DECANATO DE INGENIERÍA E INFORMÁTICA

ESCUELA DE INFORMÁTICA

DEPARTAMENTO DE CURSO MONOGRÁFICO

Trabajo Final de Grado para optar por el título en

INGENIERÍA EN SISTEMAS DE COMPUTACIÓN

Título de la monografía:

Aseguramiento de la Calidad Orientado a la Seguridad de una Empresa de Desarrollo Web

Estudiantes:

Yrolf Sánchez Santana	**2005-0223**
Francis Johmel León Rosario	**2007-2221**
Alejandro Santos De Los Santos	**2010-2001**

Asesor:

Willy Alfredo Padua Ruiz

Coordinación Curso Monográfico: Dra. Sención Raquel Yvelice Zorob Avila

Distrito Nacional, República Dominicana

2021

DESARROLLO DEL TRABAJO FINAL

TABLA DE CONTENIDO

DEDICATORIA

A mi difunto padre, Alexis Sánchez Fernández, quien fue para mí, motor e inspiración en este muy largo viaje que esta presto a culminar. A su memoria por el deseo incansable de lograr ver este momento, sé que desde algún lugar privilegiado está viendo orgulloso en primera fila que si se puede.

Yrolf Sánchez Santana

A mi padre, Francisco A. León, quien siempre me ha motivado para que continúe y termine la carrera de Ingeniería en Sistemas.

A mi madre, Lidia Rosario, por todo su apoyo a lo largo de todos estos años.

Francis Johmel León Rosario

En primer lugar, gracias por el Dios que iluminó mi camino durante esta caminata. Dedico este trabajo a mi difunta Abuela, Santana Aquino Lorenzo, a mi madre Adelaida de los santos Aquino y mi familia por ser parte esencial en mi vida, motores de mis proyectos, guías y ayuda presente en el momento de los problemas que se me presentaron. Con gran agradecimiento, este trabajo es para ustedes.

Alejandro Santos De Los Santos

AGRADECIMIENTOS

Quisiera expresar un especial agradecimiento a mi esposa, Patricia Bruno, quien ha sido apoyo fundamental en esta recta final, nunca me dejó desfallecer por particularmente difícil que fueran las situaciones, en múltiples ocasiones renunciando a tiempo familiar y momentos de esparcimiento por brindar apoyo moral a esta causa.

A mi madre, María Elizabeth Santana, porque si llegué al lugar que estoy hoy y con el nivel que en la actualidad tengo fue por su orientación, enseñanzas y cuidados.

A Dios porque siempre mantuvo los medios disponibles para que pudiera continuar formándome en las distintas habilidades supliendo los recursos necesarios.

Yrolf Sánchez Santana

Agradezco a mi padre, Francisco León, por la motivación e insistencia para continuar una carrera que por razones personales abandoné en varias ocasiones. Hoy, gracias a él, tengo la convicción de seguir adelante y no detenerme hasta cerrar este capítulo. El presente trabajo es fruto de dicha determinación.

También agradezco a mí madre, Lidia Rosario, por todo su apoyo, especialmente durante los momentos difíciles. Quiero que sepas lo mucho que valoro el esfuerzo y apoyo que me has brindado hasta el día de hoy. Sin ti, no estaría aquí.

Finalmente, agradezco a Tatiana Maytet por su amistad, la cual ha perdurado desde que nos conocimos en el 2007 cuando empezamos juntos la carrera de Ingeniería en Sistemas. Gracias por siempre estar ahí para mí.

Francis Johmel León Rosario

Agradezco a Dios que sin él no tendría la fuerza para este proyecto, agradezco a mi madre Adelaida De los santos Aquino y colegas que me ayudaron a completar la monografía. También le agradezco a mi familia, que de una manera especial y cariñosa me ha dado fuerza y coraje, apoyando mis momentos difíciles y me iluminaron de una manera especial en esta recta final.

Alejandro Santos De Los Santos

RESUMEN

La presente investigación se efectuó con los objetivos de entender los conceptos de **ciberseguridad**, y como las estrategias de aseguramiento de calidad pueden ser utilizadas en las pruebas realizadas a las aplicaciones "**web**" para mejorar su seguridad, midiendo si se está en conformidad con los requerimientos del cliente y los estándares internacionales dispuestos por diferentes instituciones como NISP, OWASP y el Instituto Nacional de Ciberseguridad de Estados Unidos.

Se estudió la empresa *WebArtRD*, la cual opera en el campo de desarrollo de aplicaciones "web", en la República Dominicana. Se procedió a realizar un levantamiento de información para verificar si la empresa incluía pruebas de seguridad en los procesos de aseguramiento de la calidad realizados a las aplicaciones que desarrolla para sus clientes, con la finalidad de hacer una serie de recomendaciones que ayuden a la empresa a mejorar la seguridad sus aplicaciones. Con la implementación de un proceso formal de aseguramiento de la calidad enfocado en pruebas de seguridad de 'software' se pretende adecuar los procesos y procedimientos establecidos de manera institucional para los cambios de "software" y de esta forma minimizar la ocurrencia de inconvenientes en el ambiente de producción proveniente de brechas de seguridad en los sistemas de la plataforma "web" (Laskowski, 2011).

GLOSARIO DE TERMINOS

Término	Definición
Aplicación Web	Ferrer (2014) las define como un conjunto de aplicaciones, servicios y tecnologías con capacidad para operar entere sí utilizando el Internet, con la finalidad de intercambiar datos para ofrecer servicios a usuarios finales. Estos servicios "proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario".
Aseguramiento de la Calidad	Es una rama de la Ingeniería de "Software" que establece metodologias que permiten verificar el nivel de calidad de las aplicaciones de 'software' y si ese nivel esta en concordancia con lo previsto por el cliente, estando libre de vulnerabilidades. (Citado en Woody, Ellison, & Nichols, 2014, pág. 3).
Back-end	Termino anglosajón, y técnico, utilizado para referirse a los servicios 'web' que se ejecutan en el servidor antes de que el resultado se envíe a la interfaz (Mozilla, s.f.).
Calidad	Desde el punto de vista del usuario, se concibe la calidad en términos de las metas específicas que este tiene. Si un producto las satisface, se puede decir que tiene calidad (Pressman, 2010).
Ciberseguridad	"Conjunto de técnicas, procedimientos y protocolos encaminados a la protección de la información vinculada a los usuarios de las ciber tecnologías" (Arroyo Guardeño et al., 2020, pág. 12)
Creative Commons	Es una licencia que permite que organizaciones y compañías pongan a disponibilidad libremente sus creaciones intelectuales.
Defecto	Según la guía de ISO 9000 (s.f.), un defecto es una inconformidad derivada de no haber podido cumplir con un requerimiento estándar.
Desarrollador	Son los profesionales que escriben las instrucciones de los programas/aplicaciones, los cuales realizan tareas específicas como procesamiento de pedidos en línea, procesamiento de pago de una universidad, entre otras (IBM Corporation, s.f.).
Desarrollador Full Stack	Es una persona que puede desarrollar tanto la interfaz de la aplicación como los servicios 'web' (W3Schools, s.f.).

Desarrollo Ágil	Es una metodologia de desarro que divide los requisitos en funciones y cumple rápidamente con esas funciones a través del desarrollo incremental (IBM Corporation, s.f.).
Dirección URL	“no es más que una direccion que es dada a un recurso único en la Web… cada URL valida apunta a un único recurso. Dichos recursos pueden ser páginas HTML, documentos CSS, imagenes, etc.” (Mozilla, s.f.)
Documento HTML	es un documento de texto no cifrado estructurado con elementos. Los elementos están rodeados de etiquetas de apertura y cierre coincidentes. Cada etiqueta comienza y termina con corchetes angulares (<>). Hay algunas etiquetas vacías o vacías que no pueden incluir ningún texto, por ejemplo, <img> (Mozilla, s.f.).
Estandar de Calidad	Modelos y prácticas que permiten a los equipos de desarrollo de ‘software’ hacer pruebas con miras a mejorar la calidad de los productos (Carrizo & Alfaro, 2018).
Framework	Los frameworks son herramientas que hacen más fácil escribir, mantener y escalar aplicaciones web (Mozilla, s.f.).
Front-end	Termino técnico utilizado para referirse a la interfaz del usuario del sitio web que los usuarios pueden utilizar (Mozilla, s.f.).
Hacker	Es una persona que explota las vulnerabilidades de un sistema informático. (CISCO, s.f.).
HTML	“HTML (Lenguaje de marcado de hipertexto) es un lenguaje descriptivo que especifica la estructura de la página web” (Mozilla, s.f.).
Inteligencia Artificial	Es una rama de la Ingenieria que se enfoca en lograr que el ‘software’ emule la toma de decisiones y el aprendizaje de los seres humanos (IBM Corporation, s.f.).
Interfaz de Usuario	Son los controles y mensajes que el usuario de una aplicación puede visualizar en su pantalla (Mozilla, s.f.).
Internet	Es un conjunto descentralizado de redes de comunicación interconectadas que utilizan los protocolos de comunicación especiales para conectarse (Jimeno Muñoz, 2019).
JavaScript	Es un lenguaje de programación para aplicaciones web
JEST	Jest es un framework para pruebas de Javascript (JEST, s.f.).
JIRA	Es una aplicacion diseñada para que los equipos de desarrollo y calidad de software puedan planificar, supervisar y publicar las pruebas de calidad (Atlassian, s.f.).
Laravel	Laravel es un framework para aplicaciones web (Laravel LLC, s.f.) .

Lenguaje de Programación	Son herramientas utilizadas para escribir instrucciones que las computadoras puedan entender (Code Academy, 2020).
PHP	"Es un lenguaje de scripting de servidor y una poderosa herramienta para crear páginas Web dinámicas e interactivas" (W3School, s.f.).
Plan de Prueba	Es un documento que sirve como guía para realiza las pruebas de calidad pues contiene los pasos para verificar si los resultados corresponden a los requerimientos.
Probador (Tester)	Según Tester-h (2019), es quien lleva a cabo las pruebas al sistema y se encarga de verificar que dicho software se encuentra de acuerdo con su especificación y cumple con los requerimientos levantados.
Protocolo HTTP	es un protocolo que establece el método para intercambiar o transferir datos entre redes informáticas (Mozilla, s.f.) .
Protocolo HTTPS	Es un protocolo para intercambiar o trasferir datos entre redes informáticas. Es una versión más segura del protocolo HTT (Mozilla, s.f.).
Pruebas	Las pruebas del software, según Pressman (2010), son una función del control de calidad que tiene un objetivo principal: detectar errores.
Pruebas de aceptación de usuario	son pruebas realizadas por el usuario final para verificar que el sistema se apega a los requerimientos levantado por este (Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management, s.f.).
Pruebas de rendimiento	son pruebas realizadas para verificar cualidades como la respuesta del sistema a excesiva de usuarios, que tan rápido responde el sistema, entre otros (Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management, s.f.).
Pruebas de seguridad	Pruebas cuya finalidad es proteger el sistema (Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management, s.f.).
Pruebas funcionales	su objetivo es verificar la funcionalidad de la aplicación (Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management, s.f.).
Requerimiento	Campderrich Falgueras (2013) afirma que los requerimientos, o requisitos, son especificaciones que describen el comportamiento, las propiedades y restricciones del "software".
Scripting	Es la técnica de utilizar lenguajes especiales para desarrollar componentes que se utilizan en aplicaciones.

	Esta tecnica es utilizada en conjunto con programas de aplicación (Ousterhout K.).
Servicio Web	Puede ser fisico (‘hardware’) o de ‘sofware’. Un servidor web de ‘hardware’ es un equipo que almacena apliciones y los archivos que componen de un sitio web. Un servidor web de ‘software’ es incluye componenetes que controlan como una aplicación web accesa los datos (What is a web server?, s.f.).
Servidor	Los servidores son equipos que albergan aplicaciones y servicios compatibles con esas aplicaciones, como bases de datos (Dell Corporation).
Servidor ‘web’	Es un sistema que “…sirve para atender y responder a las diferentes peticiones de los navegadores, proporcionando los recursos que soliciten usando el protocolo HTTP o el protocolo HTTPS (la versión cifrada y autenticada)” (Ferrer Martínez, 2014, pág. 18).
Software	Es el conjunto de instrucciones que forman un programa/aplicación que le comunican a un ordenador/computadorqué hacer (IBM Corporation, s.f.).
SQL	“SQL (Lenguaje de consulta estructurado) es un lenguaje informático descriptivo diseñado para actualizar, recuperar y calcular datos en bases de datos basadas en tablas” (SQL, s.f.).
XML	es un formato de texto diseñado para intercambiar datos a través de diferentes sistemas conectados a una red a gran escala (liam, 2016).

ACRÓNIMOS

Acrónimo	Significado
ASC	Aseguramiento de La Calidad de ‘Software’
TI	Tecnología de la Información
QA	Acrónimo del inglés Quality Assurance (Aseguramiento de Calidad en español)
IA	Inteligencia Artificial

INTRODUCCION

La presente investigación tiene como objetivo aportar conocimiento sobre la seguridad de ‘software’ y como las prácticas de **aseguramiento de la calidad** pueden ayudar a las empresas de software a brindar mejores productos que se ajusten a las necesidades de sus clientes, brindándoles una mejor calidad, específicamente en el área de **ciberseguridad**, la cual es importante ahora más que nunca debido a la era de globalización en la cual vivimos; una época donde la interconexión a través de aparatos electrónicos, usando la Internet, es un hecho.

Según Carrizo & Alfaro (2018), el objetivo de toda empresa es entregar servicios y productos que satisfagan las necesidades de sus clientes, las cuales variaran dependiendo de un conjunto de variables que van desde la edad, el sexo, la etnia y la cultura del cliente, hasta sus preferencias y su educación. Entregar un producto que cumpla con dichas necesidades, frecuentemente en la forma de requerimientos contractuales, es lo que se denomina **calidad** (Carrizo & Alfaro, 2018). Para velar por su cumplimiento, las empresas se valen de metodologías y técnicas que en su conjunto se conocen como **Aseguramiento de la Calidad**.

Las empresas de desarrollo de ‘software’ no están exentas de la necesidad de entregar productos que cumplan con las necesidades y demandas de sus clientes, debido a que, en la actualidad el “software” tiene una importancia dual. “Es un producto y al mismo tiempo es el vehículo para entregar un producto” (Pressman, 2010, pág. 2). Las aplicaciones toman datos y los convierten en información, la cual es el activo más importante de nuestra época, según Pressman. Esto ha provocado que la industria se haya convertido en un factor dominante en la economía mundial, obligando a las empresas a mantenerse a la vanguardia a la hora de desarrollar sus aplicativos.

Existen actualmente un conjunto de modelos y estándares - como el modelo CMMI – los cuales trazan las pautas para que los equipos de desarrollo de ‘software’ puedan realizar pruebas para mejorar la calidad de los productos (Carrizo & Alfaro, 2018). Sin embargo, muchas empresas toman el camino enfocando sus esfuerzos de aseguramiento de calidad solo a ciertas funcionalidades, como la interfaz de usuario o la lógica del negocio, pasando por alto verificar otros niveles factores de calidad, como lo es **ciberseguridad**. Esto se debe, según los autores, a obstáculos como presupuesto limitado, estructura organizacional pequeña o carencia de personal capacitado.

Los defectos relacionados con la seguridad deben considerarse un problema de control de calidad (Wisseman, s.f.). El autor argumenta que el

“software” con defectos, fallas de calidad y código de mala calidad, es más propenso a tener vulnerabilidades de seguridad, lo cual, y desde la perspectiva del usuario, se manifiesta como mala usabilidad. Para un atacante, brinda la oportunidad de hacer cosas inesperadas con el sistema, como estresarlo, acceder a la información, o inhabilitarlo. Laskowski (2011), a su vez, explica que las aplicaciones web, por ejemplo, son explotadas debido a los agujeros de seguridad que existen en ellas.

En ese sentido, Laskowski afirma que aquellos equipos de tecnología que hacen uso del aseguramiento de la calidad enfocado a la seguridad tienen mejores resultados. Dicha premisa quedó demostrada en un estudio llevado a cabo en 2014 por Woody et al., para el Instituto de Ingeniería de Software de Pittsburg, el cual arrojo resultados contundentes que prueban que los equipos de desarrollo que aplican técnicas de aseguramiento de la calidad orientadas a la ciberseguridad tendían a tener productos con una mayor calidad y menos vulnerabilidades. Demostrando así que la **calidad** y la **seguridad** del software no deben tratarse por separado.

En República Dominicana existen precedentes de modelos de aseguramiento de la calidad utilizados por empresas como *Newtech S.R.L* y *Argentum Inc.*, los cuales toman en cuenta la seguridad. Newtech particularmente es una empresa con una larga trayectoria ofreciendo varios servicios, entre ellos

Desarrollo y Aseguramiento de la Calidad de “Software”. Cuenta con un equipo calificado en estándares internacionales como **ITIL y ISO9001: 2000**, con certificaciones de los Institutos de **Garantía de Calidad y Gestión de Proyectos**. Argentum por su parte labora en el mercado dominicano desde 2007, también ofreciendo diversos servicios de TI, entre ellos el aseguramiento de la calidad, usando estándares como ISO 9000 y herramientas certificadas y avaladas por IBM (IMB Rational Tester).

Sin embargo, también existen en el país empresas que prestan servicios de ‘software’ las cuales no cuentan con un modelo robusto de aseguramiento de la calidad, y mucho menos con modelo que incluya pruebas de seguridad. Este es el caso de la empresa WebArtRD, la cual ha sido selecciona como caso de estudio, para analizar y proponer un modelo existente de pruebas para seguridad de ‘software’ para ser incorporado a las pruebas de calidad que dicha empresa realiza a sus aplicaciones, permitiéndoles entregar producto más robusto, en términos de ciberseguridad, a sus clientes.

