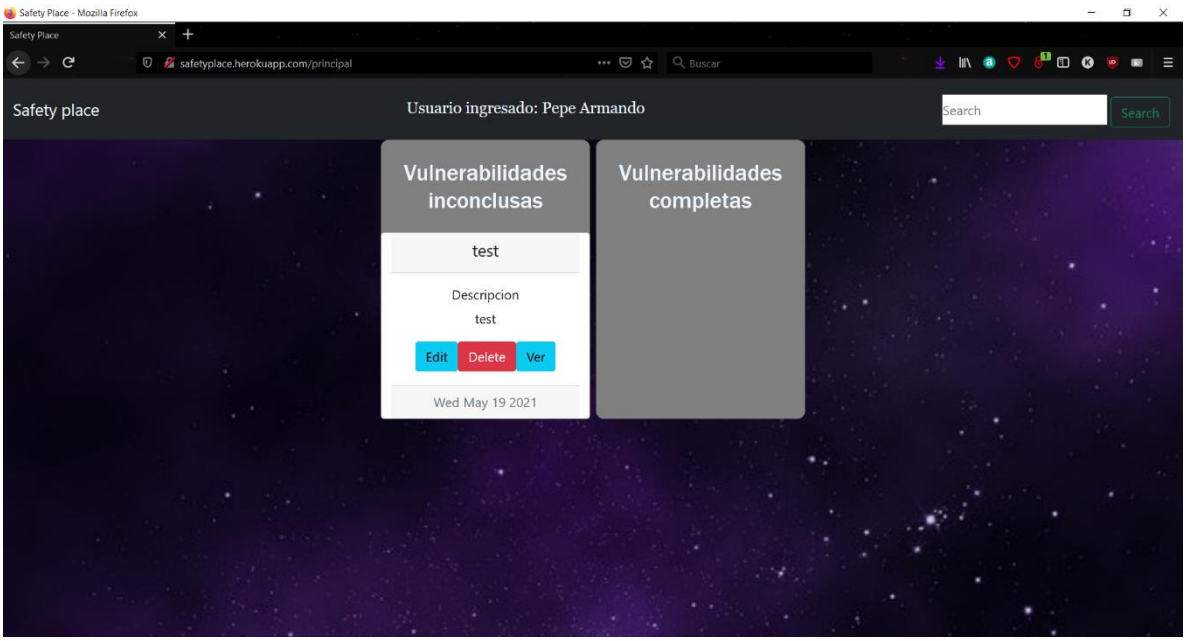
Figura 13. Agregar Vulnerabilidades




Fuente: Tomado de http://safetyplace.herokuapp.com

Cuando se desea agregar una vulnerabilidad, ya sea completa o incompleta, se desplegará el formulario que se puede agregar en la imagen, en donde se puede agregar información como puede ser: nombre de la vulnerabilidad, tipo de vulnerabilidad, tecnología a la cual correspondo o afecta esta vulnerabilidad, descripción de lo que puede llegar a hacer dicha vulnerabilidad, el impacto que puede llegar a tener, los usuarios que pueden llegar a ser afectados por la vulnerabilidad y la contramedida. Como ya fue mencionado no es necesario tener todos los ítems para agregar una vulnerabilidad, ya una vez llenados los campos deseados, se da click en el botón “Add”, así creando una tarjeta en una de las columnas.

Figura 14. Creación de tarjetas

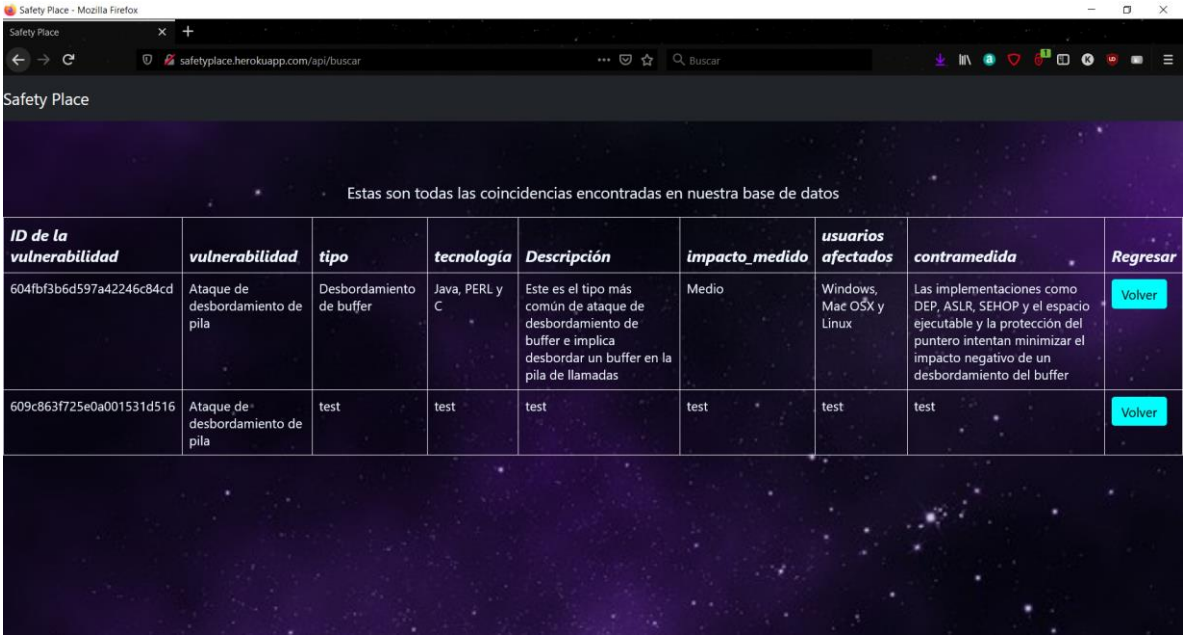



Fuente: Tomado de http://safetyplace.herokuapp.com

Ya realizando todos los pasos necesarios para agregar una nueva vulnerabilidad, se consigue la adición de una nueva tarjeta, que el usuario agrego, al momento de que el usuario realiza esta acción, la vulnerabilidad se agrega directamente a la base de datos, para así poder ser consultada no solo por él, sino que por los demás usuarios.