Para ello se ha diseñado la siguiente estructura: en el Capítulo I se habla sobre las principales amenazas a las cuales las empresas desarrolladoras de ‘software’ se enfrentan, así como las tendencias de las pruebas de seguridad en el aseguramiento de la calidad. Por último, se realiza el diagnostico en la empresa WebArtRD. En el Capítulo II se exploran las metodologías y tipos de pruebas de

seguridad que pueden ayudar a una empresa a mejorar su seguridad de ‘sofware’, enfocándonos especialmente en la guía OWASP. Finalmente, en el Capítulo III se enfoca en el plan de mejora para la empresa seleccionada y su valoración

CAPÍTULO I: LOS RIESGOS Y VULNERABILIDADES DE SEGURIDAD DE LAS APLICACIONES WEB

1.1. INTRODUCCIÓN

Los ‘hackers’ atacan aplicaciones web porque “…los equipos de control de calidad y los equipos de seguridad de TI no trabajaban juntos” (Laskowski, 2011), lo que genera enormes agujeros de seguridad, y esta ultima “… es un problema mundial” (Shaukat, Faisal, Masood, Usman, & Shaukat, 2016, pág. 1). La empresa de telecomunicaciones CISCO, en su artículo “¿Qué es un hacker? (s.f.) publicado en su página web, define al ‘hacker’ como una persona que vulnera un sistema informático.

Las aplicaciones web son una categoría de “software” que se definen por su capacidad conectar mediante redes una amplia gama de aplicaciones (Pressman, 2010). En su forma más sencilla, las aplicaciones web “… son poco más que un conjunto de archivos de hipertexto vinculados que presentan información con uso de texto y gráficas limitadas” (Pressman, 2010, pág. 9). Más allá de esto, desde que surgió Web 2.0, las aplicaciones han evolucionado a lo que se puede describir como ambientes sofisticados los cuales proveen funciones de cómputo complejo y contenido para el usuario final, ya integradas con bases de datos corporativas y aplicaciones de negocios (Pressman, 2010).

Los autores Urcuqui López et al. (2018) revelan que, en el año 2018, el Internet conectaba aproximadamente tres mil millones de usuarios en todo el

mundo. El uso y alcance del Internet permite que tanto las personas, como las entidades, intercambien información, y acceder a páginas web como Facebook, o la página web de una Universidad. A la vez que las facilidades de los seres humanos para usar la interconexión como método de comunicación, también han aumentado la cantidad de vulnerabilidades de los sistemas que hacen posible esa conexión, aumentando lógicamente los ataques.

En ese sentido, existen casos reales de vulnerabilidades que fueron explotadas por personas con objetivos criminales. Los autores Urcuqui López et al. (2018) destacan la frecuencia de los ataques de denegación de servicios, haciendo mención del ataque que sufrió la empresa DynDNS en octubre del 2016, proveedora tecnológica de empresas como Twitter, Spotify y Reddit, cuyas direcciones 'URL' fueron redireccionados por 'hackers', de tal manera que cuando un usuario intentaba cargar la página, esta se reportaba inaccesible.

Otro caso, reportado por Clay (2013), fue el de la corporación Amazon, la cual sufrió una caída de sus servidores, resultando así en la imposibilidad de que los usuarios accedieran a su portal web de compras, hecho que le costó a Amazon, y a su fundados Jeff Bezos, la suma de alrededor seis millones de dólares (aprox. 66,000 $US por minuto).

Ante el peligro que representan los ‘hackers’, es imperativo realizar pruebas orientadas a la seguridad durante en todo el ciclo de vida de desarrollo de software y no después, como tienden a hacer muchas empresas (Laskowski, 2011). En ese sentido, Laskowski explica que el equipo de control de calidad, ayudado por el equipo de seguridad - en aquellas empresas donde existan - deberá velar por definir requisitos no funcionales de seguridad, que son la base para que crear código más seguro. En el caso de aquellas empresas donde no exista un equipo de seguridad, Laskowski recomienda que la responsabilidad sea asumida por el equipo de aseguramiento de calidad, el cual deberá velar por la inclusión de casos de prueba para verificar que se controlan todos los riesgos posibles que fueron identificados. De lo contrario, advierte el autor, ignorar la seguridad presenta riesgos que pueden desencadenar en amenazas con resultados no deseados para las empresas.

1.2. ANTECEDENTES DE LOS ATAQUES A LA SEGURIDAD WEB

Un ciber ataque se puede definir como un ataque a la seguridad de una página o aplicación web. De ahí que se utilice el termino ciber, pues hace referencia al “ciberespacio” o Internet. Para Arroyo Guardeño et al. (2020) el ciberespacio es la constitución de un mundo lógico (información, datos y aplicaciones) en el mundo físico por medio denominados sistemas ciberfísicos.

Por ello, se escucha hablar de ciberamenazas, ciberdelitos, cibercriminales y ciberriesgo de manera intercambiada con los términos amenazas, criminales, delitos y riesgos, para diferenciarlos de sus contrapartes físicas en el mundo real. Es decir, para resaltar que el riesgo de perder la cartera es diferente al (ciber)riesgo de que una persona robe sus datos de tarjeta y saque dinero de su cuenta, ya que este último implica el uso de la tecnología. En la práctica, ambos son válidos cuando se habla del 'software'; por ejemplo: tanto riesgo como ciberriesgo es válido a la hora de hablar de riesgos tecnológicos.

El autor Jimeno Muñoz (2019) cita el primer ciber ataque documentado en noviembre de 1988, un programa gusano al que se denominó "El Gusano de Morris", el cual que funcionaba con un código auto replicante para así explotar vulnerabilidades en el correo electrónico del servidor software "Sendmail" causando el retraso y congestión de parte de la red. El autor afirma que la documentación de la prensa de la época dio indicios de que el motivo se trató de la curiosidad intelectual del autor material.

Ya luego, en los años 90, con el auge del correo electrónico y las herramientas de procesamiento de texto se produjo ataques con virus y gusanos a gran escala (Jimeno Muñoz, 2019). En ese sentido, el autor destaca dos ataques producidos en 2001, denominados "Código Rojo" (julio de 2001) y "NIMDA" (septiembre de 2001). Jimeno Muñoz afirma que ambos ataques se produjeron a

través de la explotación de vulnerabilidades en sistemas de aplicaciones web, los cuales realizaron daños importantes afectando a la disponibilidad de muchas redes comerciales en Internet.

Siguiendo a Jimenes Muñoz (2019), encontramos que en 2004 la técnica de la ingeniería social tomo relevancia. Lo cual condujo a la adopción de la tecnología de autenticación, la definición de riesgos y a la toma de conciencia colectiva para educar sobre la ciberseguridad, lo cual ayudó a establecerla como un aspecto clave. Finalmente, entre 2007 y 2008, los ataques denominados Denegación Distribuida de Servicios (DDoS por sus siglas en inglés) se intensificaron. En 2010, el autor cita un ataque cibernético denominado “Stuxnet” que afectó a una central nuclear iraní al explotar vulnerabilidades desconocidas (Jimeno Muñoz, 2019).

En el año 2013 hubo un incremento de los ciber ataques, que han dejado sus efectos en compañías como Google, Amazon, eBay, The New York Times, Facebook, Sony y Twitter, afectado a más de 100 millones de usuarios (Jimeno Muñoz, 2019). En ese sentido, las técnicas de ataque utilizadas hoy en día son similares a las ya descritas en los párrafos anteriores. Siguen destacando los ataques DDoS contra las entidades y organizaciones del sistema financiero (Jimeno Muñoz, 2019).

1.3. CONCEPTOS DE VULNERABILIDAD, RIESGO Y AMENAZA

Una **amenaza** es cualquier acción que comprometa alguno de los objetivos de seguridad - confiabilidad, integridad, disponibilidad, autorización, auditabilidad, autenticación (Arroyo Guardeño et al., 2020). Estos objetivos serán explorados más adelante en el subcapítulo ***La Seguridad de Software***. Por ahora, ilustraremos un ejemplo real de una amenaza: un correo malicioso, el cual contiene un virus, llega a los servidores de correo de una organización, representando el peligro de que un empleado abra el correo por error, desatando así el virus e infectando al computador o la red completa de la empresa.

Partiendo del concepto de amenaza, una **vulnerabilidad de 'software'** es toda aquella debilidad en un sistema que puede ser explotada por una cualquier amenaza (Arroyo Guardeño et al., 2020). Un ejemplo de vulnerabilidad es una empresa que no requiera a sus empleados que utilicen contraseña para ingresar al sistema, haciendo posible que cualquier persona no afiliada a la institución tenga acceso a la computadora de dicho empleado. Esta posibilidad es lo que se denomina **riesgo** - la probabilidad de que una vulnerabilidad de 'software' sea explotada por una amenaza (Arroyo Guardeño et al., 2020). En este último ejemplo de la empresa que no utiliza contraseña, la amenaza potencial sería una persona no autorizada accediendo a alguna computadora.

1.4. LAS FASES DE UN CIBERATAQUE

Al ciclo de vida de los ciberataques se les conoce con el termino anglosajón “Kill Chain” Arroyo Guardeño et al. (2020). Tiene varias fases que se describen a continuación en la siguiente tabla:

Fase	Descripción
Reconocimiento	El atacante recaba la información acerca de su objetivo. Ejemplo: sitio web, redes sociales, información personal, entre otras. Con esta información valora si su ataque puede tener éxito o no y si procede a su intento (Arroyo Guardeño et al., 2020).
Preparación	El atacante analiza y elabora el método de ataque basado en la información levantada en el paso anterior
Distribución	Se difunde la aplicación maliciosa que elegida por el ataque
Explotación	El atacante explota la vulnerabilidad con el método elegido.
Instalación	La aplicación maliciosa es instalada en el sistema atacado
Comando y Control	Es la fase en la cual el atacante tiene control sobre el sistema y puede realizar lo que quiere, como sacar información del sistema víctima.
Acción sobre objetivos	El atacante intenta atacar otros objetivos una vez obtiene la información que buscaba.

Tabla – Fases de un ciberataque. Autoría: propia.

1.5. MOTIVACIONES

Es conveniente conocer cuáles son las motivaciones y los perfiles de los atacantes, para tener una visión de a qué se enfrentan los usuarios y como combatirlo. Según la empresa Cisco (s.f.), existen diferentes tipos de hacker según sus objetivos:

Tipo de Hacker	Motivación
Cibercriminales	Son los hackers cuyos objetivos son no éticos, es decir irrumpen en un sistema para robar o destruir información.
Activistas	Irrumpen en sistemas de empresas y organizaciones políticas con el objetivo de hacer activismo social. Por ejemplo, para liberal al público información de actos de corrupción.
Éticos	Este tipo son profesionales de TI que utilizan sus conocimientos para ayudar a las empresas a encontrar vulnerabilidades de seguridad. Pueden trabajan como empleados directos o consultores de las organizaciones.
Terroristas	Utilizan los ataques a Internet con fines de financiación (como pedir rescate), coordinación, propaganda, reclutamiento y radicalización (Villalba Fernández, 2015).

Tabla – Tipos de hackers. Autoría: propia.

1.6. LOS TIPOS DE AMENAZAS DE ‘SOFTWARE’ WEB

Las aplicaciones web son vulnerables a numerosas amenazas, porque según Pressman (2010), las mismas se encuentran disponibles mediante las redes, lo cual dificulta el poder limitar la población de usuarios finales que pueden acceder a la aplicación, lo que da paso al riesgo de que puedan ser accedidas por personas las cuales no tienen permiso, o no deberían, vulnerando así la información.

El problema, según Wisseman, es que los equipos de control de calidad tienen dificultades para implementar técnicas para asegurarse que la seguridad de una aplicación web cumple un grado de seguridad debido a las diferencias entre las pruebas de seguridad y las pruebas funcionales y de rendimiento (Wisseman, s.f.). En las pruebas funcionales y de rendimiento, los resultados esperados para los casos de prueba se documentan antes de que comience la prueba, luego el equipo de aseguramiento de calidad analiza los resultados para ver si coinciden con los resultados esperados (requerimientos del cliente). Las pruebas de seguridad por otro lado velan por certificar que las probabilidades de vulnerar una aplicación web debido a debilidades del software son mínimas.

En otras palabras, las pruebas de aseguramiento de la calidad se enfocan en verificar si la aplicación funciona de acuerdo con lo esperado (¿cumple con sus

requisitos?) (Wisseman, s.f.). Por otro lado, la seguridad del software se asegura de que la aplicación es lo suficientemente robusta para no ser explotada por un "hacker" mediante vulnerabilidades como "inyección de SQL, scripts entre sitios, falsificación de solicitudes entre sitios, desbordamientos de búfer, uso de contraseñas codificadas, cifrado débil, datos confidenciales, entre otras" (Wisseman, s.f.). A continuación, una lista no exhaustiva de las amenazas de 'software' más comunes.

1.6.1. Malware

Arroyo Guardeño et at. (2020) lo define como "…todo tipo de software malicioso o dañino cuyo objetivo sea infectar ordenadores, tabletas o teléfonos móviles" (pág. 19). Por ejemplo, un programa que se instale sin autorización en el celular de una persona para robar información de ella es un tipo de "malware".

Principales diferencias entre virus, malware, antivirus y antimalware.

VIRUS/ANTIVIRUS	MALWARE/ANTIMALWARE
Un virus es un tipo específico de malware.	El malware se refiere a todo tipo de software dañino.
Todos los virus son malware.	No todo el malware es un virus.
Un antivirus es un software diseñado para detectar y destruir virus.	Un antimalware es un programa que protege el sistema de toda clase de malware, incluyendo virus, troyanos, gusanos, spyware, adware, etc.
Los antivirus no pueden proteger el sistema de formas avanzadas de programas de malware.	Los antimalware son capaces de defender el sistema de todo tipo de malware, ya sea clásico o avanzado.

Figura – Diferencias entre varios tipos de aplicaciones maliciosas. Fuente: Arroyo Guardeño et al. (2018). Pág. 25

1.6.2. Virus

Es un tipo de “malware” formado por código malicioso que infecta una aplicación en el sistema donde reside y se propaga una vez que se ejecuta (Arroyo Guardeño et al., 2020).

1.6.3. Gusano

En ingles ‘Worm’, es un tipo de ‘malware’ que se copia a si mismo al utilizar mecanismos de propagación como los correos o archivos compartidos (Arroyo Guardeño et al., 2020).

1.6.4. Trojano

Arroyo Guardeño et al. (2020) los describe como programas que se activan debido a ciertas circunstancias (el primer día de cada mes, después de finalizada la jornada de trabajo) para así enviar información confidencial a los atacantes, o en su defecto, permite que los mismos atacantes tengan acceso al computador donde esta alojada desde afuera.

1.6.5. ‘Ransomware’

El Instituto Nacional de Ciberseguridad de España, INCIBE, define el “Ransomware” como “es una extorsión que se realiza a través de un “malware”

que se introduce en los equipos de las empresas: ordenadores, portátiles y dispositivos móviles" (s.f.). El código malicioso secuestra la información de la empresa, utilizando generalmente técnicas de cifrado, impidiendo el acceso a la misma, con el objetivo de solicitar un rescate a cambio de su liberación. De ahí su nombre – "ransom" (rescate) y "ware" proveniente de la palabra "software". Esta técnica, utilizada cada vez más por los cibercriminales, causa "pérdidas temporales o permanentes de información, interrumpe la actividad normal, ocasiona pérdidas económicas y daños de reputación" (INCIBE, s.f.).

Otro factor que facilita esta técnica es el uso creciente de las criptomonedas, las cuales son "monedas virtuales que permiten el pago anónimo entre particulares" (INCIBE, 2017, pág. 5). El anonimato es posible gracias a los servicios de "mixing" o "tumbling" de las criptomonedas, lo cual hace posible el acceso desde la red anónima "Tor", logrando que sea posible mezclar fondos de distintas carteras virtuales, resultando en una especie de lavado de activos que dificulta que se pueda seguir el rastro de las transacciones (INCIBE, 2017). Lo anterior facilita que los cibercriminales puedan extorsionar a sus víctimas sin que los organismos del orden, como la policía, pueda dar con su paradero.

1.6.6. ‘Spyware’

Es un “malware” espía que recopila y envía información confidencial a un servidor externo. En general, los atacantes buscan datos que puedan utilizar más tarde, normalmente con fines econó-micos. La palabra spyware viene de spy, ‘espía’, y la aféresis ware de software (Arroyo Guardeño et al., 2020).

Expertos de la compañía de “software” AVG, creadores de aplicaciones contra virus, afirman que la capacidad de este tipo de ‘malware’ de “...evitar su detección mientras monitoriza su información más privada hacen de él uno de los más peligrosos” (Lemonnier & Latto, 2020). Según los autores, los virus normales pueden dañar los dispositivos, como celulares o computadores, así como sus datos, sin embargo, el ‘spyware’ va más allá, ya que puede robar la identidad de la persona y sus bienes reales.

La manera en que el ‘spyware’ roba los datos es a través de técnicas como registrar cada teclado presionada por un usuario para saber lo que escribió, tomar el control del equipo haciendo cambios en la configuración, dejando al usuario incapaz de hacer algo al respecto, y hasta rastrear su actividad en internet (ver cuales sitios visito).

1.6.7. ‘Phishing’

“Los programas de ‘phishing’… sustituyen las direcciones de páginas de Internet legítimas con otras parecidas pero controladas por los atacantes” (Arroyo Guardeño et al., 2020, pág. 23), así el usuario termina introduciendo datos confidenciales en la página falsa pensando que se trata de la original.

El Superintendente en Jefe de la Policía Metropolitana de Londres, Michael Gallagher, en su libro “El Pequeño Libro de los Ciber Engaños 2.0” (2019) afirma que los atacantes frecuentemente se hacen pasar por una persona o institución, como un banco, para engañar a la víctima y que esta le provea información confidencial. Es decir, una persona puede mandar un correo con código malicioso, y hacerse pasar por su banco, pidiendo a la víctima que abra el correo o introduzca información, como su número de cuenta, para así desencadenar el código malicioso, el cual le robara la información (Gallagher, 2019).

1.6.8. ‘Smishing’

Una variante de ‘Phising’ llevada a cabo a través de mensajes SMS de teléfonos móviles (Gallagher, 2019). Con esta técnica, el atacante disfraza su número móvil original y lo hace ver como el numero de una persona allegada a la víctima. El objetivo es el mismo, engañar a la persona para que provea

información sensible la cual será robada por el atacante. Ejemplo, podría recibir el mensaje de algún pariente diciéndole que se accidento y que necesita que le haga una transferencia bancaria urgente para poder pagar los servicios médicos, y que la única forma de realizarla es mediante un enlace recibido en el mensaje SMS.

1.6.9. ‘Spear Phishing’

Según Gallagher (2019) es una más directa del ‘Phising’, donde el atacante manda un correo haciéndose pasar por un allegado, como un colega del trabajo, utilizando información personal de la víctima como sitio de trabajo, universidad a la que fue, entre otras. Mientras que con el ‘phishing’, el correo parecerá provenir de una institución conocida, como un banco, en el ‘spear phising’ el correo parecerá provenir de una persona allegada.

1.6.10. Inyección de SQL

Para García-Moran, Fernández Hansen y Martínez Sánchez (2014) la inyección de SQL es “…una técnica que consiste en añadir código malicioso a las sentencias SQL originales ejecutadas por un programa en el motor de base de datos”. (pág. 276). Dicha alteración se realiza con el objeto de obtener acceso no autorizado a la información almacenada.

Estas vulnerabilidades permiten que los atacantes puedan acceder a los datos de la aplicación, según los expertos en seguridad ‘web’ de la compañía Mozilla, creadores del Navegador Firefox. Una vez tienen acceso, puedan modificar o borrar información, crear nuevas identidades con derechos de administración en la base de datos o incluso acceder a todos los datos en el servidor para destruirlos (Mozilla, 2020).

Las inyecciones de SQL usualmente se logran al introducir instrucciones de código SQL en un campo (ejemplo: contraseña) de una página web, donde los desarrolladores no tomaron medidas para validar los datos de los campos, o en aquellas aplicaciones donde las instrucciones de SQL para hacer consultas a la base de datos están escritas directamente en el código. Ejemplo:

```
statement = "SELECT * FROM users WHERE name = '" + userName + "';"
```

Mozilla (2020). Seguridad de sitios Web. [Figura].

Con el código anterior, si el usuario introduce su nombre, la aplicación funciona como se espera. Sin embargo, si ‘hacker’ cambia completamente la sentencia a la nueva sentencia de abajo, al especificar para UserName. Ejemplo:

```
SELECT * FROM users WHERE name = 'a';DROP TABLE users; SELECT * FROM userinfo WHERE 't' = 't';
```

Mozilla (2020). Seguridad de sitios Web. [Figura].

La nueva sentencia crea un comando SQL válido que borra la tabla de usuarios y selecciona todos los datos de la tabla 'userinfo', revelando la información de todos los usuarios. Esto porque la primera parte del texto inyectado (a';) completa la sentencia original - ' es el símbolo para indicar una cadena literal en SQL (Mozilla, 2020).

1.6.11. Ataque de denegación de servicio

El ataque de denegación de servicio (DDoS, por sus siglas en inglés), es aquel en el cual el acceso al mismo es comprometido para los usuarios legítimos autorizados para dicho servicio. Cualquier ataque de los mencionados en los puntos anteriores puede devenir en un DoS.

1.6.12. 'Cross Site Scripting'

es un término anglosajón que se utiliza para describir ataques que permiten al atacante "inyectar scripts de lado cliente, *a través* del sitio web, hasta los exploradores de otros usuarios" (Mozilla, 2020). El código inyectado surge del servidor del sitio al explorador, se supone de confianza, por ende, puede hacer

tareas como acceder a las cookies de autorización. Una vez que el atacante tiene las cookies podrá iniciar sesión en el sitio como si fuera un usuario autorizado. Si el sitio es un banco, esto implicaría que el atacante tenga acceso a los detalles de su tarjeta de crédito, o contraseñas.

Hay dos técnicas para conseguir que el sitio devuelva scripts inyectados al explorador: explotación de vulnerabilidades XSS **reflejadas y persistentes**. Según los expertos de la empresa Mozilla (2020), una vulnerabilidad XSS reflejada ocurre cuando el contenido referente al usuario que pasa por un servidor se devuelve inmediatamente y sin modificar. Consideremos una función de búsqueda en un sitio donde los términos de búsqueda están codificados como parámetros URL y estos términos se presentan junto con los resultados. Un atacante puede construir un enlace de búsqueda que contenga un script malicioso como parámetro. Ejemplo:

```
http://mysite.com?q=beer<script%20src="http://evilsite.com/tricky.js"></script>
```

Luego, lo envía como un enlace por correo electrónico a otro usuario. Si el destinatario hace clic en él, se ejecutará el script cuando se muestren en pantalla los resultados de la búsqueda. Esto hace posible que el atacante acceda a la información que necesita (Mozilla, 2020).

En cambio, una vulnerabilidad XSS persistente es una donde el script malicioso se almacena en el sitio web para más presentarse en pantalla para que otros usuarios lo ejecuten involuntariamente. Considere un foro de discusión donde los usuarios pueden comentar, este foro podría almacenar un script malicioso de un atacante y al mostrarse los comentarios ejecutará el script, lo cual enviará al atacante la información requerida para acceder a la cuenta del usuario.

1.6.13. 'Cross Site Request Forgering'

Los ataques de CSRF se llevan a cabo ejecutando acciones donde se usan las credenciales de otro usuario sin el conocimiento o consentimiento de éste.

Considere un 'hacker' que conoce un sitio el cual permite a los usuarios, que han iniciado sesión, enviar dinero a una cuenta específica usando una petición HTTP POST que incluye el nombre de la cuenta y una cantidad de dinero. El 'hacker' puede construir un formulario que incluya detalles de su banco y una cantidad de dinero X, como campos ocultos. Luego lo envía por correo electrónico a otros usuarios del sitio, con un botón de **Enviar** camuflado como enlace a un sitio llamado "hazte rico rápidamente"). Si la victima hace clic en el botón, se envía al servidor una petición HTTP POST que contiene los detalles de la transacción y todas las cookies de la víctima que el explorador asocia con el sitio. El servidor

comprobará las cookies, y las usará para determinar si el usuario ha iniciado sesión o no y si tiene permiso para hacer la transacción (Mozilla, 2020).

1.6.14. ‘Clickjacking’

En un ataque donde un ‘hacker’ secuestra los clics del ratón el computador que vayan dirigidos a un sitio y los redirige a una página escondida. Esta técnica se usará típicamente para presentar un sitio malicioso como un sitio bancario legítimo, para así capturar las credenciales. Alternativamente, se usa para conseguir que el usuario haga clic sobre un botón invisible en un sitio (Mozilla, 2020).

1.6.15. Ingeniería Social

La Agencia de Seguridad para la Ciberseguridad & Infraestructura de los Estados Unidos (CISA), establece que la ingeniería social es una técnica donde una persona, denominada ‘hacker’, utiliza habilidades sociales para obtener información de una víctima, con la finalidad de utilizar dicha información obtenida en un futuro para vulnerar un sistema informático (CISA, 2009). Al hacer preguntas sobre su sitio de trabajo, por ejemplo, el ‘hacker’ puede obtener alguna información sobre qué tipo de sistemas utiliza la empresa, lo que le permita en un futuro entrar al sistema de la para robar información.

1.7. LA CALIDAD DEL SOFTWARE

Para poder velar por el cumplimiento de la calidad, los profesionales del “software” hacen uso de las metodologías y técnicas existentes de aseguramiento de calidad ya aplicadas en si al desarrollo de sus aplicaciones, conociéndose como Aseguramiento de la Calidad de “Software”. Dicha metodología es definida por el Comité para los Sistemas de Seguridad Nacional de los Sistemas (CNSS, por sus siglas en inglés) como una rama de la Ingeniería de “Software” que incorpora actividades que

> “…aseguran la implementación de las aplicaciones con un nivel de confianza en que este funciona según lo previsto y está libre de vulnerabilidades, ya sea intencional o involuntariamente diseñado o insertado como parte del software, durante todo el ciclo de vida” (Citado en Woody, Ellison, & Nichols, 2014, pág. 3).

Es una actividad sombrilla que se aplica en todo el proceso del software. El aseguramiento de la calidad del software incluye lo siguiente: 1) un proceso de aseguramiento general, 2) tareas específicas de aseguramiento y control de la calidad (incluidas revisiones técnicas y una estrategia de pruebas relacionadas entre sí), 3) prácticas eficaces de ingeniería de software (métodos y herramientas), 4) control de todos los productos del trabajo de software y de los cambios que sufren (véase el capítulo 22), 5) un procedimiento para garantizar el

cumplimiento de los estándares del desarrollo de software (cuando sea aplicable) y 6) mecanismos de medición y reporte (Pressman, 2010, pág. 369).

En la calidad de software se enfatizan tres puntos importantes, según Pressman (2010). Primero, la infraestructura que da apoyo a cualquier esfuerzo de elaboración de un producto de software de alta calidad. Los aspectos de administración del proceso generan las verificaciones y equilibrios que ayudan a evitar que el proyecto caiga en el caos, contribuyente clave de la mala calidad. Las prácticas de ingeniería de software permiten al desarrollador analizar el problema y diseñar una solución sólida, ambas actividades críticas de la construcción de software de alta calidad.

1.8. LA SEGURIDAD DE SOFTWARE

La seguridad del software es una actividad del aseguramiento del software que se centra en la identificación y evaluación de los peligros potenciales, descritos en la sección anterior, los cuales podrían afectarlo negativamente a una empresa, ocasionando que falle todos sus sistemas. Si los peligros se identifican al principio del proceso de desarrollo de software, los riesgos se pueden eliminar y/o controlar (Pressman, 2010, pág. 378).

Como parte de la seguridad del software, se lleva a cabo un proceso de modelado y análisis. Inicialmente se identifican los peligros y se clasifican según su riesgo. Una vez identificados y analizados los peligros, pueden especificarse requerimientos relacionados con la seguridad para el software. Es decir, la especificación contendría una lista de eventos indeseables y las respuestas deseadas del sistema ante ellos. Después se indicaría el papel del software en la administración indeseable de los mismos.

Aunque la confiabilidad y la seguridad del software están muy relacionadas, es importante entender la sutil diferencia entre ellas. La primera utiliza técnicas de análisis estadístico para determinar la probabilidad de que ocurra una falla del software. Sin embargo, la ocurrencia de una falla no necesariamente da como resultado un peligro o riesgo. La seguridad del software examina las formas en las que las fallas generan condiciones que llevan a un peligro. Es decir, las fallas no se consideran en el vacío, sino que se evalúan en el contexto de la totalidad del sistema basado en computadora y de su ambiente.

“La seguridad de la información abarca todo aquello que tiene que ver con la protección de la información, ya sea almacenada o transmitida” (Arroyo Guardeño et al., 2020, pág. 15). Recordemos, que la información es un activo crítico para cualquier organización, pues es esta la que recoge todos los datos

que van desde la situación económica de la empresa hasta sus operaciones y como se ejecutan.

1.8.1. Ciberseguridad

Se puede catalogar como "… el conjunto de técnicas, procedimientos y protocolos encaminados a la protección de la información vinculada a los usuarios de las ciber tecnologías" (Arroyo Guardeño et al., 2020, pág. 12). La diferencia con la seguridad de la información es que la ciberseguridad no solo abarca la transmisión de la información, sino además los activos para transmitir esa información. Esto demanda que se custodie la información y todos los elementos precisos para su correcta gestión. La ciberseguridad tiene como objetivo proteger todo tipo de activo o recurso de valor para una persona, empresa u organización (Arroyo Guardeño et al., 2020).

1.8.2. Objetivos de la ciberseguridad

Según Arroyo et al. (2020), la seguridad tiene seis objetivos principales: **confiabilidad**, la **integridad** y la **disponibilidad** – estos tres conocidos por sus siglas "**CID**". El ultimo trio se conocen como las tres "***Au***": **autorización**,

auditabilidad y **autenticación**. A continuación, un resumen de en qué consisten estos objetivos para así ver su importancia:

a) **Confiabilidad:** "…garantiza la protección de la información de modo que sea secreta para quienes no tienen derecho a acceder a la misma" (Arroyo Guardeño et al., 2020, pág. 16). Por ejemplo, el amigo de una persona que trabaje en el área de caja de una institución bancaria no debería poder acceder, o conocer, información de los cuadres diarios de esa organización.
b) **Integridad**: "Asegura la autenticidad de los datos almacenados, de modo que no puedan ser modificados, manipulados ni alterados por terceras partes sin permiso para ello" (Arroyo Guardeño et al, pág. 16).
c) **Disponibilidad:** La disponibilidad de los datos almacenados en un sistema informático obliga a que su acceso sea posible en cualquier momento que sea solicitado por cualquier parte que esté autorizada a ello. (Arroyo Guardeño et al., 2020, pág. 16).
d) **Autenticación**: Según Arroyo Guardeño et al. (2020), "es una propiedad de la seguridad de la información que permite confirmar la identidad de un usuario y, eventualmente, la de sus dispositivos" (pág. 16). Los autores continúan estableciendo que existen tres modos de utilizar la autenticación: utilizando el conocimiento (ejemplo: una contraseña), un objeto físico (una

tarjeta de crédito) o algo propio de nuestro cuerpo (como la retina o la huella digital).

e) **Autorización:** "…es el proceso por el que se controla el acceso de un usuario a determinado servicio para realizar ciertas tareas" (Arroyo Guardeño et al., 2020, pág. 16). Para ello, se pueden usar políticas de autorización para así señalar lo que la entidad autorizada tiene permitido hacer.

f) **Auditabilidad:** es la técnica que permite registrar y monitorizar como y para qué son usados los distintos recursos (Arroyo Guardeño et al., 2020). Esta actividad es fundamental para comprobar que los sistemas de información están funcionando de manera adecuada, permitiendo identificar a tiempo las fallas de seguridad y las responsabilidades.

1.8.3. Organismos Internacionales de Ciberseguridad

Existen varios organismos internacionales que trabajan en coordinación con empresas e instituciones gubernamentales para fortalecer la seguridad tanto de las aplicaciones de dichas instituciones, así como lograr que el Internet sea un lugar más seguro para las personas, especialmente para los niños y grupos vulnerables. A continuación, presentamos algunos de los más importantes.

INCIBE

El Instituto Nacional de Ciberseguridad de España (INCIBE) es:

> *"una sociedad dependiente del Ministerio de Asuntos Económicos y Transformación Digital a través de la Secretaría de Estado de Digitalización e Inteligencia Artificial y consolidada como entidad de referencia para el desarrollo de la ciberseguridad y de la confianza digital de ciudadanos, red académica y de investigación, profesionales, empresas y especialmente para sectores estratégicos"* (Que es INCIBE: INCIBE, s.f.).

Surge el 28 de octubre del 2014, sustituyendo al Instituto Nacional de Tecnologías de la Comunicación (INTECO), gracias al acuerdo de la Junta General, el 27 de octubre del 2014. Su trabajo tiene como finalidad ser un instrumento del gobierno para así prestar servicios que contribuyan a la ciberseguridad nacional e internacional. Operan basándose en estudios, prestación de servicios y coordinación con los agentes con competencias en materia de ciberseguridad.

Su misión es "reforzar la ciberseguridad, la confianza y la protección de la información y privacidad en los servicios de la Sociedad de la Información" (Que es INCIBE: INCIBE, s.f.) para así aportar valor al sector público y privado en

materia de ciberseguridad. INCIBE forma parte de foros y grupos de trabajo en ciberseguridad del plano nacional e internacional, colaborando con actores estratégicos como Antiphishing Working Group (APWG), ECSO (European Cybersecurity Organization), European Network and Information Security Agency (ENISA), entre otros. (Con Quien Trabajamos, s.f.)

NIST

El Instituto Nacional de Estándares y Tecnología (NIST) fue fundado en 1901 y ahora es parte del Departamento de Comercio de los Estados Unidos (About NIST, s.f.). El Congreso de Estados Unidos lo estableció para hacer frente al desafío a la competitividad industrial de aquel entonces. Entre sus funciones están crear estándares para todo tipo de tecnologías desde red eléctrica inteligente, registros electrónicos de salud, relojes atómicos, nanomateriales avanzados y chips informáticos. (About NIST, s.f.)

Su misión es promover la innovación y la competitividad industrial de los Estados Unidos mediante el avance de la ciencia, los estándares y la tecnología de medición (NIST Mission, Vision, Core Competencies, and Core Values, s.f.) para así fomentar economía y mejorar la calidad de vida pública. Su visión es ser el líder mundial en la creación de soluciones para medición críticas, así como

promover estándares equitativos, estimular la innovación y fomentar la competitividad.

Entre sus logros podemos encontrar una alianza público-privada para la investigación de semiconductores (Public-Private Partnership for Semiconductor Research, s.f.), para así ver cuáles son los mayores retos en esa rama. Además, su trabajo para un “framework” de ciberseguridad, en 2014, para mitigar riesgos y apoyar a las empresas (Cibersecurity Framework, s.f.), el cual fue reconocido por el congreso de Estados Unidos. El “framework” integra estándares de la industria de TIC y utiliza un lenguaje universal para orientar a las organizaciones en el manejo de la ciberseguridad – ejemplo: establece parámetros de como recuperarse de incidentes de ciberseguridad. Entre las compañías que han adoptado el “framework” destacan las compañías tecnológica Microsoft e Intel, el banco JP Morgan Chase, Nipón Telegraph y la Junta de Energía de Ontario, Canadá.