Para la consulta de vulnerabilidades, se dispuso de una barra de búsqueda, que básicamente lo que haces es hacer una consulta en la base de datos, entre se podrá consultar no solo las vulnerabilidades de otros usuarios sin que también otras que
ya estaban definidas con anterioridad.

Figura 15. Búsqueda de vulnerabilidades




Fuente: Tomado de http://safetyplace.herokuapp.com

En la anterior imagen se puede apreciar, que al momento de que el usuario desee buscar una vulnerabilidad, aparecerá esto en su pantalla, una tabla con todas las coincidencias de lo que el desea buscar, en donde está el ID de la vulnerabilidad, el nombre de la vulnerabilidad, tipo de vulnerabilidad, descripción, el impacto que esta pueda llegar a tener, los usuarios que afecta y por último la contramedida.

Teniendo claro las características de funcionamiento de las que goza la herramienta Safety Place, se procede a preparar a el grupo de muestra que consiste de diez personas, las cuales, con anterioridad ya habían trabajado con la arquitectura cliente-servidor, para que así, al momento de que ellos entren a utilizar Safety Place puedan generar un contraste de experiencias, en donde la primera experiencia que ellos tuvieron fue donde trabajaron de manera “individual” por decirlo de alguna manera, y la nueva experiencia que genera el uso de aplicación.

Una vez que cada uno de los participantes haya hecho uso de la herramienta, tiene una idea clara, de cómo fue el haber trabajado sin ella y como es trabajar con ella, con esta idea fresca en cada una de las mentes de los participantes, se precederá a extender la herramienta de recolección de datos que tiene como propósito el evaluar aspectos fundamentales de la aplicación.

# 6. ANÁLISIS DE RESULTADOS

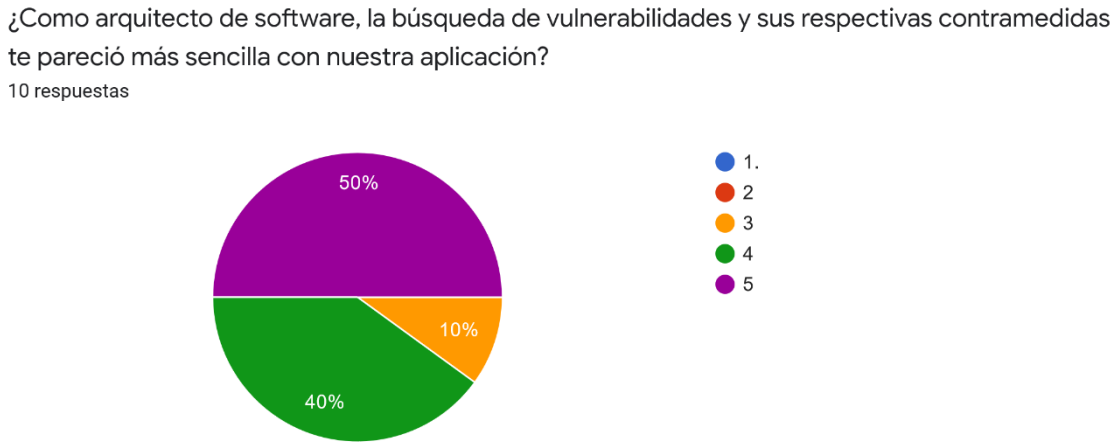
En este capítulo se dará un análisis descriptivo de los datos que se recolectaron gracias al instrumento de medición que fue diseñado, esto con el objetivo de dar una respuesta o validación a las variables que se plantearon en un principio, para poder corroborar la eficiencia, impacto, usabilidad y utilidad, que puede llegar a tener la herramienta de estrategia de la administración de conocimiento para apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de la información, cuyo nombre es Safety Place.

## 6.1 RESULTADOS POR PREGUNTA.

A continuación, se mostrarán los resultados de cada pregunta que fue planteada, para el instrumento de medición, esto con el objetivo d poder llegar a brindar un mejor análisis de los datos obtenidos. Las preguntas fueron planteadas de manera en que la respuesta sea un número del 1 al 5, donde el 1 sea la repuesta más baja y el 5 la más alta.

### 6.1.1 ¿Como arquitecto de software, la búsqueda de vulnerabilidades y sus respectivas contramedidas te pareció más sencilla con nuestra aplicación? .

Figura 16. Primera Pregunta




Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta se tiene planteado, la declaración de la sencillez que puede llegar a tomar no solo la búsqueda de vulnerabilidades, sino que también la búsqueda de la respectiva contramedida, según el anterior grafico es válido inferir que es mucho más sencillo este tipo de búsquedas con la herramienta Safety Place, esto debido a que la mitad del grupo muestral otorgo la máxima calificación, y no solo eso sino que también se cuenta con un 40% del grupo que le pareció más sencilla pero se encontraron con algunos obstáculos en el camino, sin embargo también hubo un participante que no la encontró ni tan sencilla ni tan difícil, se encuentra en un punto neutro.

### 6.1.2 ¿Qué tan molesto es el tener que buscar información respecto a una vulnerabilidad específica en la infinidad de wikis existentes y foros dedicados a esta temática? .

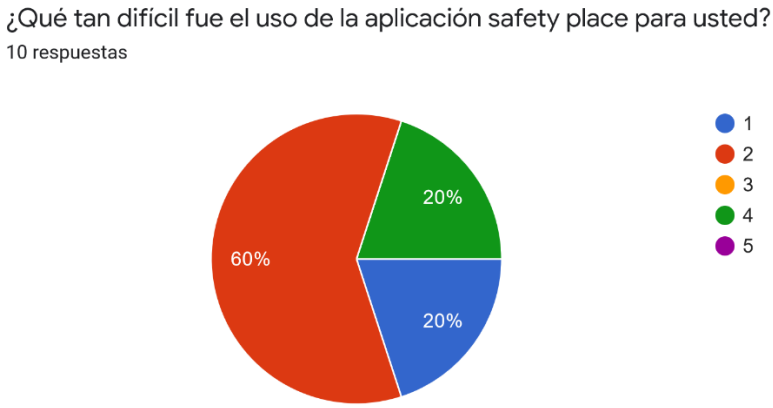
Figura 17. Segunda Pregunta


Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta, se plantea el caso de realizar, una labor de investigación de vulnerabilidades y contramedidas, pero en este caso sin la herramienta Safety Place, como es relejado en la gráfica, a más de la mitad del grupo le resulta muy molesto realizar este tipo de investigación, debido a que la información se encuentra muy dispersa haciendo que esta labor se vuelva muy extensa y tediosa.