ENISA

La Agencia Europea de Seguridad de la Red y la Información (ENISA) es una agencia de la Unión Europea (UE) dedicada a prevenir y abordar los problemas de seguridad de la red y la seguridad de la información. También ayuda

a la Comisión Europea a actualizar y desarrollar la legislación de la Agencia de la Comunidad Europea en el campo de la Seguridad de Redes e Información.

Actúa como una «Agencia de la Comunidad Europea» y trabaja con las instituciones de la UE y los Estados miembros para desarrollar una cultura de redes y seguridad de la información para ayudar a los ciudadanos, consumidores, empresas y organizaciones del sector público de la Unión Europea a abordar y prevenir la red de seguridad y problemas de seguridad de la información.

ENISA es supervisada por un consejo de administración compuesto por representantes de los Estados miembros de la UE, la Comisión de la UE y otras partes interesadas.

OWASP

La fundación Open Web Application Security Project (OWASP) es un órgano sin ánimo de lucro con miras a promover la seguridad del software, particularmente las aplicaciones web en particular. Se fundo en el año 2001, y tiene como misión producir código lo suficientemente seguro en el que las empresas puedan para confiar.

OWASP depende para su mantenimiento de las donaciones y las cuotas de los socios, particulares y empresas. La revisión de los controles, definidos por esta metodología, permite al equipo de auditores garantizar que una revisión de la plataforma se realiza de forma adecuada, garantizando que todos los vectores de ataque han sido analizados y que los fallos de seguridad han sido detectados. Este proceso ayuda a mejorar la seguridad y la protección de los sistemas informáticos de nuestros clientes.

Agencia de Seguridad para la Ciberseguridad & la Infraestructura de Estados Unidos (CISA)

Es una agencia que pertenece al Departamento de Seguridad Nacional de los Estados Unidos y funciona como un consejero para el riesgo de ciberseguridad en dicho país (CISA, s.f.), proveyendo herramientas, servicios de respuestas y verificación de riesgo para proteger los portales de las entidades del gobierno americano. Además, esta aliada al sector privado, con un conjunto de programas orientados a analizar, entender y mitigar los riesgos y vulnerabilidades de seguridad en Internet.

1.9. TENDENCIAS DEL ASEGURAMIENTO DE LA CALIDAD DE SOFTWARE EN EL CAMPO DE LA SEGURIDAD

1.2.1. Pruebas con inteligencia artificial

Para Konitz (2019), la inteligencia artificial (IA) es muy útil en múltiples dominios e industrias. Según el autor, la IA podría ayudar a realizar pruebas desde la perspectiva del usuario final, ya que permite identificar algunos problemas complejos para los seres humanos. La IA puede comparar una imagen con otra de referencia para detectar las diferencias entre ellas, como la diferencia entre la atenuación de los colore de fondo. Además, el autor indica que un algoritmo de IA puede determinar fácilmente si la textura se representa correctamente, pixel a pixel, algo bastante difícil para el ojo humano, por no decir imposible. (Konitz, 2019).

La aceptación de la inteligencia artificial (IA) y aprendizaje automático ha aumentado en el área de aseguramiento de la calidad. En una encuesta realizada, el 88% de los encuestados dijo que la IA era ahora el área de crecimiento más fuerte de sus actividades de prueba, según Capgemini (2020), mientras que el 86% considera que la IA es clave para seleccionar nuevas soluciones de control de calidad. El mayor desafío en la aplicación de la IA y el ML en el aseguramiento de la calidad son las brechas de habilidades, mencionadas por el 34% de los encuestados (dos puntos porcentuales menos que el año pasado).

Se espera que el uso de la inteligencia artificial (IA) siga creciendo debido al gran número de aplicaciones que son cada vez más comunes en el mundo globalizado (Mikhalchuk, 2020). El autor asegura que se espera que la inversión en el área de inteligencia artificial supere los 200,000 millones de dólares sólo en América del Norte para el año 2025.

1.2.2. Pruebas automatizadas

Una encuesta mencionada por Mikhalchuk (2020), revela que la automatización de pruebas es un enfoque que debe ser prioritario. En dicha encuesta, El 35% de los profesionales entrevistados informó que sus organizaciones tienen la mitad de las pruebas automatizadas, mientras que el 12% dijo que ya han automatizado completamente sus pruebas.

Mikhalchuk (2020) considera que hoy en dia el proceso de mejora continua no sería posible sin pruebas las automatizadas. Según el autor, con el enfoque ágil, los defectos y errores se detectan más rápido, dando como resultado un aseguramiento de calidad sea más eficiente. Además, agrega que la externalización de pruebas de software es una forma de evitar prácticas de prueba obsoletas, permitiendo a los equipos enfocarse en asignar recursos a escenarios de prueba complejos que impulsan el valor del producto.

Sin embargo, dice el autor que las llamadas pruebas manuales siguen siendo particularmente relevantes e importantes. Por ello, recomienda que los equipos de control de calidad utilicen una combinación de pruebas manuales y automatizadas. La mezcla, dice Mikhalchuk, da como resultado una estrategia de prueba completa.

1.2.3. Pruebas enfocadas a la seguridad

Las pruebas de seguridad son una prioridad para las empresas que se preocupan por el flujo de datos, en un mundo que esta interconectado y globalizado. Mikhalchuk (2020) propone que la introducción temprana de pruebas de seguridad resulta clave para los equipos de aseguramiento de la calidad.

El autor cita un estudio llamado "Better Security And Business Outcomes With Security Performance Management" realizado por la organización BitSight, el cual mostró que el 82% por ciento de los entrevistados estaban de acuerdo en que la seguridad es muy importante para sus clientes y socios, siendo una pieza clave en como toman decisiones. Lo que para el autor habla de cómo la ciberseguridad y ha cambiado las listas de prioridades en la gestión de calidad, diciendo que la seguridad pronto será un estándar que no podrá ignorarse. Desde su óptica, Mikhalchuk dice que las pruebas de penetración ayudan a las

empresas a ahorrar bastante, ya que proporcionan una comprensión de los puntos débiles de la aplicación del negocio antes de que los “malos lo hagan” (Mikhalchuk, 2020). Con las pruebas de ciberseguridad, hay una semi garantía de que, si se produce un contratiempo, la recuperación no será tan costoso y dañino como si no se estuviera preparado. Las pruebas de penetración regulares contribuyen a la buena reputación de las organizaciones, ayudándoles a ganarse una mayor confianza entre las organizaciones y sus clientes y sus socios (Mikhalchuk, 2020).

1.2.4. IoT y Big Data

Con la industria tecnológica moviéndose a una velocidad vertiginosa, la cantidad de información sigue creciendo, haciendo de las pruebas de seguridad una prioridad para las empresas que se preocupan por el flujo de datos y excluyendo cualquier fuga, error de código y agujeros. En conjunto, la introducción temprana de pruebas de seguridad resulta ser rentable en casi todos los casos.

En particular, un estudio llamado "Better Security And Business Outcomes With Security Performance Management" de BitSight mostró que el 82 por ciento de las partes interesadas estaban de acuerdo en que la forma en que sus clientes y socios perciben la seguridad es cada vez más importante para la forma en que sus organizaciones toman decisiones. Esto por sí solo habla mucho sobre cómo

la ciberseguridad y el cumplimiento han cambiado las listas de prioridades más altas. No hay una sola razón para ignorar esta tendencia que pronto será estándar, y aquí hay algunas razones por las que:

Las pruebas de penetración ahorran un centavo bastante: las violaciones de datos agravan progresivamente la posición ya vulnerable que las empresas encuentran en medio de la pandemia de 2020.

1.10. DIAGNOSTICO ACTUAL EN LA EMPRESA WebArtRD

1.3.1. Introducción

La investigación se realiza tomando como muestra a las empresas de desarrollo web en República Dominicana, delimitando a la empresa WebArtRD. Debido al tipo de trabajo que realiza la empresa, las vulnerabilidades detectadas por los mismos empleados, y las exigencias del cliente, creemos que el conocimiento que será levantado en la investigación puede ser de ayuda para que la empresa mejore su proceso de calidad orientado a la seguridad.

En la actualidad la empresa no cuenta con una metodología formal para realizar pruebas de seguridad, una omisión que significa el desconocimiento de

muchas de las vulnerabilidades de sus aplicaciones, exponiéndolas a potenciales riesgos que van de algo tan sencillo como que un empleado/cliente descargue un virus, a ser atacado por “hackers” para robar información crítica, o que el sistema se vea afectado por un ataque de denegación de servicio (DoS, por sus siglas en inglés) y los clientes no puedan acceden a este, lo que resultaría en perdida de dinero. La presente investigación plantea solucionar esta problemática.

1.3.2. Historia

La compañía dominicana WebArtRD se dedicada al desarrollo de aplicaciones y páginas “web” para otras empresas, locales y extranjeras, contando con una trayectoria de 12 años en el mercado.

Es sus palabras es “…una agencia enfocada en abastecer conocimientos, herramientas y desarrollo tecnológico para empresas que necesitan evolucionar en el mundo digital. Ofrecemos un conjunto de servicios líderes en el mercado, desde proyecciones de marca hasta soluciones estratégicas orientadas a la web y comercio electrónico. Nuestra trayectoria de más de 10 años en el mercado representa nuestra pasión por brindar servicios de calidad alineados con los objetivos estratégicos de nuestros clientes.”

Actualmente, la empresa cuenta con dos analistas de calidad quienes hacen pruebas en las aplicaciones desarrollas. enfocadas solo a la funcionalidad de la interfaz y a la lógica del negocio; sin embargo, su proceso de calidad **no** incluye pruebas dirigidas a verificar la calidad de la **seguridad**, o ciberseguridad. La omisión significa que la empresa desconoce muchas de las vulnerabilidades de sus aplicaciones, exponiéndolas a potenciales riesgos que van de algo tan sencillo como que un empleado/cliente descargue un virus, a ser atacado por "hackers" para robar información crítica, o que el sistema se vea afectado por un ataque de denegación de servicio (DoS, por sus siglas en inglés) y los clientes no puedan acceden a este, lo que resultaría en perdida de dinero.

De hecho, la empresa WebArtRD ya se enfrentó a un caso donde se vio vulnerada la seguridad de una de sus aplicaciones. Aunque se debe resaltar que **no fue un ataque llevado a cabo por "hackers"**, como los dos casos anteriores, pero si una persona tuvo acceso por error a información confidencial de uno de los clientes de la empresa, gracias a una vulnerabilidad de seguridad desconocida, resultando en **insatisfacción del cliente con la aplicación, y con la misma empresa**.

Para mitigar riesgos potenciales como ataques de "hackers", y atendiendo tanto a las demandas de su cliente, para que presten atención al factor seguridad (para que no se vuelva a repetir la situación donde alguien accede a datos

confidenciales), como de un mundo globalizado e interconectado por la Internet, WebArtRD se ve en la necesidad de implementar técnicas que le permitan identificar las áreas de la aplicación que **tienen defectos y vulnerabilidades de seguridad**, lo que permitiría corregirlas. En ese sentido, el aseguramiento de la calidad orientado a la seguridad plantea métodos y técnicas que la empresa pudiera implementar, ya que está en su interés verificar que la seguridad de sus aplicaciones está en conformidad con los estándares internacionales y con los requerimientos de sus clientes, dándole satisfacción al cliente, al mismo tiempo que se mejoran los procesos internos de desarrollo.

1.3.3. Estructura Organizacional

Figura – Estructura Organizacional WebArtRD. autoría: propia.

Breve descripción de los puestos de trabajo:

Puesto	Descripción
Presidente y Fundador	Líder de la empresa
Desarrollador Web en Jefe	Encargado de liderar y administrar al equipo de desarrollo.
Directora administrativa	Encargada de las tareas administrativas como contratación de personal, pago de nóminas, dirigir el área de publicidad, entre otras.
Gerente de Proyectos	Encargado de realizar el levantamiento los requerimientos al reunirse con el cliente
Analista Programador	Encargado de diseñar la aplicación a desarrollar, basándose en los requerimientos.
Desarrollador 'Full-Stack'	Encargado de desarrollar el 'backend', el 'front-end' y los 'web services' de la aplicación 'web'.
Ingeniero QA de 'Software'	Encargado de hacer las pruebas de calidad de 'software'.
Publicista	Encargado de la publicidad y la imagen pública de la empresa.
Diseñador Gráfico	Encargado de diseñar la interfaz de usuario de las aplicaciones.

1.3.4. Misión, Valores y visión

Misión: ofrecer soluciones web apegados a la innovación y creatividad que le permitan a los clientes un desarrollo exitoso de sus productos. Apoyar a la industria nacional ayudándolas a posicionarse en el mercado global, al mismo tiempo que le brindamos un servicio de calidad a nuestros clientes internacionales.

Nuestras soluciones web se basan en la creatividad e innovación, dando valor agregado que impulsen el negocio web de nuestros clientes.

Visión: Ser la compañía líder a nivel nacional e internacional en el diseño y desarrollo de aplicaciones web y ser reconocida por la innovación, simpleza y generación de valor de sus soluciones de marketing digital, con una alta productividad y calidad humana de su equipo.

Valores: Espíritu innovador, creatividad, valores éticos, esfuerzo, confianza.

1.3.5. Análisis de la Metodología de Desarrollo

La empresa, utiliza la metodología de desarrollo ágil SCRUM, un proceso en el que se aplican un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, para obtener resultados deseables un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos (Proyectos Agiles, s.f.). Scrum se basa en:

a) El desarrollo incremental de los requisitos del proyecto en bloques temporales cortos y fijos (iteraciones de un mes natural y hasta de dos semanas, si así se necesita).

b) La priorización de los requisitos por valor para el cliente y coste de desarrollo en cada iteración.
c) El control empírico del proyecto. Por un lado, al final de cada iteración se demuestra al cliente el resultado real obtenido, de manera que pueda tomar las decisiones necesarias en función de lo que observa y del contexto del proyecto en ese momento. Por otro lado, el equipo se sincroniza diariamente y realiza las adaptaciones necesarias.
d) La potenciación del equipo, que se compromete a entregar unos requisitos y para ello se le otorga la autoridad necesaria para organizar su trabajo.
e) La sistematización de la colaboración y la comunicación tanto entre el equipo y como con el cliente.
f) El 'timeboxing' de las actividades del proyecto, para ayudar a la toma de decisiones y conseguir resultados

Las actividades en su proceso de desarrollo de software son las siguientes (donde las actividades del 2 al 6 serán actividades cíclicas dadas la naturaleza de la metodología SCRUM):

1. Reunión con el 'stackeholder' y levantamiento de requisitos.
2. Creación/Actualización del backlog de producto.
3. Diseño y análisis del sistema.
4. Desarrollo del sistema.

5. Pruebas internas.
6. Revisión del producto
7. Certificación del sistema.
8. Entrega a producción

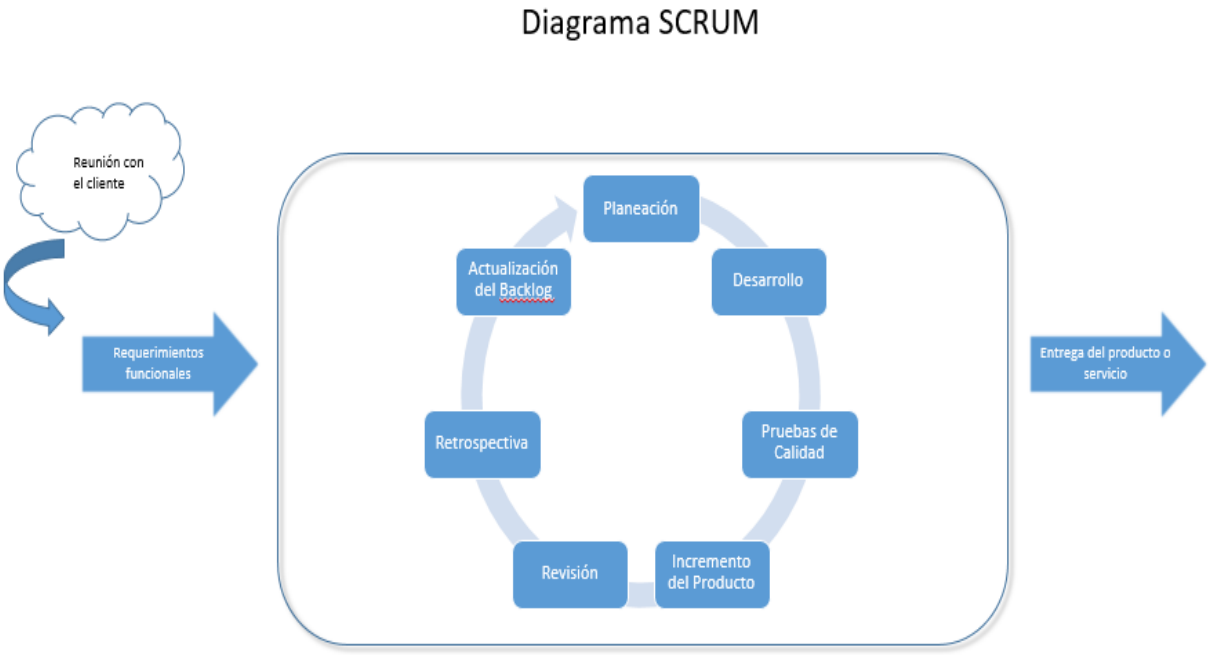


Figura – Proceso de desarrollo en la empresa WebArtRD. autoría: propia.