### 6.1.3 ¿Qué tan difícil fue el uso de la aplicación safety place para usted? .

Figura 18. Tercera Pregunta

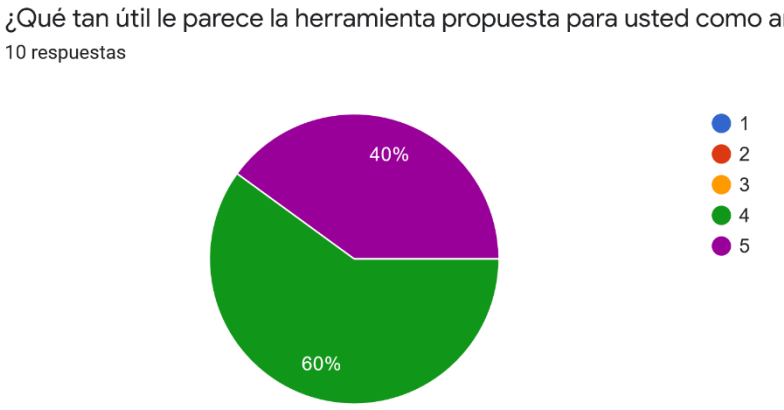



Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta se planteó un escenario sencillo, que no vas halla de su función, se plantea la declaración de la facilidad al momento de utilizar el aplicativo Safety Place, en donde un 80% del grupo muestral le resulto que es una aplicación sencilla para su uso.

### 6.1.4 ¿Qué tan útil le parece la herramienta propuesta para usted como arquitecto de software?.

Figura 19. Cuarta Pregunta

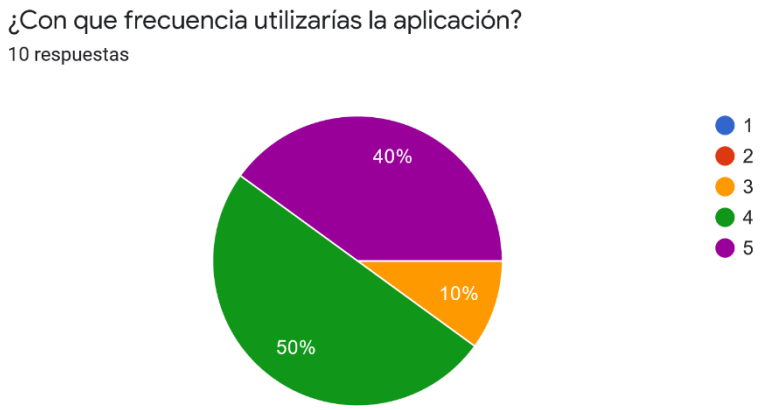



Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta se plantea la una declaración con respecto a la utilidad que los participantes percibieron durante la prueba, dando como resultado que todos ellos encontraron la herramienta útil, ni uno solo de ellos se encuentra en un punto neutro o negativo.

### 6.1.5 ¿Con que frecuencia utilizarías la aplicación?.

Figura 20. Quinta Pregunta

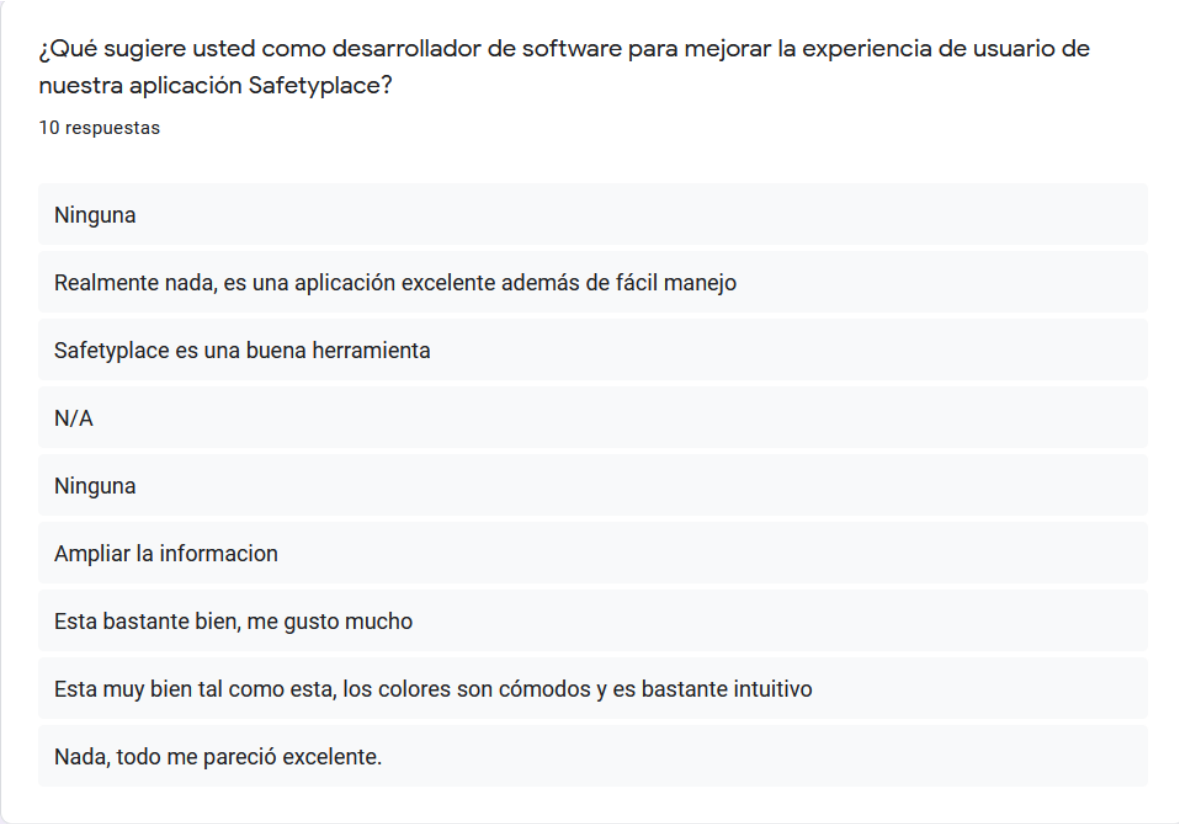



Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta se declara, la frecuencia de uso, que los arquitectos de software que formaron parte del grupo muestral le darían a la herramienta Safety Place, como resultado se obtiene que el 90% del grupo usaría la aplicación de forma recurrente y solo una persona una persona la usaría de manera casual.

### 6.1.6 ¿Qué sugiere usted como desarrollador de software para mejorar la experiencia de usuario de nuestra aplicación safety place?.

Figura 21. Sexta Pregunta




Fuente: Tomado de Encuesta SafetyPlace

En esta pregunta, a diferencia de las otras se planteó como una pregunta abierta, para así poder realizar una retroalimentación de la opinión de cómo se podrá mejorar la experiencia de usuario, teniendo en cuanta las repuestas suministradas por el grupo de muestra, se puede afirmar que un 90% de este grupo encuentra que la aplicación se encuentra en óptimas condiciones, al explicarlo de mejor manera, para estos arquitectos de software encuentran a la aplicación perfecta tal como está.

## 6.2 RESULTADOS POR CATEGORÍA.

Se llevará a cabo un análisis de los resultados obtenidos por medio del instrumento de medición, este análisis tiene como enfoque tres categorías en

específico, las cuales son: facilidad de uso percibida, utilidad percibida e intención de uso.

**6.2.1 Facilidad percibida.** En cuanto a la facilidad de uso que percibieron los usuarios con respecto a la aplicación Safety Place, es concluyente que para el 80% de los participantes, les resulto que a herramienta es sencilla de usar. Este factor de facilidad que percibe el usuario es de suma relevancia debido a que, si en dado caso un usuario llegara a encontrar que la herramienta es difícil de usar, que no sabe que esta haciendo con la herramienta, resulta que no se puedo facilitar una labor en concreto, esto haciendo referencia que el objetivo básico de una herramienta es facilitar una acción, en este caso la seguridad.

**6.2.2 Utilidad percibida.** Cuando se refiere a la utilidad que percibe un usuario, en cuanto a la herramienta, resulta que se evalúan aspectos tan fundamentales como puede ser la funcionalidad, en este factor puntual de la utilidad es posible inferir que el 100% del grupo de muestra que llego a utilizar la herramienta Safety Place encuentra que herramienta le resulta útil, al momento de que ellos ejerzan su función como arquitectos de software y que entre a juego un papel tan importante como lo es la seguridad.

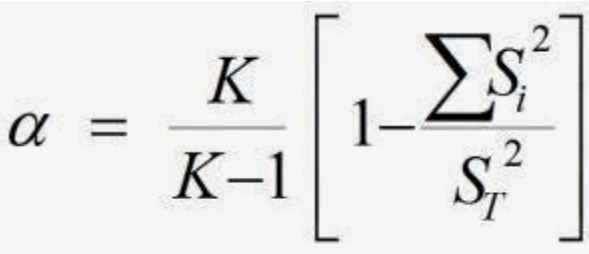
**6.2.3 Intención de uso.** La intención de uso que puede llegar a tener cada usuario que pertenece al grupo de muestra, tiene el objetivo que en el diseño de un aplicativo con arquitectura cliente-servidor, se tenga presente el tema de seguridad, el cual por medio de la herramienta Safety Place hace esto mucho más fácil. Se encuentra que el grupo muestral en su totalidad puede llegar a cumplir este objetivo.

## 6.3 ÍNDICE ALFA DE CRONBACH.

En el presente trabajo se tiene planteado interpretar si la hipótesis planteada, es rechazada o aceptada, esto se tiene planteado realizar por medio del índice alfa de Cronbach, el cual consiste en un investigador trata de medir una cualidad no directamente observable en una población de sujetos. Para ello mide n variables que sí son observables de cada uno de los sujetos.

En donde se supone que las variables están relacionadas con la magnitud inobservable de interés. En particular, las n variables deberían realizar mediciones estables y consistentes, con un elevado nivel de correlación entre ellas.

Figura 22. Formula




**Fuente: Tomado de https://www.redalyc.org/jatsRepo/773/77349627039/html/index.html**

Este índice de Cronbach consiste en evaluar un índice de consistencia interna, de valores que van desde 0.1 a 1.0, que sirve para revisar si el instrumento que se está evaluando recopila información defectuosa y por consiguiente nos llevaría a conclusiones erróneas o si hablamos de una herramienta fiable que hace mediciones estables y consistentes. Al momento de analizar el índice de consistencia interna, se toma que los valores entre 0 y 0.2 son calificados como de muy baja confiabilidad, los valores entre 0.2 y 0.4 se califican como de baja confiabilidad, mientras que si el valor se encuentra entre 0.4 y 0.6 es de confiabilidad moderada, los valores de 0.6 a 0.8 se califican como de moderada confiabilidad y por ultimo los valores de 0.8 a 1.0 se consideran de alta confiabilidad.

## 6.4 DISCUSIÓN.

En este capítulo, se dará respuesta a que la hipótesis sea rechazada o aceptada, la hipótesis que se planteó para este trabajo es: El desarrollo de una herramienta de administración, cuyo enfoque principal es la ayuda en la toma de decisiones con respecto a temas de seguridad en una arquitectura cliente servidor, resultará benéfica para los desarrolladores de software, por medio del índice alfa de Cronbach, es una herramienta que permite sin rodeos el determinar si una hipótesis es aceptada o rechazada, esto gracias a un análisis de los resultados obtenidos con el instrumento de medición, se obtuvo como resultado el índice alfa de Cronbach del instrumento de medición arrojo un valor de 0.69, en donde en el anterior capitulo se explica la escala de calificación que se utiliza en el índice alfa de Cronbach, se puede inferir que los datos que fueron obtenidos en el instrumento de medición se determina que son de moderada confiabilidad. Esto quiere decir que la hipótesis propuesta es aceptada.