El proceso inicia cuando el cliente (o 'stackholder') se reúne con el Gerente de Proyectos para realizar una petición en la forma de requerimientos, los cuales serán plasmados en el documento de Análisis y Diseño, sin importar que sea una petición de modificación o el desarrollo de un nuevo sistema. En la empresa,

existe el documento de análisis y diseño done el Gerente de Proyectos establece los requerimientos para el diseño y desarrollo del ‘software’. Las solicitudes son analizadas por el Gerente de Proyectos y el analista de calidad quien traza el diseño técnico correspondiente. Dependiendo de las generalidades del producto, se pauta una reunión con el cliente para debatir y aclarar cualquier eventualidad en el diseño.

Cuando los requerimientos quedan claros, se prepara el backlog del proyecto que incluye desde el cronograma de actividades, el diseño, los tiempos y costo estimado del proyecto. Este documento se le envía al cliente. Al recibir aprobación, se empieza el proceso interno de desarrollo de la solución. En el transcurso del desarrollo el Ingeniero de Calidad de ‘Software’ va realizando las pruebas de funcionalidad interna de lugar, para validar que lo que está desarrollando cumple con los requisitos.

Al terminar el proceso de codificación de los requerimientos, y el proceso de certificación de calidad interno en los ambientes de prueba, el siguiente paso es entregar el sistema al cliente y ellos prueban y validan que está correcto. Si el cliente determina que hay una no conformidad, se devuelve para corregir si se encontraron defectos.

Finalmente, y si el sistema pasa la certificación del cliente, el producto es pasado a producción y luego empieza la fase de mantenimiento. Todos los documentos generados en cualquiera de estas fases son almacenados y archivados en la carpeta del proyecto al cual corresponde.

1.3.6. Análisis de la Metodología de aseguramiento de la calidad

La empresa cuenta con dos analistas de calidad quienes hacen pruebas enfocadas solo a la funcionalidad de la interfaz y a la lógica del negocio. La empresa utiliza principalmente un modelo de pruebas unitarias para complementar su metodología ágil de SCRUM. Estas pruebas consisten en aislar una parte del código y verificar que la lógica funciona de manera adecuada. Las pruebas se llevan a cabo principalmente de manera manual, aunque la empresa cuenta con algunas pruebas automatizadas.

Existen diversos frameworks para realizar este tipo de pruebas que dependerán del lenguaje. Los utilizados por la empresa son el framework **JEST**, que permite realizar las pruebas para el lenguaje 'Javascript' y **Laravel**, para el lenguaje PHP.

Para el reporte y seguimientos de defectos y la gestión de sus planes de prueba, la empresa utiliza la herramienta JIRA.

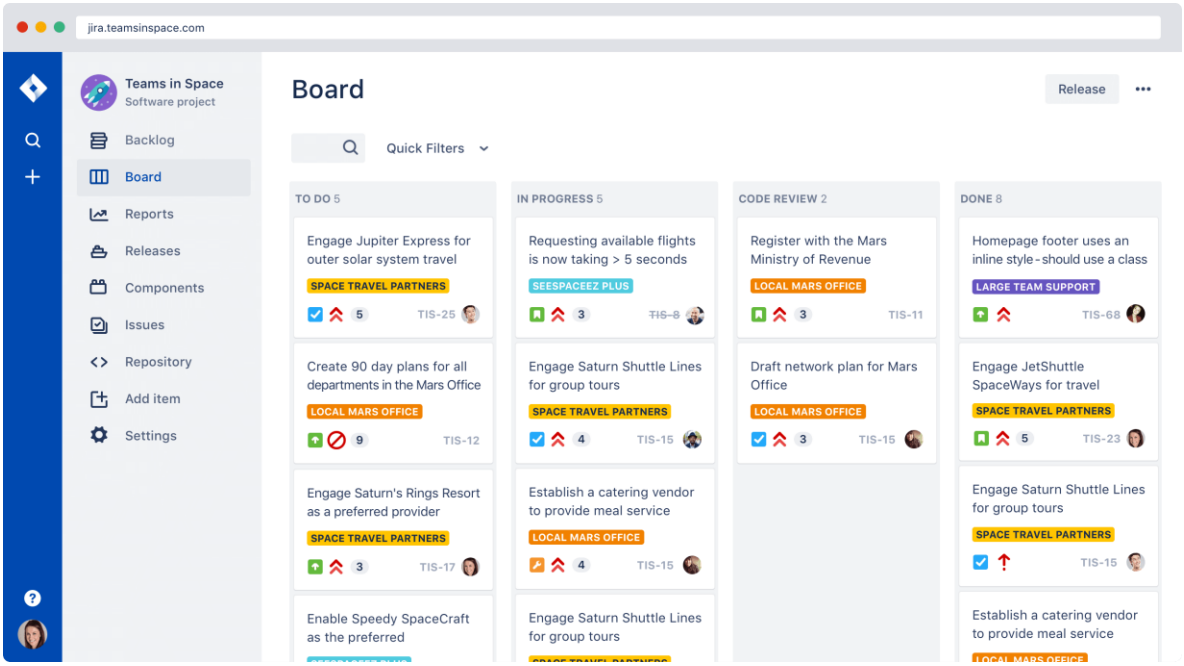


Figura – Pantalla de trabajo de la herramienta JIRA. Fuente: Jira | Software de seguimiento de proyectos e incidencias (atlassian.com)

.

Estructura de las pruebas unitarias

Las pruebas unitarias de la empresa siguen el patrón llamado AAA (por sus siglas en ingles), que significa ‘Arrange’, ‘Act’ y ‘Assert’. Esta estructura se puede definir de la siguiente manera:

1. **Arrange** (Organizar): se establece las condiciones iniciales para poder realizar la prueba y el resultado que se espera. Ejemplo: en esta fase se declaran las variables y se crear las instancias de los objetos.
2. **Act** (Accionar): se ejecuta el fragmento de código que se quiere probar.
3. **Assert** (Comprobar): se verifica que el resultado coincide con el resultado esperado

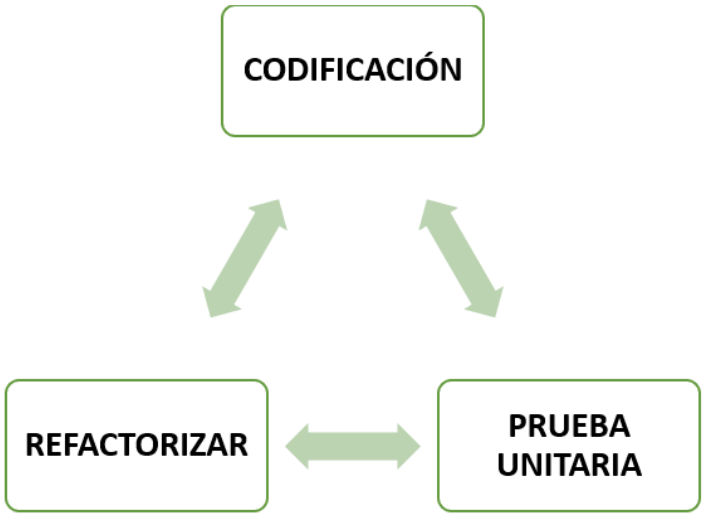


Figura – Proceso de Pruebas Unitarias WebArtRD. autoría: propia.

El proceso es simple: el programador codifica una funcionalidad del sistema. Luego, realiza el desarrollador o ingeniero de calidad realiza la prueba unitaria. Finalmente, refactoriza si se encontraron errores. El proceso es cíclico y se repite durante todo el ciclo de vida de desarrollo. En caso de ser el ingeniero

de calidad que realice las pruebas y encuentre un error, reporta al desarrollar un ticket con los datos del error para que este haga las correcciones de lugar. Para realizar las pruebas unitarias, utilizan el framework Laravel para las pruebas al código PHP, y JEST para JavaScript.

Estructura de las pruebas de funcionales de progresión

Estas pruebas son realizadas por los analistas de calidad y se basan en un plan de prueba que detalla los pasos a seguir para verificar que el resultado de la prueba está en conformidad con los requerimientos.

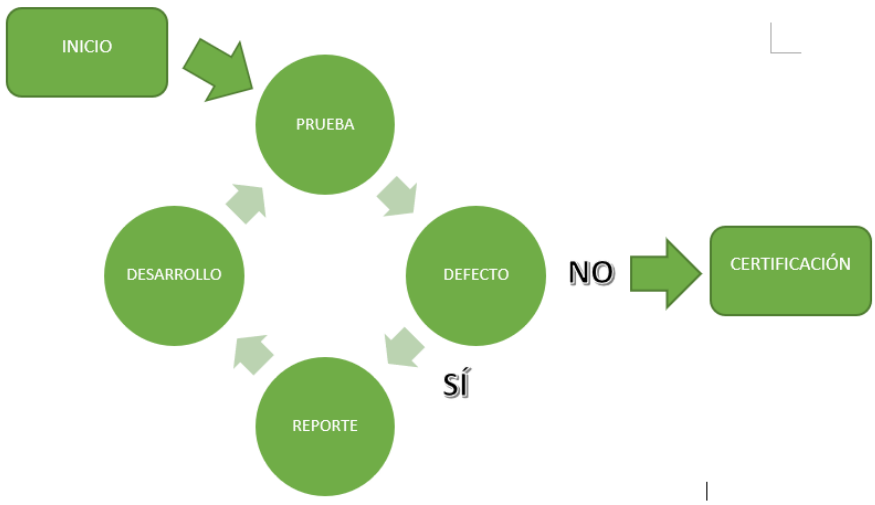


Figura – Proceso de Pruebas Funcionales WebArtRD. autoría: propia.

El proceso se detalla a continuación:

1. El analista de calidad elabora un plan de prueba en JIRA detallando los pasos a seguir y el resultado esperado.
2. El analista realiza la prueba
3. Si el resultado es el esperado, se certifica la funcionalidad
4. Si se encuentra una no conformidad, esta es documentada en JIRA y comunicada al desarrollador para que la arregle.
5. El analista vuelve a probar cuando el desarrollador arregla el defecto.
6. Si el defecto persiste, se devuelve al desarrollador
7. En caso de funcionar de acuerdo con lo esperado, se certifica.

Estructura de las pruebas funcionales de regresión

Las pruebas de regresión se utilizan para probar que funcionalidad previa no haya sido alterada por los nuevos cambios. El ciclo es básicamente el mismo de las pruebas de progresión. El analista de calidad utiliza un documento Excel con los pasos para realizar las pruebas. Si encuentra un defecto, abre un ticket para el desarrollador. Luego deberá probar cuando el desarrollador le comunique que está arreglado. Si no encontró defectos, el desarrollador puede certificar la prueba.

1.3.7. Problemática de seguridad

Como indicamos en la introducción, la empresa no cuenta con una metodología formal para realizar pruebas de seguridad debido a ello la empresa WebArtRD ya se enfrentó a un caso donde se vio vulnerada la seguridad de una de sus aplicaciones. Aunque se debe resaltar que **no fue un ataque llevado a cabo por "hackers"**, sí una persona tuvo acceso por error a información confidencial de uno de los clientes de la empresa, gracias a una vulnerabilidad de seguridad desconocida, resultando en **insatisfacción del cliente con la aplicación, y con la misma empresa**.

Para mitigar riesgos potenciales como ataques de "hackers", y atendiendo tanto a las demandas de su cliente, para que presten atención al factor seguridad (para que no se vuelva a repetir la situación donde alguien accede a datos confidenciales), como de un mundo globalizado e interconectado por la Internet, WebArtRD se ve en la necesidad de implementar técnicas que le permitan identificar las áreas de la aplicación que **tienen defectos y vulnerabilidades de seguridad**, lo que permitiría corregirlas. En ese sentido, la guía OWASP plantea métodos y técnicas que la empresa puede implementar, ya que está en su interés verificar que la seguridad de sus aplicaciones está en conformidad con los estándares internacionales y con los requerimientos de sus clientes, dándole satisfacción al cliente, al mismo tiempo que se mejoran los procesos internos de

desarrollo. En el siguiente capítulo exploraremos la propuesta para que la empresa pueda superar esta problemática.

CAPÍTULO II: PRUEBAS DE CALIDAD ENFOCADAS A LA SEGURIDAD DE APLICACIONES WEB

2.1. ANTECEDENTES DEL LAS PRUEBAS DE SEGURIDAD

Actualmente existen pocos **modelos** para pruebas de aplicaciones “web”. Uno de ellos es el creado por el proyecto *OWASP*, la **Guía de pruebas de OWASP**, la cual presenta una serie de buenas prácticas para realizar pruebas de seguridad en las aplicaciones web e identificar vulnerabilidades que pueden aparecer desde las primeras fases del ciclo de desarrollo (Diaz Diaz). Dicha metodología se basa en las distintas categorías de las vulnerabilidades de las aplicaciones “web”, las cuales los autores Urcuqui López, García Peña, Osorio Quintero & Navarro Canavid (2018) identifican como:

> *inyección*, *pérdida de autenticación*, *exposición de datos sensibles*, *pérdida del control de acceso*, *configuración de seguridad incorrecta*, *deserialización insegura*, *uso de componentes con vulnerabilidades conocidas* y *monitoreo insuficientes*.

También, el modelo incluye las pruebas de caja negra o **penetración,** que es la manera como la mayoría de los atacantes puede tener acceso a una aplicación (Diaz Diaz). En este capítulo, estaremos analizando el modelo OWASP más adelante en el subcapítulo ***El Modelo OWASP***.

2.2. TIPOS DE PRUEBAS DE SEGURIDAD WEB

2.2.1. Escáneres de vulnerabilidad

Los escáneres de vulnerabilidades evalúan los equipos pertenecientes a una red en busca de debilidades, mejor conocidos como puntos de entrada que pueden ser explotados por los ciberdelincuentes que esperan obtener acceso a sus datos (Avast Business Team, 2018).

Los escáneres comparan los servicios y aplicaciones que se ejecutan en el sistema con una base de datos de debilidades conocidas en el servicio, incluidos puertos y scripts que potencialmente podrían ser explotados por piratas informáticos que buscan puntos de entrada. En ese sentido, los escáneres de vulnerabilidad actúan como 'hackers' que investigan vulnerabilidades potenciales.

Los escaneos se pueden ejecutar de dos maneras diferentes. El primero se utiliza sin autenticación, lo presentar al sistema como un extraño lo vería, sin permisos de acceso especiales. El segundo tipo, es un análisis autenticado, en el cual se tendría el mismo acceso que un usuario típico en el sistema (Avast Business Team, 2018).

2.2.2. Pruebas de penetración

También conocido como 'pen testing', son un tipo de pruebas en las cuales los expertos en ciberseguridad deliberadamente 'atacan' una red para revisar lo seguro que tan segura es. El objetivo es simular un ataque real, pero de forma controlada. Como tal, el término "hacking ético" a veces se aplica a las pruebas de penetración (Avast Business Team, 2018). Generalmente, este tipo de pruebas es más avanzado que las pruebas de escaneo.

Las pruebas de penetración se diferencian de los análisis de vulnerabilidades en que los últimos resaltan cualquier debilidad en una red, mientras que las pruebas de penetración incluyen esto y además el determinar qué tipo de actividad maliciosa es posible si esas debilidades son explotadas. En otras palabras, cómo sus debilidades pueden ser hackeadas.

Estas pruebas se pueden realizar desde dentro o fuera de una red, según los expertos de la corporación Avast (creadores del antivirus Avast), dependiendo de lo que la prueba está tratando de descubrir. Un caso sería poder realizar una prueba para probar cómo se podría acceder a la base de datos de clientes o qué tan fácil sería acceder a la red empresarial desde un determinado equipo.

Los autores Shaukat et al. (2016) reconocen tres tipos de metodologías a para realizar pruebas de penetración. Están las pruebas de caja blanca, también conocidos como pruebas de “información completa”, donde el equipo de IT realiza un análisis integral, que evalúa toda la infraestructura del sistema, como topografía, contraseñas, IPs, claves de acceso y todos los demás datos que se refieren a la red - servidores, firewalls, etc. Por otro lado, están las pruebas de caja negra, también conocidas como pruebas de “información cero”, en las cuales no se posee información específica el sistema. Por lo tanto, es el más cercano a simular un ataque externo. Finalmente, tenemos las pruebas de caja gris, conocidas como pruebas de “aprendizaje incompleto”. Es una mezcla de los dos tipos anteriores, donde solo se posee cierta información para realizar la prueba de intrusión.

Metodologías de Pruebas de Penetración

Shaukat et al., (2016), indican que hay dos tipos de metodologías para las pruebas de penetración, las públicas y las propietarias. Las públicas incluyen frameworks conocidos como CISSP, CISA, OWASP y CHECK, los cuales pueden ser accedidos en la red por el público que tenga un interés en estas metodologías. Por su parte, las propietarias pertenecen a empresas privadas y por lo tanto no están regularmente accesible al público.

Los tipos de Pruebas de Penetración

a) **Prueba de Servicios en Red:** se realizan análisis en la infraestructura de red de la corporación, en busca de fragilidades que pueden ser solidificadas. En este aspecto, se evalúa la configuración del firewall, pruebas de filtrado, entre otras.

b) **Prueba en Aplicación Web:** se realizan pruebas de intrusión profundas, con un análisis detallado para verificar las vulnerabilidades en aplicaciones web.

c) **Prueba de 'Client Side':** en este tipo de prueba, es posible explorar software, programas de creación de contenido y Web browsers (como Chrome, Firefox, Microsoft Edge, Opera).

d) **Prueba en Red Inalámbrica:** Se realizan pruebas a los protocolos de red inalámbrica, puntos de acceso y credenciales administrativas, etc.

e) **Prueba de Ingeniería Social:** se realizan pruebas de tipo psicológico para tratar que los usuarios repasen todos los puntos de seguridad, como claves.