# 7. CONCLUSIONES

Para concluir con el trabajo presentado, nuestra herramienta Safety Place ayuda a los arquitectos de programa a buscar, y añadir información sobre diversos tipos de vulnerabilidades brindándoles diferentes resoluciones presentadas como contramedidas, para contribuir a reducir el peligro de exponer vulnerabilidades en aplicaciones que se permanecen desarrollando o bien que ya permanecen terminadas sin embargo que muestran vulnerabilidades. Como antes se ha dicho, la iniciativa de nuestra aplicación es que la información crezca y sea enriquecida por los mismos usuarios que se usaran para la aplicación, brindando diversas resoluciones a una vulnerabilidad, y permitiéndole a los usuarios elegir la que más se adecue a su desarrollo. Como parte de un trabajo a futuro se podría llevar a cabo un sistema de sugerencias con base en ia (inteligencia artificial), que tenga como finalidad recomendar automáticamente una contramedida como solución a una vulnerabilidad específica. Paralelamente en la parte visual estaría realmente bien mejorar todas las vistas para que la vivencia de los usuarios sea muchísimo más placentera.

# BIBLIOGRAFÍA


ARIAS SÁNCHEZ, Pablo Xavier. Diseño de una red LAN/WAN segura para el Tribunal Constitucional aplicando la metodología de 3 capas de CISCO. DSpace [página web]. (2011). [Consultado el 13, mayo, 2021]. Disponible en Internet: <http://repositorio.puce.edu.ec/handle/22000/6371>.

BELCIC, Ivan. ¿Qué es el pharming y cómo protegerse de ataques? AVG [página web]. (7, noviembre, 2019). [Consultado el 31, marzo, 2021]. Disponible en Internet: <https://www.avg.com/es/signal/what-is-pharming>.

BOJACÁ GARAVITO, Edgar Alonso. Diseño de un sistema de gestión de seguridad informática basado en la norma ISO/IEC 27001- 27002 para el área administrativa y de historias clínicas del hospital san francisco de gachetá. - 10596/12685. Universidad Nacional Abierta y a Distancia UNAD - [página web]. [Consultado el 15, julio, 2021]. Disponible en Internet: <https://repository.unad.edu.co/handle/10596/12685>.

BONILLA, Sandra M. y GONZÁLEZ, Jaime A. Modelo de seguridad de la información | Ingenierías USBMed. Revistas Editorial Bonaventuriana [página web]. (30, junio, 2012). [Consultado el 27, febrero, 2021]. Disponible en Internet: <http://www.revistas.usb.edu.co/index.php/IngUSBmed/article/view/259>.

CAMELO PINZÓN, Javier Camilo. Seguridad informática en el sistema operativo android Y los riesgos presentes en internet. Unipiloto [página web]. [Consultado el 30, julio, 2021]. Disponible en Internet: <http://polux.unipiloto.edu.co:8080/00002698.pdf>.

CENTRO CRIPTOLÓGICO NACIONAL. Buenas prácticas en seguridad informática. CCN-CERT [página web]. (13, agosto, 2009). [Consultado el 24, mayo, 2021]. Disponible en Internet: <https://www.ccn-cert.cni.es/seguridad-al-dia/noticias-seguridad/819-buenas-practicas-en-seguridad-informatica.html>.GIMÉNEZ, Christian, *et al.* Una arquitectura cliente-servidor para modelado conceptual asistido por razonamiento automático. SEDICI - Repositorio de la Universidad Nacional de La Plata [página web]. (Abril, 2016). [Consultado el 26, febrero, 2021]. Disponible en Internet: <http://sedici.unlp.edu.ar/handle/10915/53044>.

DAVILA GOMEZ, Amalia. Identificacion y control de sistemas estocasticos con observaciones incompletas mediante modelos neurodifusos. Handle Proxy [página web]. (2012). [Consultado el 24, mayo, 2021]. Disponible en Internet: <http://hdl.handle.net/20.500.12622/139>.

DI LUCA, Marlon Altamirano. Modelo para la gestión de la seguridad de la información y los riesgos asociados a su uso. Sistema de Información Científica Redalyc, Red de Revistas

Científicas [página web]. (27, abril, 2019). [Consultado el 15, mayo, 2021]. Disponible en Internet: <https://www.redalyc.org/journal/6378/637869113010/html/>.

EL DIARIO DEL PROFESIONAL TI. Informe Kaspersky Lab e INTERPOL: 1 de cada 5 usuarios de Android experimenta ciberataques | Diario TI. El diario del profesional TI [página web]. (28, agosto, 2014). [Consultado el 20, agosto, 2021]. Disponible en Internet: <https://diarioti.com/informe-kaspersky-lab-e-interpol-1-de-cada-5-usuarios-de-android-experimenta-ciberataques/83656>.

ESPAÑA, MINISTERIO DE INDUSTRIA, MINERÍA Y TURISMO

FIGUEROA SUÁREZ, Juan A., et al. La seguridad informática Y la seguridad de la información. Polo del Conocimiento [página web]. (15, diciembre, 2017). [Consultado el 24, abril, 2021]. Disponible en Internet: <https://polodelconocimiento.com/ojs/index.php/es/article/view/420>.

FONSECA, Jose; VIEIRA, Marco y MADEIRA, Henrique. Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks. IEEE Xplore [página web]. (3, marzo, 2008). [Consultado el 18, junio, 2021]. Disponible en Internet: <https://ieeexplore.ieee.org/abstract/document/4459684>.

GUNNAR, Wolf. Tipos de ataque. RU-Económicas [página web]. (2017). [Consultado el 3, marzo, 2022]. Disponible en Internet: <http://ru.iiec.unam.mx/4047/>.

HERNÁNDEZ SAUCEDO, Ana Laura y MEJIA MIRANDA, Jezreel. Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones Web - Guide of attacks, vulnerabilities, techniques and tools for Web application | ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica. ReCIBE, Revista electrónica de Computación, Informática, Biomédica y Electrónica [página web]. (6, diciembre, 2017). [Consultado el 24, mayo, 2021]. Disponible en Internet: <http://recibe.cucei.udg.mx/index.php/ReCIBE/article/view/43>.

HERNÁNDEZ VILLARREAL, Yohan Esneider. Guía de remediación de vulnerabilidades informáticas para el software. Los libertadores Fundacion universitaria [página web]. (21, julio, 2017). [Consultado el 16, mayo, 2021]. Disponible en Internet: <https://repository.libertadores.edu.co/handle/11371/1300>.