Fases de Pruebas de Penetración

El Ministerio de Tecnologías de la Información y las Comunicaciones (MINTIC) de Colombia, en su guía titulada Guía metodológica de Pruebas de Efectividad, establece seis fases principales para las pruebas de penetración, las cuales se detallan brevemente a continuación. Cabe resaltar que las fases de las pruebas de penetración tienen similitud con las fases de ataque de ciberseguridad, pues el objetivo es emular lo que haría un atacante para encontrar las vulnerabilidades a tiempo.

Figura – Fases de las pruebas de Penetración. Fuente: (ReYDeS, s.f.)

Fase	Descripción
Recolección de Información	en esta fase se colecta la información para identificar objetivos específicos. Es en esta fase donde se pueden escoger el tipo de prueba, de caja blanca, de caja negra o caja gris (MINTIC).
Modelado de amenazas	se analizan los riesgos a los que está expuesto el sistema a partir de la información recolectada en la fase anterior.
Análisis de vulnerabilidad	se identifican las brechas de seguridad con la información utilizada en la fase uno.
Ataque o explotación	Se atacan las vulnerabilidades descubiertas en la fase de análisis, utilizando los datos levantados en la fase de recolección.
Ataque posterior	se realizan análisis a los resultados de la fase de ataque.
Reporte	se realiza un informe detallado con los resultados.

Tabla – Descripción de las Fases de las pruebas de Penetración. Autoría: propia.

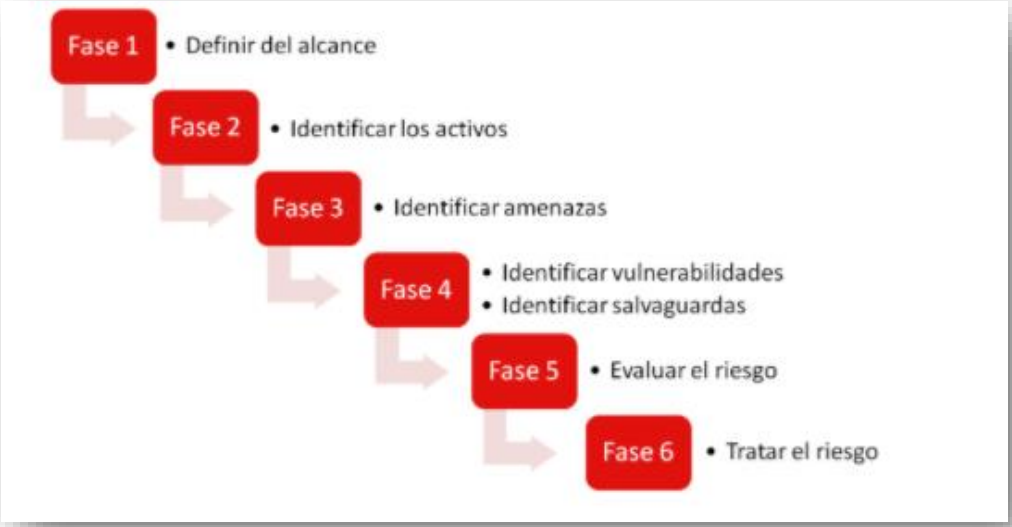
2.2.3. Pruebas de actualización

El equipo de expertos de la corporación Avast afirman que el software que no se actualiza regularmente ofrece a los ‘hackers’ una mejor oportunidad que exploten a los sistemas, ya que no tendrán las últimas actualizaciones. En este tipo de pruebas, los ‘testers’ verifican que los sistemas tengan las ultimas actualizaciones que solucionan el problema problemas de seguridad. Este tipo de pruebas debe utilizarse en conjunto con otro tipo de pruebas como las pruebas de penetración (Avast Business Team, 2018).

2.2.4. Análisis de Riesgos

Los expertos de INCIBE (2017) afirman que el análisis de riesgos es uno de los tipos de pruebas más importantes a la hora de mejorar la seguridad de la información. Lo definen como un conjunto de fases, dirigidas a reducir los riesgos a los que está expuesta la organización hasta unos niveles aceptables a partir de un análisis de la situación inicial. Este tipo de análisis permitirá centrar la atención en los riesgos asociados a los sistemas, procesos y elementos dentro del alcance del plan de pruebas. El objetivo es mitigar la posibilidad de tener algún tipo de incidente de ciberseguridad.

Fases del Análisis de Riesgo



INCIBE (2017). ¡Fácil y sencillo! Análisis de riesgos en 6 pasos [figura].

Fase 1. Definir el alcance

"El primer paso a la hora de llevar a cabo el análisis de riesgos es establecer el alcance del estudio" [INCIBE, 2017]. Los autores recomiendan que el análisis de riesgos cubra a totalidad las áreas estratégicas sobre las que mejorar la seguridad. Por ejemplo, análisis de riesgos sobre los procesos del sistema del departamento Administración, análisis de riesgos sobre el sistema de contabilidad o análisis de riesgos relacionados con la página web de la empresa, etc.

Fase 2. Identificar los activos

Una vez definido el alcance, se debe identificar los activos más importantes que guardan relación con el departamento, proceso, o sistema objeto del estudio. Puede ser suficiente con hacer uso de una hoja de cálculo o tabla como la que se muestra a continuación a modo de ejemplo:

ID	Nombre	Descripción	Responsable	Tipo	Ubicación	Crítico
ID_01	Servidor 01	Servidor de contabilidad.	Director Financiero	Servidor (Físico)	Sala de CPD1	Sí
ID_02	RouterWifi	Router para la red WiFi de cortesía a los clientes.	Dept. Informática	Router (Físico)	Sala de CPD1	No
ID_03	Servidor 02	Servidor para la página web corporativa.	Dept. Informática	Servidor (Físico)	CPD externo	Sí
...						

INCIBE (2017). ¡Fácil y sencillo! Análisis de riesgos en 6 pasos [figura].

Fase 3. Identificar las amenazas

El siguiente paso consiste en identificar las amenazas a las que estos están expuestos los activos identificados en el paso anterior. Como el conjunto de amenazas es amplio y diverso debemos hacer un esfuerzo en mantener un enfoque práctico y aplicado. *Por ejemplo*, si nuestra intención es evaluar el riesgo de que un servidor de ficheros sea destruido, es conveniente considerar los riesgos como las averías del servidor, la posibilidad de daños la rotura de una cañería o los daños por fuego.

Fase 4. Salvaguardar las vulnerabilidades

Esta fase consiste en estudiar los puntos débiles de los activos y las medidas para contrarréstalas. Imagine una posible vulnerabilidad en un conjunto de ordenadores cuyo sistema antivirus no están actualizados. A la hora de evaluar el riesgo se pueden aplicar penalizaciones para reflejar las vulnerabilidades identificadas (INCIBE, 2017). También se analizarán las medidas de seguridad implantadas en nuestra organización. *Por ejemplo*, es posible que se haya instalado un sistema SAI (Sistema de Alimentación Ininterrumpida) o un grupo electrógeno para abastecer de electricidad a los equipos del CPD. Ambas medidas de seguridad contribuyen a reducir el riesgo de las amenazas relacionadas con el corte de suministro eléctrico.

Fase 5. Evaluar el riesgo

Con las informaciones recolectas en las fases anteriores, se procede a calcular el riesgo. Para cada par activo-amenaza, se estima la probabilidad de que la amenaza se materialice y verifica el impacto sobre el negocio que produciría. El cálculo de riesgo se puede realizar usando tanto criterios cuantitativos como cualitativos (INCIBE, 2017). Por ejemplo, se pueden hacer tres tablas, una evaluando la probabilidad, otra para el impacto y finalmente una para el riesgo. El riesgo será igual a PROBABILIDAD x IMPACTO. Veamos unos ejemplos:

Cualitativo	Cuantitativo	Descripción
Baja	1	La amenaza se materializa a lo sumo una vez cada año.
Media	2	La amenaza se materializa a lo sumo una vez cada mes.
Alta	3	La amenaza se materializa a lo sumo una vez cada semana.

INCIBE (2017). ¡Fácil y sencillo! Análisis de riesgos en 6 pasos [figura Calculo de la probabilidad].

Cualitativo	Cuantitativo	Descripción
Bajo	1	El daño derivado de la materialización de la amenaza no tiene consecuencias relevantes para la organización.
Medio	2	El daño derivado de la materialización de la amenaza tiene consecuencias reseñables para la organización.
Alto	3	El daño derivado de la materialización de la amenaza tiene consecuencias graves reseñables para la organización.

INCIBE (2017). ¡Fácil y sencillo! Análisis de riesgos en 6 pasos [figura Calculo del Impacto].

		IMPACTO		
		Bajo	Medio	Alto
PROBABILIDAD	Baja	Muy bajo	Bajo	Medio
	Media	Bajo	Medio	Alto
	Alta	Medio	Alto	Muy alto

INCIBE (2017). ¡Fácil y sencillo! Análisis de riesgos en 6 pasos [figura Calculo del riesgo].

Fase 6. Tratar el riesgo

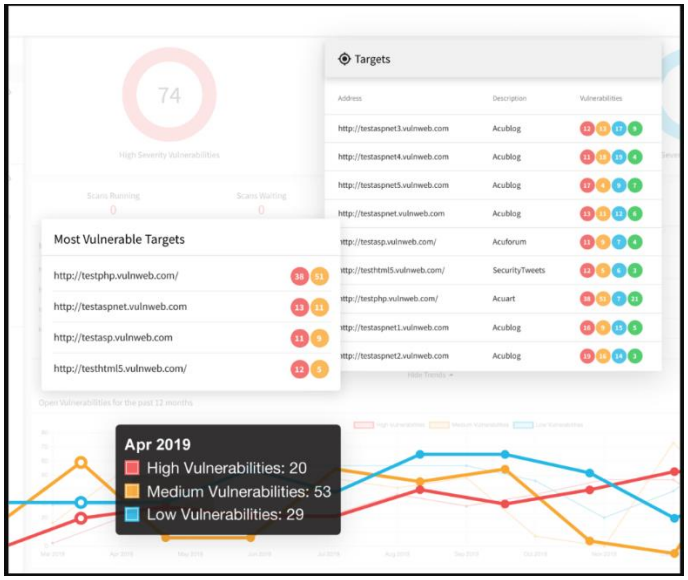
Finalmente, una vez calculado el riesgo, se deben tratar aquellos riesgos que superen un límite que se haya establecido. En los ejemplos anteriores*, se tratarán* aquellos riesgos cuyo valor sea superior a “4” o superior a “Medio” en caso de que hayamos hecho el cálculo en términos cualitativos.

A la hora de tratar el riesgo, existen cuatro estrategias principales: transferir, eliminar, asumir e implantar. Transferir el riesgo es pasarlo a un tercero, ejemplo: un seguro que cubra los daños físicos en los servidores. Eliminar el riesgo es deshacerse de los procesos o sistemas que tengan un riesgo alto, ejemplo: eliminar el ‘WiFi’ gratis si no hay necesidad de ello. Asumir el riesgo es enfrentar las posibles consecuencias, ejemplo: asumir que es posible que los usuarios compartan información debido a prácticas de ingeniería social y que no hay nada que se pueda hacer. Implantar el riesgo es utilizar medidas para mitigarlo, ejemplo: contratar un servicio en la nube para respaldar la información en caso de que se pierda un ‘backup’ (INCIBE, 2017).

2.2.5. Escaneo de Seguridad

Es un tipo de prueba donde, de manera automatizada, se escanea elementos de una red, aplicación o dispositivo para buscar fallas de seguridad. Se realiza con regularidad para garantizar que la información y las aplicaciones

permanezcan seguras. Típicamente se utiliza un ‘software’ especial, conocidos como ‘Aplicaciones para Pruebas de Seguridad Dinámicas’. Un ejemplo de este tipo de aplicaciones es el programa ACUNETIX (¿Qué es un scanner de vulnerabilidades web? – Acunetix, 2020).



Ciberseguridad (2020). ¿Qué es un escáner de vulnerabilidades web? [figura].

En este tipo de pruebas se realiza lo siguiente: escanear y auditar los servidores conectados a Internet en busca de vulnerabilidades. Segundo, identifica las versiones vulnerables de la aplicación y verifica que los servidores no estén ejecutando código maligno como troyanos o malware. Tercero, se utilizan técnicas, como la toma de huellas digitales del sistema operativo (‘OS Fingerprinting’), para descubrir la información que el sistema este exponiendo.

Finalmente, asegura que todos los servicios, como FTP y correo electrónico, no sufran de malware.

Existen los escáneres de vulnerabilidades de web los cuales son herramientas que escanean aplicaciones web, generalmente desde el exterior, para buscar vulnerabilidades de seguridad como 'cross-site scripting', inyección SQL, 'path traversal' y configuración de servidor insegura. El escáner de vulnerabilidades web envía datos maliciosos a un sitio web, como lo haría un 'hacker' malicioso. Sin embargo, lo hace de forma segura. Si la respuesta de su sitio o aplicación web muestra que puede ser hackeado, el escáner lo informa e indica cómo solucionarlo.

2.3. EL MODELO OWASP

2.3.1. Concepto del modelo OWASP

El modelo es un proyecto creado en 2001 por la fundación Open Web Application Security Project (OWASP), sin ánimo de lucro, orientado a promover la seguridad del software, particularmente las aplicaciones web en particular. Es una guía bajo la licencia 'Creative Commons', con contenido desarrollado por decenas de profesionales del área con el desarrollo y seguridad del 'software'. La

guía esta entre las publicaciones más valoradas en relación con el sector de auditorías de seguridad (López, 2014).

En 2008 se editó la versión número tres (3) de la guía, con traducción al idioma castellano, proceso en donde participó activamente el Instituto Nacional de Ciberseguridad de España (INCIBE). Ya en 2014, aparece la versión cuatro (4), consolidándose como el material indispensable para profesionales del desarrollo y pruebas de software y para los especialistas en seguridad de la información (López, 2014).

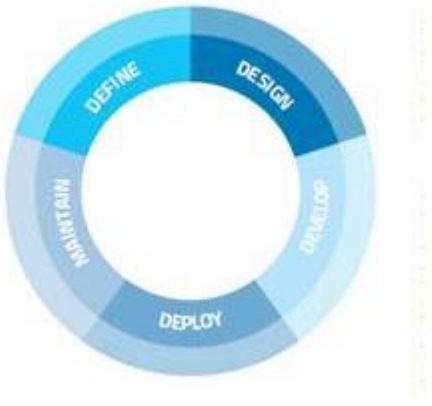
2.3.2. Metodología del modelo OWASP

La guía se presenta de forma organizada y sistemática todas las posibles áreas que supongan vectores de ataque a una aplicación web, para que sea posible a los equipos de tecnología implementarla y seguirla para que puedan auditar de forma eficaz la seguridad de un desarrollo web (López, 2014).

Etapas del Ciclo de vida

La auditoría se organiza en etapas según el estado de madurez en el desarrollo de la aplicación, guiando a los profesionales de seguridad a realizar

actividades durante todo el ciclo de vida de esta. Estas actividades son: definición, diseño, desarrollo, despliegue y finalmente el mantenimiento (López, 2014).