JÁUREGUI TORRES, Diana Pilar. Implementación de un servidor ftp autentificado con cliente-servidorfilezilla. Repositorios latinoamericanos [página web]. (2014). [Consultado el 17, abril, 2021]. Disponible en Internet: <https://repositorioslatinoamericanos.uchile.cl/handle/2250/2794375>.

LA PORTA, Liarna. Malicious profiles – one of the most serious threats to iPhones. Jamf [página web]. (9, mayo, 2018). [Consultado el 20, marzo, 2022]. Disponible en Internet: <https://www.jamf.com/blog/malicious-profiles-come/>.

LACOON SECURITY INC. Threats to ios mobile devices. Idency - Secure Your Digital World [página web]. [Consultado el 21, mayo, 2021]. Disponible en Internet: <https://idency.com/wp-content/uploads/2014/08/Lacoon-White-Paper-iOS-Threats.pdf>.

LÓPEZ VALLEJO, Marco Roberto. Hacking ético. Vulnerabilidad de Sistemas Operativos en el acceso por contraseñas. CORE [página web]. (2017). [Consultado el 24, julio, 2021]. Disponible en Internet: <https://core.ac.uk/download/pdf/236645046.pdf>.

MACÍAS VALENCIA, David G. y QUIROZ ZAMBRANO, Silvia M. Seguridad en informática. Dialnet [página web]. (2017). [Consultado el 20, febrero, 2021]. Disponible en Internet: <https://dialnet.unirioja.es/servlet/articulo?codigo=6137824>.

MALDONADO LUNA, Sandra Margarita. Manual práctico para el diseño de la escala likert. Dialnet [página web]. (2007). [Consultado el 31, enero, 2021]. Disponible en Internet: <https://dialnet.unirioja.es/servlet/articulo?codigo=4953744>.

MARINI, Emiliano. El modelo cliente/servidor. Linuxito [página web]. (Septiembre, 2012). [Consultado el 26, junio, 2021]. Disponible en Internet: <https://www.linuxito.com/docs/el-modelo-cliente-servidor.pdf>.

MARRERO TRAVIESO, Yran. La Criptografía como elemento de la seguridad informática. E-LIS repository [página web]. (2003). [Consultado el 20, marzo, 2021]. Disponible en Internet: <http://eprints.rclis.org/5034/1/criptografia.pdf>.

MAUCAYLLE LEANDRES, Alex. Construcción de un modelo de red virtual para aplicar técnicas de hacking ético y poder analizar los eventos relacionados a la seguridad informática sobre una infraestructura virtual. Repositorio Institucional - UNAJMA [página web]. (24, julio, 2019). [Consultado el 28, febrero, 2022]. Disponible en Internet: <http://repositorio.unajma.edu.pe/handle/123456789/489>.

MECÍAS, Castro. Método para la especificación de requerimientos de seguridad del software como vía para la implementación de un proceso de desarrollo de software seguro - mersec. 1Library.Co [página web]. (10, junio, 2015). [Consultado el 28, febrero, 2021]. Disponible en Internet: <https://1library.co/document/ynlgdn1q-metodo-especificacion-requerimientos-seguridad-software-implementacion-desarrollo-software.html>.

MILANO, Pablo. Seguridad en el ciclo de vida del desarrollo de software. CYBSEC Security Systems - Capacitación [página web]. (12, julio, 2007). [Consultado el 24, abril, 2021].

Disponible en Internet: <http://www.cybsec.com/upload/cybsec_Tendencias2007_Seguridad_SDLC.pdf>.

MONSALVE-PULIDO, Julián Alberto; APONTE NOVOA, Fredy Andrés y CHAVES TAMAYO, David Fernando. Estudio y gestión de vulnerabilidades informáticas para una empresa privada en el departamento de Boyacá (Colombia). SciELO Colombia- Scientific Electronic Library Online [página web]. (Diciembre, 2014). [Consultado el 17, septiembre, 2021]. Disponible en Internet: <http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292014000200007>.

MONTESINO PERURENA, Raydel; BALUJA GARCÍA, Walter y PORVÉN RUBIER, Joelsy. Gestión automatizada e integrada de controles de seguridad informática. SciELO - Scientific Electronic Library Online [página web]. [Consultado el 21, agosto, 2021]. Disponible en Internet: <http://scielo.sld.cu/scielo.php?pid=S1815-59282013000100004&script=sci_arttext&tlng=pt>.

MONTOYA SALAZAR, Yeny Patricia y VANEGAS, Andrés Ferney. Análisis de vulnerabilidades en el sistema de seguridad físico E informático del departamento de policía caquetá. Universidad Nacional Abierta y a Distancia UNAD [página web]. [Consultado el 17, junio, 2021]. Disponible en Internet: <https://repository.unad.edu.co/jspui/bitstream/10596/25972/1/%20 ypmontoyas.pdf>.

MOURATIDIS, Haris y GIORGINI, Paolo. When security meets software engineering: a case of modelling secure information systems. ResearchGate [página web]. [Consultado el 16, junio, 2021]. Disponible en Internet: <https://www.researchgate.net/publication/222653864_When_security_meets_software_engineering_A_case_of_modelling_secure_information_systems>.

ORLANDO PHILCO, Luis Rosero. Electrónicas en línea y la criptografía como modelo de seguridad informática | philco | gaceta sansana. Publicaciones Cientificas de la Universidad Santa Maria [página web]. (2014). [Consultado el 5, febrero, 2021]. Disponible en Internet: <http://publicaciones.usm.edu.ec/index.php/GS/article/view/44>.

PEDRAZA GARCIA, Gilberto; ASTUDILLO, Hernan y CORREAL, Dario. A methodological approach to apply security tactics in software architecture design. IEEE Xplore [página web]. (21, julio, 2014). [Consultado el 19, junio, 2021]. Disponible en Internet: <https://ieeexplore.ieee.org/document/6860432>.

PESADO, Patricia y ETEROVIC, Jorge. eds. Computer science – CACIC 2020 [en línea]. Cham: Springer International Publishing, 2021 [consultado el 9, abril, 2021]. Disponible en Internet: <https://doi.org/10.1007/978-3-030-75836-3>. ISBN 9783030758356.J. Prieto, "Modelado estadístico y control de codificadores de vídeo", *Dialnet*, 2015. [Online]. Available: https://dialnet.unirioja.es/servlet/tesis?codigo=237289. [Accessed: 06- Feb- 2021].

REDDY KANDUKURI, Balachandra Reddy Kandukuri; PATURI V, Ramakrishna y RAKSHIT, Atanu. Cloud Security Issues. IEEE Xplore [página web]. (13, octubre, 2009). [Consultado el 24, mayo, 2021]. Disponible en Internet: <https://ieeexplore.ieee.org/document/5283911>.

REDDY KANDUKURI, Balachandra Reddy Kandukuri; PATURI V, Ramakrishna y RAKSHIT, Atanu. Cloud security issues. IEEE Xplore [página web]. (13, octubre, 2009). [Consultado el 24, mayo, 2021]. Disponible en Internet: <https://ieeexplore.ieee.org/document/5283911>.

ROA BANQUEZ, Katherine; MARTÍNEZ BARRERA, Crisman y CABRERA MARTÍNEZ, Carlos. Experiencias en el aula virtual como mediación pedagógica para el apoyo al aprendizaje en el espacio académico de lenguaje cliente servidor | CITAS. Revistas Universidad Santo Tomás - Colombia [página web]. (13, agosto, 2020). [Consultado el 13, mayo, 2021]. Disponible en Internet: <https://revistas.usantotomas.edu.co/index.php/citas/article/view/6075>.

ROMANIZ, Susana. Seguridad de aplicaciones web: vulnerabilidades en los controles de acceso. SEDICI - Repositorio de la Universidad Nacional de La Plata [página web]. (2008). [Consultado el 21, mayo, 2021]. Disponible en Internet: <http://sedici.unlp.edu.ar/bitstream/handle/10915/21581/1927+-+Seguridad+de+aplicaciones+web+vulnerabilidades+en+los+controles+de+acceso.pdf?sequence=1>.

SÁNCHEZ PRIETO, Iago. Herramienta de análisis automático de vulnerabilidades SSL. Biblos-e Archivo [página web]. [Consultado el 16, septiembre, 2021]. Disponible en Internet: <https://repositorio.uam.es/handle/10486/673437>.

SARANG, Na; HWANKUK, Kim y TAEEUN, Kim. A study on the classification of common vulnerabilities and exposures using naïve bayes. SpringerLink [página web]. (22, octubre, 2016). [Consultado el 10, julio, 2021]. Disponible en Internet: <https://link.springer.com/chapter/10.1007/978-3-319-49106-6_65>.

SARUBBI, Juan Pablo. Seguridad Informática: Técnicas de defensa comunes bajo variantes del sistema operativo Unix. 1Library.Co [página web]. (2008). [Consultado el 9, febrero, 2021]. Disponible en Internet: <https://1library.co/document/zg6845nq-seguridad-informatica-tecnicas-de-defensa-comunes-bajo-variantes-del-sistema-operativo-unix.html>.

SEGURIDAD INFORMÁTICA: hacking ético [Anónimo]. España: Eni, 2015. ISBN 9782746097247.

SHAKIRAT OLUWATOSIN, Haroon. Client-Server model. CiteSeerX [página web]. (Febrero, 2014). [Consultado el 19, junio, 2021]. Disponible en Internet: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1083.8741&rep=rep1&type=pdf>.

SOLARTE SOLARTE, Francisco Nicolás; ENRIQUEZ ROSERO, Edgar Rodrigo y DEL CARMEN BENAVIDES, Mirian. Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001. Revista Tecnológica - ESPOL [página web]. (31, diciembre, 2015). [Consultado el 26, junio, 2021]. Disponible en Internet: <http://www.rte.espol.edu.ec/index.php/tecnologica/article/view/456>.

STALLINGS, William. Sistemas operativos. [s.l.]: Editorial Limusa S.A. De C.V., 2002. 200 p. ISBN 9789681853006.

T, Aditya. Remote code execution using ICMP modified structured storage covert channels without elevation of privileges. IEEE Xplore [página web]. (2019). [Consultado el 24, mayo, 2021]. Disponible en Internet: <https://ieeexplore.ieee.org/document/8940485>.

TARAZONA, Cesar. Amenazas informáticas y seguridad de la información. HeinOnline [página web]. [Consultado el 9, junio, 2021]. Disponible en Internet: <https://heinonline.org/HOL/LandingPage?handle=hein.journals/dpencrim28&div=29&id=&page=>.

VIVAS, Héctor Luis, *et al.* Arquitectura de Software con websocket para aplicaciones web multiplataforma. RID-UNRN [página web]. [Consultado el 18, junio, 2021]. Disponible en Internet: <https://rid.unrn.edu.ar/handle/20.500.12049/148>.

VOUTSSAS, Juan. Preservación documental digital y seguridad informática. Scielo [página web]. (2010). [Consultado el 28, febrero, 2021]. Disponible en Internet: <http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S0187-358X2010000100008>.

WEI, Ke; MUTHUPRASANNA, M. y KOTHARI, Suraj. Preventing SQL injection attacks in stored procedures. IEEE Xplore [página web]. (8, mayo, 2006). [Consultado el 23, octubre, 2020]. Disponible en Internet: <https://ieeexplore.ieee.org/document/1615052?arnumber=1615052>.

XI, Kai, *et al.* A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. Wiley Online Library [página web]. (3, diciembre, 2010). [Consultado el 27, febrero, 2021]. Disponible en Internet: <https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.225>.