López (2014). Etapas de OWASP Testing Guide v4.0. Guía de seguridad en aplicaciones Web

Fase I. Definir

Al inicio se debe definir un Ciclo de Vida del Desarrollo (SDLC) con la seguridad en mente. Luego de deben establecer las políticas, estándares y documentación apropiadas. Por ejemplo, si se los 'web service' aplicación se desarrollaran utilizando el lenguaje de programación Java, debe de crearse un estándar para la codificación. Sí la aplicación usa criptografía, debe de haber un estándar relacionado. En esta primera fase, también se establecen los criterios de medición – los requerimientos, para tener una idea clara de cuando hay una no conformidad en la seguridad (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

Fase II. Diseño

En esta fase se deben revisar los requerimientos para verificar que no haya asunciones o puntos ciegos. Ejemplo, si hay un requerimiento de seguridad que dicta que un usuario debe ganar acceso antes de acceder a una parte de una aplicación/página web, se debe puntualizar si el usuario tiene que registrarse en la aplicación o solo iniciar sesión. Además, se deberá revisar la documentación de la arquitectura para ver si incluyen los requerimientos de seguridad levantados en la fase anterior. Ejemplo: si el diseño requiere que se utilicen múltiples puntos de acceso en la aplicación, puede ser apropiado crear un solo framework de autorización para que el usuario se inicie sesión solo una vez. Finalmente, en esta fase se crean y se analizan los modelos UML que describen como funcionan la aplicación. Si ya existen, se debe verificar para ver si contienen los requerimientos específicos de seguridad. Utilizando los modelos UML se pueden crear modelos de amenazas para determinar si el diseño responde a las amenazas de manera adecuada. (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

Fase III. Desarrollo

Teóricamente, el desarrollo es la implementación del diseño. Sin embargo, en la práctica, muchas decisiones de diseño se llevan a cabo durante el mismo

debido la naturaleza iterativa del desarrollo de ‘software’ (es posible que, durante la fase de desarrollo, los ingenieros se dieran cuenta de que hacía falta algún detalle). Por ello, la guía OWASP establece que es buena práctica definir revisiones de código, donde los desarrolladores le muestren el código fuente al equipo de seguridad, explicándole la lógica y el flujo del programa. Dicha práctica permite que el equipo que hará las pruebas de seguridad obtenga un entendimiento clave de porque los requerimientos se implementan de tal manera. Al evaluar el código, se recomienda tener una lista de validación que incluya: los requerimientos de negocio, los requerimientos de estándares establecidos (ISO, IEC, etc.), la lista de requerimientos de OWASP (descritos en el documento ‘Los 10 Riesgos De Seguridad Web Mas críticos’), requerimientos particulares de la tecnología que se esté usando (ejemplo Javascript) (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

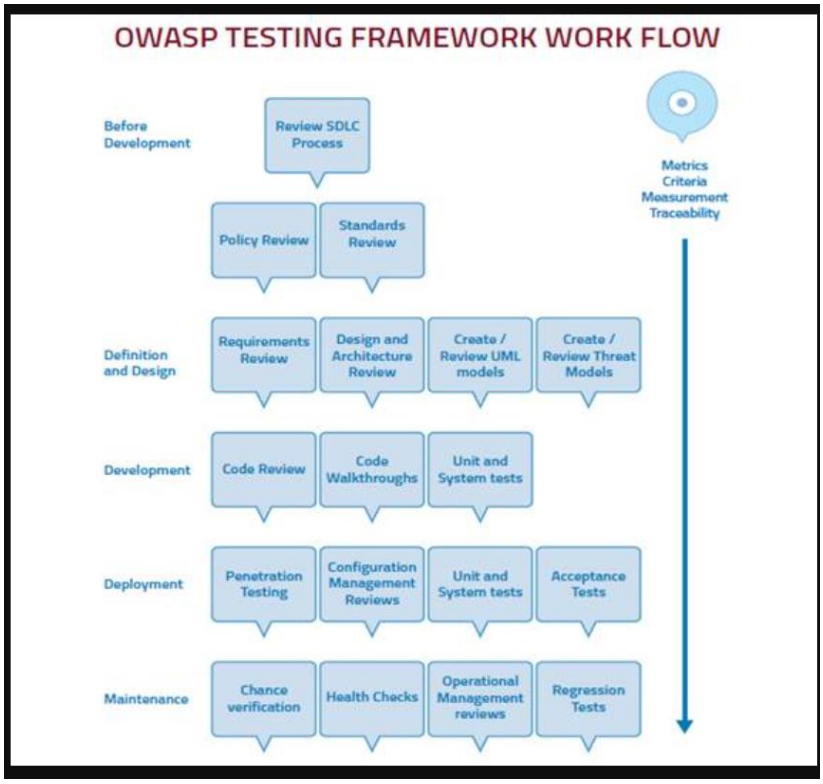
Fase IV. Despliegue

En esta fase se realizan pruebas para verificar la conformidad de los requerimientos de seguridad, además de actividades de análisis y revisión de código. Un ejemplo del tipo de pruebas a realizar en esta fase son pruebas de penetración para confirmar que se cumplió con los requerimientos con éxito. Las pruebas deben incluir pruebas para verificar que la configuración de la aplicación

desplegada en producción es la adecuada. (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

Fase V. Mantenimiento

En la última fase, se llevan a cabo revisiones operacionales para ver que como la aplicación sigue funcionando. Además, se aconseja conducir pruebas de salud (‘health checks’) cada mes, o por lo menos cada cuarto de mes, para verificar que no hayan aparecido nuevos riesgos de seguridad. Finalmente, se debe velar por verificar que, si hay nuevos cambios en la aplicación, dichos cambios no hayan afectado la funcionalidad existente (pruebas de regresión) (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).



López (2014). Framework de OWASP Testing Guide v4.0. Guía de seguridad Web

2.3.3. Framework de pruebas

El framework propone los puntos de control sobre los que se aplicarán las pruebas correspondientes. (López, 2014). La metodología propone dos fases, una pasiva, donde se observa el funcionamiento de la aplicación y analizan todas las funcionalidades posibles. El objetivo de esta fase es entender la lógica de operación e identificar los posibles vectores de ataque y/o vulnerabilidades. La

segunda fase es donde se realizan las pruebas de activa las pruebas según los vectores identificados en la fase anterior.

Las pruebas se agrupan en 11 categorías para sumar un total de 91 puntos de control:

CATEGORIA DE PRUEBA	EJEMPLO
Recopilación de información	Prueba de Huella Para Servidores
	Prueba de Descubrimiento de Motor de Busqueda
	Mapa de Arquitectura de la Aplicación
Pruebas de configuración y administración de despliegue	Prueba de Metodos HTTP
	Prueba de Extension de Archivos
Pruebas de gestión de identidad	Prueba de Registro de Usuarios
	Prueba de Definicion de Roles
Pruebas de autenticación	Prueba de Credenciales
	Prueba de Recuerdo de Contraseña
	Prueba de Política de Contraseña Débil
Pruebas de autorización	Prueba de Escalación de Privilegios
Pruebas de gestión de sesiones	Prueba para Termino de Sesión
Pruebas de validación de entrada	Prueba de Inyección SQL
	Prueba de Inyección XML
	Prueba de Inyección Xpath
Manejo de errores	Análisis de Códigos de Errores
Criptografía	Prueba de SSL Débil
Pruebas de lógica empresarial	Prueba de Validación de Lógica del Negocio
Pruebas del lado cliente	Prueba de Inyección HTML

Figura – Tipos de Prueba de OWASP. Autoría Propia.

A lo largo de los puntos de control se describen todos los pormenores y se presentan ejemplos de las pruebas a realizar en cada categoría. Los puntos cubren temas tan importantes como la inyección SQL, fuga de información,

métodos de autenticación o cifrado débil, validación incorrecta de parámetros y otros muchos son descritos en detalle, proporcionando al auditor una visión clara del problema de seguridad y las contramedidas a adoptar (López, 2014).

CAPÍTULO III: PROPUESTA DE IMPLEMENTACIÓN DEL MODELO OWASP PARA LA SEGURIDAD EN LA EMPRESA WEBARTRD

3.1. JUSTIFICACIÓN

Actualmente, la empresa WebArtRD no incluye las pruebas de ciberseguridad en su estrategia de aseguramiento de la calidad, lo cual la hace vulnerable a potenciales ataques por cibercriminales. La implementación de modelos de aseguramiento de la calidad de software, enfocados a la seguridad web, ayudaría a la empresa dominicana WebArtRD a tener un proceso de control de calidad estándar para todas sus aplicaciones “web” (presentes y futuras), permitiéndoles entregar producto más robusto, en términos de ciberseguridad, a sus clientes. Ayudaría a sus desarrolladores a entender mejor las necesidades de sus clientes, así como les facilitaría comprender como realizar una evaluación de amenazas, riesgos y desafíos específicos.

El equipo de calidad de la empresa podrá crear una colección de pruebas para garantizar el estado seguro la de aplicación, permitiendo utilizar un enfoque pragmático para depurar los casos de prueba. El equipo podrá, por ejemplo, verificar que una pantalla de inicio de sesión cumple con los requisitos de seguridad concernientes a la longitud de la contraseña.

Los gerentes de proyectos y líderes de equipos lograrían hacer mejor uso de los recursos, tanto en términos de tiempo y monetarios, ya que tendrían un cronograma de trabajo que arroje luz sobre las necesidades, objetivos de negocios y requisitos de seguridad para poder manejar las fallas en la calidad de los sistemas. Esto se traduciría en menores defectos en el producto final, reduciendo la necesidad de gastos de postproducción. A partir de estos, la gerencia, en conjunto con los clientes, podrán hacer una evaluación al producto final para validad si sus niveles de calidad se ajustan a lo establecido en la documentación.

3.2. DISEÑO DE LA PROPUESTA

La siguiente propuesta establece una metodología de pruebas de seguridad web a ser incorporadas en el plan de aseguramiento de la calidad actual de la empresa WebArtRD, la cual está basada en la guía de pruebas OWASP la cual detalla las fases y las pruebas que deben realizarse a una aplicación web para verificar sus niveles de seguridad. Sin embargo, la guía es bastante amplia ya que contiene más de 200 tipos de pruebas para seguridad web. De todo el catálogo de pruebas, hemos elegido las más pertinentes considerando el tamaño de la organización, así como de los equipos de desarrollo y de calidad.

La empresa deberá contratar al menos un analista de seguridad, quien hará el análisis y el levantamiento de los requerimientos de seguridad y un tester para que realice las pruebas de seguridad. El salario de un tester ronda entre los 1,500 y 2,000 dólares.

3.3. PLAN DE ACCIÓN

El plan de acción estará divido en dos fases principales: una fase de análisis y otra de prueba. A su vez, la fase de prueba estará divida en dos fases: una fase pasiva y una fase pasiva.

3.3.1. Fase de Análisis

Consiste en hacer un levantamiento de las áreas que potencialmente representan un mayor riesgo para de ese modo priorizar las pruebas. Se realizarán las siguientes tareas:

Identificar el riesgo

El analista identificara el nivel de riesgo de seguridad tomando en cuenta las amenazas, las vulnerabilidades. Finalmente hará una estimación sobre el

impacto en el negocio. (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

Estimación de la probabilidad

Ya identificados los riesgos, se podrá estimar la probabilidad de que una amenaza sea descubierta y explotada. Para ello, se establecerán los siguientes parámetros cuantitativos (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.):

- **Alta**. Define una vulnerabilidad que si es explotada comprometería de forma grave la seguridad de la información ocasionando un impacto negativo. Deben solucionarse inmediatamente.
- **Media**. Define una vulnerabilidad que si es explotada tendría un impacto leve sobre las operaciones del negocio. El tiempo de respuesta puede ser prudente.
- **Baja**. Define una vulnerabilidad que si es explotada tendría un impacto imperceptible sobre las operaciones del negocio. La solución no necesariamente tiene que ser inmediata.

Estimación del impacto

El analista podrá estimar dos tipos de impacto. El impacto técnico, basado en los criterios representados en la siguiente imagen (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.):

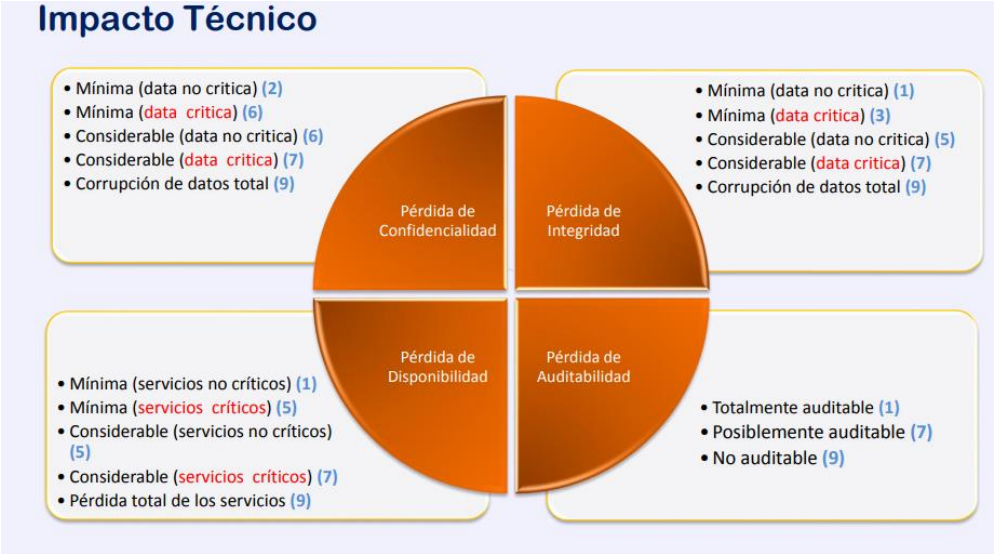


Figura – Impacto técnico. Fuente: Análisis de Riesgos Aplicando la Metodología OWASP (Machaca)

Finalmente, el impacto del negocio, basado en los siguientes criterios (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.):

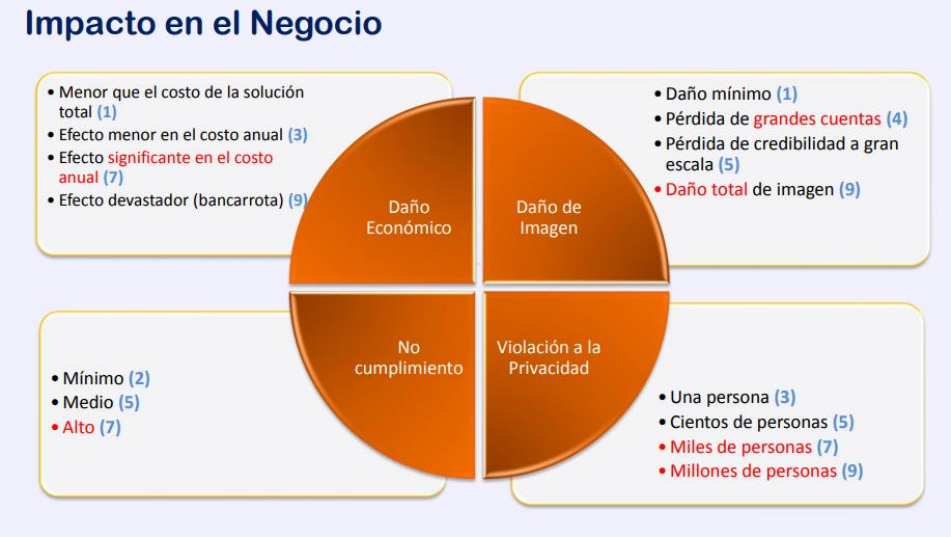


Figura – Impacto del negocio. Fuente: Análisis de Riesgos Aplicando la Metodología OWASP (Machaca)

Determinando la severidad del riesgo

Se realizará un análisis de la severidad del riesgo al evaluar la probabilidad y el impacto, partiendo de un modelo donde se hace una valoración cuantitativa de la probabilidad y el impacto (ejemplo, del 0 al 3 se considera un impacto y probabilidad baja) (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.):

CÁLCULO DE LA PROBABILIDAD E IMPACTO	
0 - 3	Bajo
4 - 6	Medio
7 - 9	Alto

Para el cálculo de la probabilidad, se le asigna un valor del 1 al 10 a cada factor de riesgo y a cada vulnerabilidad. Luego se realiza la de los factores de riesgo y los factores de vulnerabilidad y se saca el promedio. El resultado será la probabilidad. En el siguiente ejemplo, si sumamos todos los factores de riesgo (5 + 2 + 7 + 1) obtenemos 10. Si sumamos los factores de vulnerabilidad (3+6+9+2) obtenemos 20. Si sacamos el promedio 10 + 20 = 35 / 8, obtendremos 4.3 que será la probabilidad.

CALCULO DE LOS FACTORES DE RIESGO			
Nivel de Habilidad	Motivo	Oportunidad	Tamaño
5	2	7	1

CALCULO DE FACTORES DE VULNERABILIDAD			
Facilidad de descubrimiento	Facilidad de explotación	Conciencia	Detección de intrusión
3	6	9	2

Para el cálculo del impacto, se calcula cada nivel de impacto por separado – el técnico y el de negocios. En el siguiente ejemplo, se suman los valores del impacto técnico (9+7+5+8) y se promedia, 29 / 4 = 7.25, el cual es alto si vemos la tabla de probabilidad e impacto. Para el impacto de negocio se hace lo mismo, lo cual dará como resultado 2.25, lo que representa un impacto bajo.

CALCULO DEL IMPACTO TÉCNICO			
Perdida de confidencialidad	Perdida de integridad	Perdida de disponibilidad	Perdida de Responsabilidad
9	7	5	8

CALCULO DEL IMPACTO DE NEGOCIO			
Daño financiero	Daño de Reputación	No conformidad	Violación de privacidad
1	2	1	5

3.3.2. Fase de prueba.

Fase Pasiva

En esta fase el 'tester' analiza la lógica de la aplicación para entenderla. Un caso de uso es que se verifique los enlaces de la aplicación. El tester podría tener una URL como la siguiente:

"https://www.miaplicacionweb.com/login/Authentic_Form.html"

Esta es una URL de autenticación en algún lugar de la aplicación, lo que le dice al tester que deberá prepararse para hacer pruebas de Autenticación.

Fase Activa

En esta fase el ‘tester’ utilizara las siguientes pruebas como se definen en el modelo:

CATEGORIA DE PRUEBA
Recopilación de información
Pruebas de configuración y administración de despliegue
Pruebas de gestión de identidad
Pruebas de autenticación
Pruebas de autorización
Pruebas de gestión de sesiones
Pruebas de validación de entrada
Manejo de errores
Criptografía
Pruebas de lógica empresarial
Pruebas del lado cliente

Figura – Tipos de Pruebas. autoría propia.

3.4. TIPOS DE PRUEBAS ACTIVAS

3.4.1. Pruebas para la recolección de la información

Con este tipo de pruebas el tester podrá verificar el número de puntos de entrada a la aplicación, los archivos metan en el servidor, los comentarios de páginas web y metadatos dejados por los desarrolladores, ya que pueden incluir información que debe ocultarse a terceros (OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web, s.f.).

.

Casos de Uso

<table>
<tr><th colspan="3">Prueba: Descubrimiento por Motores de Búsqueda</th></tr>
<tr><th>Objetivos</th><th>Como Probar</th><th>Herramientas</th></tr>
<tr><td rowspan="4">Comprobar si hay información sensitiva expuesta a través de los motores de búsqueda.</td><td>El tester usara un motor de busque como Google para buscar:</td><td rowspan="4">Google Hacking Database - un sitio que contiene una lista de patrones de búsqueda</td></tr>
<tr><td>contraseñas y nombres de usuario</td></tr>
<tr><td>contenidos de mensajes de errores</td></tr>
<tr><td>correos electrónicos</td></tr>
</table>

Ejemplo: una busqueda basica es buscar el sitio en el motor de busqueda Google para ver si existe utilizando el patron “site:” como muestra la imagen mas abajo.