ZORNOSA FAJARDO, David Enrique. Método de análisis y selección de soluciones tecnológicas de seguridad informática para el sector empresarial y las organizaciones. Universidad Santo Tomás [página web]. [Consultado el 18, marzo, 2022]. Disponible en Internet: <https://repository.usta.edu.co/handle/11634/2769>.Digital Evidence and Electronic Signature Law Review, 6 (2009), pp. 153-157

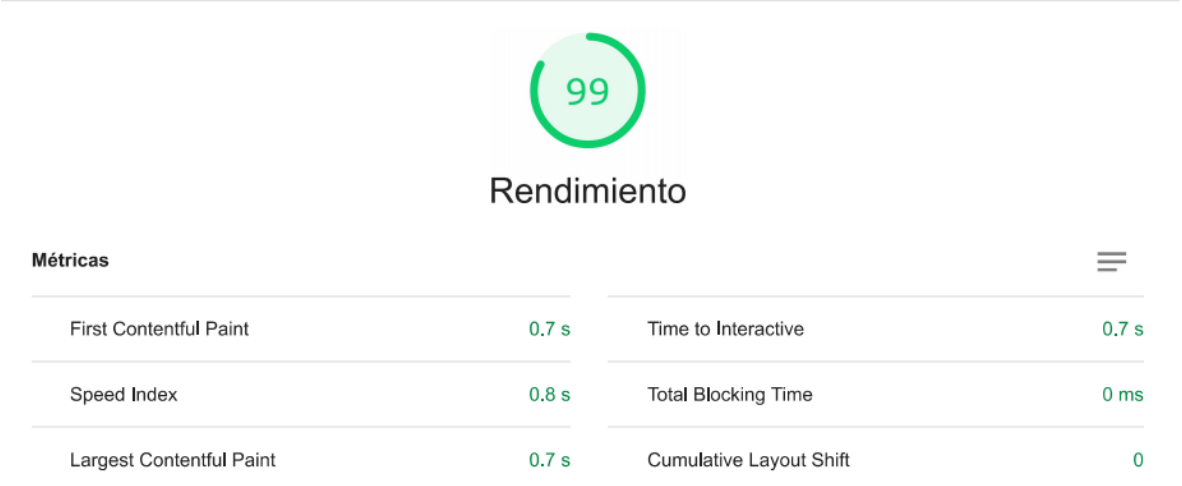
ZÚÑIGA SILGADO, Issac; PACHECO MENESES, Javys y MARTÍNEZ MOLINA, Kelly Johanna. Firewall-linux: una solución de seguridad informática para pymes (pequeñas y medianas empresas) | Revista UIS Ingenierías. Portal de Revistas UIS [página web]. (2, diciembre, 2019). [Consultado el 4, febrero, 2021]. Disponible en Internet: <https://revistas.uis.edu.co/index.php/revistauisingenierias/article/view/506>.

## ANEXOS

En este capítulo se mostrarán aspectos del desarrollo de la herramienta Safety Place, que no toman cierta importancia, debido a este factor se toma en cuanta para ser agregadas en este capítulo. Uno de los aspectos que se expondrán, son unas pruebas a las que se sometió la herramienta de administración de conocimiento Safety Place, en estas pruebas se determinó la accesibilidad, rendimiento.

### ANEXO A

Figura 23. Rendimiento



Fuente: Tomado de prueba lighthouse de la aplicación SafetyPlace

En la prueba de rendimiento que se realizado, se tomaron en cuenta aspectos fundamentales como velocidad, tiempo de respuesta, el tiempo que se toma en interactuar y demás aspectos que se pueden apreciar en la anterior imagen. El resultado de la prueba de rendimiento arrojo como resultado una puntuación de 99 que es una de las mayores calificaciones que se pueden obtener en el rendimiento de la aplicación Safety place.

Figura 24. Accesibilidad


Fuente: Tomado de prueba lighthouse de la aplicación SafetyPlace

En la prueba de accesibilidad que fue realizada, se tomaron en cuenta aspectos como la sintaxis del código html, que tiene que ver tanto como con el back-end y front-end, adicionalmente la aplicación que se utilizó para realizar estas pruebas ofrece la característica de consejos para mejorar en los aspectos testeados. En esta categoría se obtuvo un valor de 94 en la puntuación, que a diferencia de en la prueba de rendimiento, se obtuvo una puntuación menor, pero en la escala 1 a 100 en la que se califica cada categoría, un puntaje de 94 es bastante alto.

## ANEXO B

| Objetivo general | Objetivos específicos | Actividades | Semanas | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| Desarrollar una estrategia de administración de conocimiento para apoyar la toma de decisiones informada en diseño de arquitectura relacionada con la aplicación de contramedidas para bloquear amenazas a la seguridad de información | Comprender la dinámica de funcionamiento de la arquitectura de software Cliente-Servidor | Comprender el funcionamiento de la capa de | | | | | | | | | | | | | | |
| | | Comprender el funcionamiento de la capa de aplicación | | | | | | | | | | | | | | |
| | | Comprender el funcionamiento de la comunicación entre los componentes del cliente y del | | | | | | | | | | | | | | |
| | | Comprender el funcionamiento de la capa de datos | | | | | | | | | | | | | | |
| | Identificar las mayores amenazas para cada una de las capas de la arquitectura de software Cliente-Servidor | Identificar cuales son los recursos esenciales en cada una de las capas | | | | | | | | | | | | | | |
| | | Identificar las vulnerabilidades más recientes que pueden afectar a cada uno de estos recursos | | | | | | | | | | | | | | |
| | | Clasificar las amenazas según su impacto por medio del método DREAD | | | | | | | | | | | | | | |
| | | Clasificar el tipo de amenaza según el método STRIDE | | | | | | | | | | | | | | |
| | Definir la contramedida adecuada para cada una de las potenciales amenazas que se puedan presentar en cada capa de la arquitectura Cliente-Servidor | Investigar en los datasets gratuitos que contienen contramedidas adecuadas para las vulnerabilidades más conocidas | | | | | | | | | | | | | | |
| | | Investigar sobre las posibles soluciones a ataques de sql injection | | | | | | | | | | | | | | |
| | | Investigar sobre las posibles soluciones de ataques de tipo "elevation of privilege" | | | | | | | | | | | | | | |
| | | Investigar cómo contrarrestar ataques de tipo "Denial of service" | | | | | | | | | | | | | | |
| | Desarrollo de un prototipo funcional de una herramienta de administración de conocimiento que a partir de la definición de escenarios problémicos en seguridad responda con alternativas de decisión para solucionar preocupaciones de diseño de arquitectura | Desarrollar el backend | | | | | | | | | | | | | | |
| | | Desarrollar el Frontend | | | | | | | | | | | | | | |
| | | Asociar datasets donde se encuentre información de las vulnerabilidades específicas para el recurso | | | | | | | | | | | | | | |
| | | Desarrollar un sistema de recomendación para la toma de | | | | | | | | | | | | | | |

Figura 25. Cronograma de Actividades

Fuente: Elaboración propia