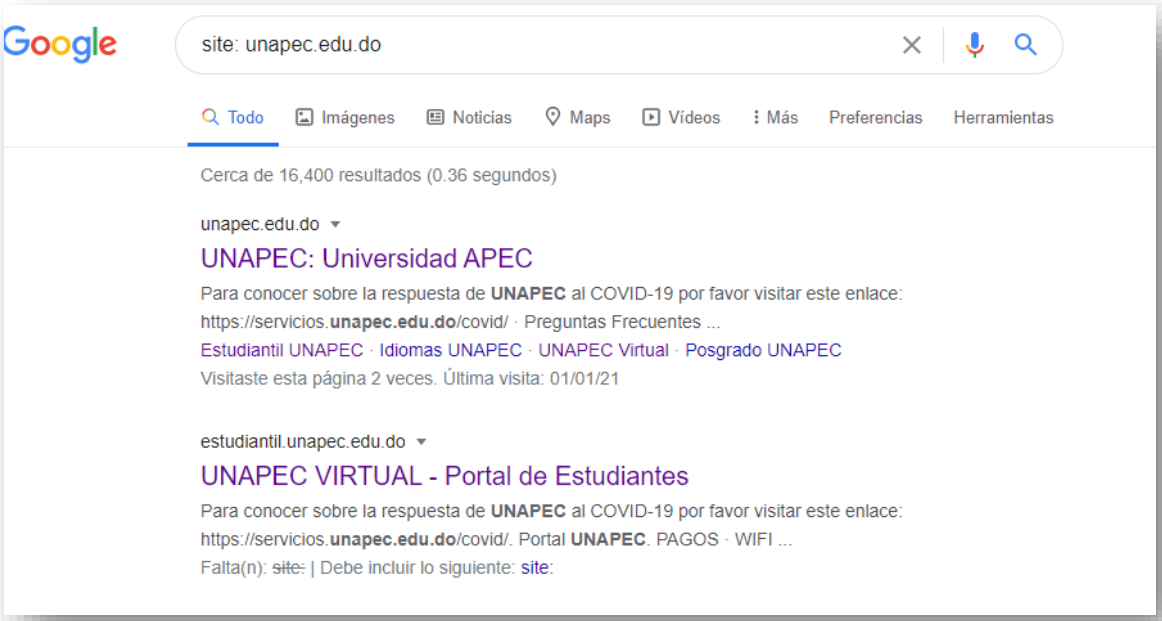


Figura – Pruebas de búsqueda con motores. autoría propia.

Casos de Uso

Prueba: Pruebas para verificar la huella digital del servidor web		
Objetivos	**Como Probar**	**Herramientas**
Comprobar la versión y el tipo de servidor web.	El tester buscara el campo “Servidor” en la respuesta HTTP, utilizando un programa como Netcat.	Netcat

Ejemplo: Considere la siguiente imagen, se puede ver que el servidor es Apache versión 1.3 por el campo “Server”, y el sistema operativo es Linux.

```
$ nc 202.41.76.251 80
HEAD / HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 16 Jun 2003 02:53:29 GMT
Server: Apache/1.3.3 (Unix)  (Red Hat/Linux)
Last-Modified: Wed, 07 Oct 1998 11:18:14 GMT
ETag: "1813-49b-361b4df6"
Accept-Ranges: bytes
Content-Length: 1179
Connection: close
Content-Type: text/html
```

Figura - Respuesta HTTP. Fuente: Fingerprint Web Server (OTG-INFO-002) - OWASP

3.4.2. Pruebas Para Administración de configuración e implementación

Con este tipo de pruebas el tester podrá verificar la configuración de los servidores ya que pueden existir varias aplicaciones en un servidor y una sola vulnerabilidad puede socavar la seguridad de toda la infraestructura.

Casos de Uso

Prueba: Pruebas para verificar la huella digital del servidor web		
Objetivos	**Como Probar**	**Herramientas**
Verificar que los métodos HTTP, los cuales se usan para realizar acciones en el servidor web, no puedan ser utilizados con fines nefastos si el servidor web está mal configurado	El tester utiliza el método OPTIONS HTTP en un programa como NETCAT para ver los métodos HTTP. El método OPTIONS representa una solicitud de información sobre las opciones de comunicación disponibles en la cadena de solicitud/respuesta identificada por una URI de solicitud.	Netcat o Telnet

Ejemplo: Como se muestra en la imagen más abajo, OPTIONS proporciona una lista de los métodos admitidos por el servidor web, indicados en el apartado “Allow”, y en este caso, podemos ver que los métodos GET, HEAD, POST y TRACE están habilitados.

```
$ nc www.victim.com 80
OPTIONS / HTTP/1.1
Host: www.victim.com

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Tue, 31 Oct 2006 08:00:29 GMT
Connection: close
Allow: GET, HEAD, POST, TRACE, OPTIONS
Content-Length: 0
```

Figura – Obteniendo los métodos del servidor. Fuente: Test HTTP Methods (OTG-CONFIG-006) - OWASP

3.4.3. Pruebas de gestión de identidades

Estas pruebas cubren los procesos de creación y reanudación de cuentas de usuario. El objetivo principal d es asegurarse de que la aplicación web permite a los usuarios acceder solo a información específica que está alineada con sus roles. El tester se enfocará en buscar indicios de que las contraseña son débiles o si el proceso de registro de usuarios es débil. También debe probar definiciones de roles y procesos de registro de cuentas.

Casos de Uso

<table>
<tr><th colspan="3">Prueba: Definiciones de roles de prueba</th></tr>
<tr><th>Objetivos</th><th>Como Probar</th><th>Herramientas</th></tr>
<tr><td rowspan="3">Validar que los roles del sistema definidos dentro de la aplicación definen y separan suficientemente cada sistema y función empresarial para gestionar el acceso adecuado a la funcionalidad e información del sistema.</td><td>Desarrollé una matriz de permisos, la cual debe enumerar todos los roles que se pueden aprovisionar y explorar los permisos que se permiten aplicar a los objetos, incluidas las restricciones.</td><td>Pruebas Manuales</td></tr>
<tr><td>Ingrese a la aplicación con cada uno de los roles</td><td rowspan="2">Opcionalmente puede utilizar una aplicación de spidering</td></tr>
<tr><td>Verifique que solo tiene acceso a las funcionalidades descritas por los roles</td></tr>
</table>

Ejemplo: Con los roles establecidos en la siguiente tabla, entrar a sitio web con rol de cliente y verificar que el cliente solo tiene acceso a su historial de compra, y no al de nadie más.

Role	Permiso	Objeto	Limites
Administrador	Lectura	Historial de Compra del Cliente	Sin limites
Gerente	Lectura	Historial de Compra del Cliente	Solo el historial de su departamento
Cliente	Lectura	Historial de Compra del Cliente	Solamente su historial de compra

Casos de Uso

Prueba: Proceso de registro del usuario		
Objetivos	**Como Probar**	**Herramientas**
Compruebe que los requisitos de identidad para el registro de usuarios están alineados con los requisitos empresariales y de seguridad.	1. Compruebe que los requisitos de identidad para el registro de usuarios están alineados con los requisitos empresariales y de seguridad:	Un proxy HTTP puede ser una herramienta útil para probar este control.
Valide el proceso de registro.	¿Cualquiera puede registrarse para acceder?	
	¿Puede la misma persona o identidad registrarse varias veces?	
	¿Se verifican las identidades registradas?	

	¿Qué prueba de identidad se requiere para que un registro tenga éxito?	
	2. Validar el proceso de registro:	
	¿Se puede falsificar o falsificar fácilmente información de identidad?	
	¿Se puede manipular el intercambio de información de identidad durante el registro?	

Ejemplo: en la página de WordPress, el único requerimiento de seguridad es una dirección de correo electrónico.

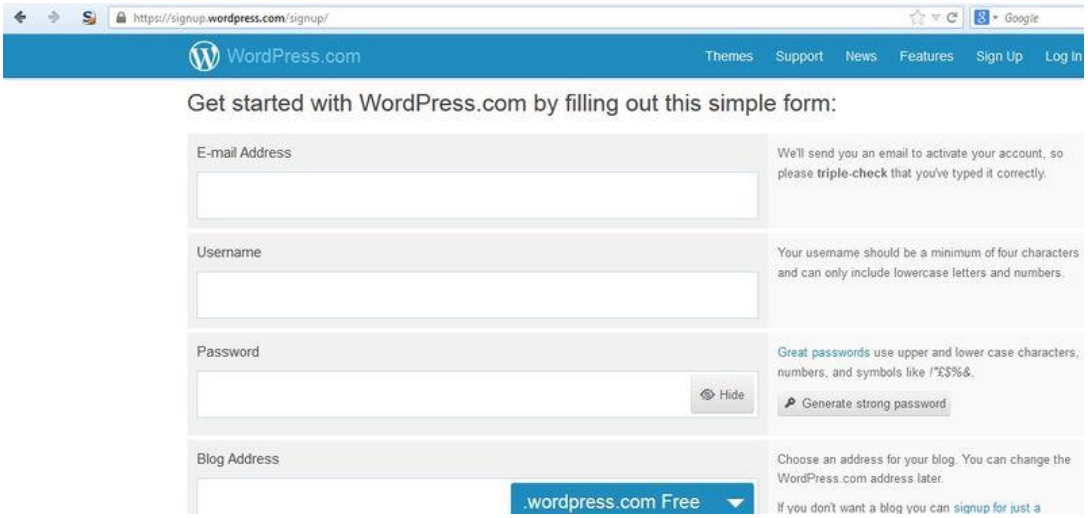


Figura – Pruebas Para Registro de Usuarios. Fuente: [Test User Registration Process (OTG-IDENT-002)]

3.4.4. Pruebas de Autenticación

Con estas pruebas de seguridad se podrá asegurar que algún tercero no tenga oportunidad de obtener acceso a la aplicación. El tester deberá comprobar:

- sin el navegador no recuerda datos sensibles
- que las contraseñas y hashes no se almacenan en las cookies
- comprobar si las respuestas a las preguntas de seguridad se pueden adivinar fácilmente
- si el CAPTCHA es resistente a los ataques de fuerza bruta
- verificar que los datos y credenciales de usuario se transfieren a través de un canal.
- un usuario que ha iniciado sesión no puede cambiar la contraseña sin escribir la contraseña existente

Casos de Uso

<table>
<tr><th colspan="3">Prueba: Política de contraseñas débiles</th></tr>
<tr><th>Objetivos</th><th>Como Probar</th><th>Herramientas</th></tr>
<tr><td rowspan="5">Determine la resistencia de la aplicación contra los ataques de fuerza brutas, utilizando diccionarios de contraseña disponibles mediante la evaluación de la longitud, complejidad, reutilización y envejecimiento de los requisitos de contraseñas.</td><td>Verificar que caracteres están permitidos y prohibidos para contraseña</td><td rowspan="5">Manual</td></tr>
<tr><td>Utilizar caracteres de diferentes conjuntos de caracteres, como letras mayúsculas y minúsculas, dígitos y símbolos especiales</td></tr>
<tr><td>Verificar la frecuencia con la que se cambiar la contraseña</td></tr>
<tr><td>Verificar que tan pronto un usuario puede cambiar su contraseña luego de cambiarla por primera vez</td></tr>
<tr><td>Verificar el tiempo de cambio de las contraseñas (ejemplo: cada 90 días)</td></tr>
</table>

	Verificar si la aplicación mantiene un historial de contraseñas	
	Verificar si la nueva contraseña puede ser la misma que la vieja contraseña	
	Verificar si se puede utilizar el nombre de usuario como contraseña	

Casos de Uso

Prueba: Recordar Contraseña		
Objetivos	**Como Probar**	**Herramientas**
Verificar si las contraseñas se almacenan en texto sin cifrar o si se pueden recuperar fácilmente en formularios codificados o cifrados en cookies.	Verificar si hay contraseñas que se almacenan en cookies, y si es el caso verificar si están como testo normal o proteger por hash	Manual
	Verificar el algoritmo de hash para probar si es un algoritmo fuerte.	
	Verificar si el hash se puede adivinar fácilmente	
	Verificar que la contraseña se envíe solo en la fase de autenticación	

3.4.5. Pruebas de manejo de Sesión

La administración de sesiones es un componente principal de las aplicaciones web, ya que cubre desde el momento en que los usuarios se autentican hasta que cierran sesión.

El objetivo de estas pruebas es engañar a la aplicación para que otorgue acceso a una cuenta de usuario sin haber proporcionado las credenciales. El tester deberá:

- Verificar los de atributos de cookies, ya que a menudo son el primer vector de ataque
- Buscar vulnerabilidades de fijación de sesión que pueda ayudar a los atacantes a obtener acceso a la cuenta de un usuario a través de una sesión activa
- Identificar las variables de sesión expuestas que pueden permitir a un atacante hacerse pasar por un usuario autorizado
- Hacer pruebas de falsificación de solicitudes entre sitios para ver si es posible comprometer los datos y operaciones de los usuarios finales
- Comprobar la funcionalidad de cierre de sesión para definir si la duración de los tokens de sesión es lo suficientemente corta como para evitar un ataque de secuestro de sesión

Casos de Uso

Prueba: Tiempo de espera de la sesión		
Objetivos	**Como Probar**	**Herramientas**
En esta fase, el tester comprueban que la aplicación cierra la sesión automáticamente a un usuario cuando este ha	El tester comprueba si existe un tiempo de espera, al iniciar sesión y esperar a que se active el cierre de sesión. Al igual que en la función de cierre de sesión, una vez transcurrido el tiempo de espera, todos los tokens de	N/A

estado inactivo durante un cierto tiempo	sesión deben destruirse o ser inutilizables.	
	Si existe el tiempo de espera, el tester debe verificar si el cliente o el servidor (o ambos) aplican el tiempo de espera.	
	Si la cookie de sesión no es persistente (si la cookie de sesión no almacena ningún dato sobre el tiempo), el tester pueden asumir que el tiempo de espera es aplicado por el servidor.	
Garantiza que no es posible "reutilizar" la misma sesión y que no quedan datos confidenciales almacenados en la caché del explorador.	Si la cookie de sesión contiene datos relacionados con el tiempo (tiempo de inicio de sesión o última hora de acceso), es posible que sea el cliente participe en el tiempo de espera	
	En el último caso, el tester intenta modificar la cookie (si no está protegida criptográficamente) para ver qué sucede con la sesión. Ejemplo, los evaluadores pueden establecer la fecha de caducidad de la cookie en el futuro y ver si la sesión se puede prolongar.	

3.4.6. Pruebas de validación de entrada

El tester valida que los datos de entrada que proceden del cliente o del entorno son procesados antes de utilizar estos datos, con la finalidad de comprar si la aplicación es sujeta a las inyecciones SQL, los ataques del sistema de archivos y las inyecciones de intérpretes. Una prueba crítica es comprobar si hay vulnerabilidades de formato de las cadenas de texto.

Casos de Uso

Prueba: Inyección de SQL		
Objetivos	**Como Probar**	**Herramientas**
Comprobar que no se puede inyectar SQL en los campos de la aplicación	Identificar las partes de la aplicación donde hay una comunicación con la base de datos (campos de autenticación, funcionalidad de búsqueda, etc.)	Manual
	La primera prueba consiste en agregar una comilla simple (') o un punto y coma (;) al campo que se está probando. Estos caracteres, si no la aplicación no los filtras, crear un error.	O una herramienta para inyección de SQL, como OWASP SQLiX
	Usar otros delimitadores como (--, /*, */) y palabras clave de SQL como AND y OR	
	Monitorea las respuestas de la página y revisa el Código fuente por mensajes de errores de SQL	

Ejemplo: Considere la consulta siguiente:

```
SELECT * FROM Users WHERE Username='$username' AND Password='$password'
```

Si la consulta devuelve un valor significa que dentro de la base de datos existe un usuario con ese conjunto de credenciales, el usuario puede iniciar sesión en el sistema, de lo contrario se deniega el acceso. Los valores de los campos de entrada se obtienen generalmente del usuario a través de un formulario web. Supongamos que insertamos los siguientes valores de nombre de usuario y contraseña:

```
$username = 1' or '1' = '1
```

```
$password = 1' or '1' = '1
```

La consulta seria:

```
SELECT * FROM Users WHERE Username='1' OR '1' = '1' AND Password='1' OR '1' = '1'
```

Si los valores de los parámetros se envían al servidor a través del método GET, y si el dominio del sitio web vulnerable es www.example.com, la solicitud que llevaremos a cabo será:

```
http://www.example.com/index.php?username=1'%20or%20'1'%20=%20'1&password=1'%20or%20'1'%20=%20'1
```

Casos de Uso

Prueba: Inyección de XML		
Objetivos	**Como Probar**	**Herramientas**
Comprobar que no se puede insertar un documento XML en la aplicación.	El primer paso para probar una aplicación para la presencia de una vulnerabilidad de inyección XML consiste en intentar insertar meta caracteres XML como comilla simple (') y comillas dobles (").	
	Una vez realizado el primer paso, el probador tendrá información sobre la estructura del documento XML. A continuación, es posible intentar insertar datos XML y etiquetas.	

Casos de Uso

Prueba: Inyección de XML		
Objetivos	**Como Probar**	**Herramientas**
Comprobar que no se puede insertar un documento XML en la aplicación.	El primer paso para probar una aplicación para la presencia de una vulnerabilidad de inyección XML consiste en intentar insertar meta caracteres XML como comilla simple (') y comillas dobles (").	
	Una vez realizado el primer paso, el probador tendrá información sobre la estructura del documento XML. A continuación, es posible intentar insertar datos XML y etiquetas.	

3.5. VENTAJAS Y DESVENTAJAS DEL MODELO OWASP

La Guía de Pruebas de OWASP adecuada como base en pruebas de seguridad para aplicaciones web. OWASP cubre las vulnerabilidades más frecuentes en aplicaciones web (Gonzáles Brito, 2018).

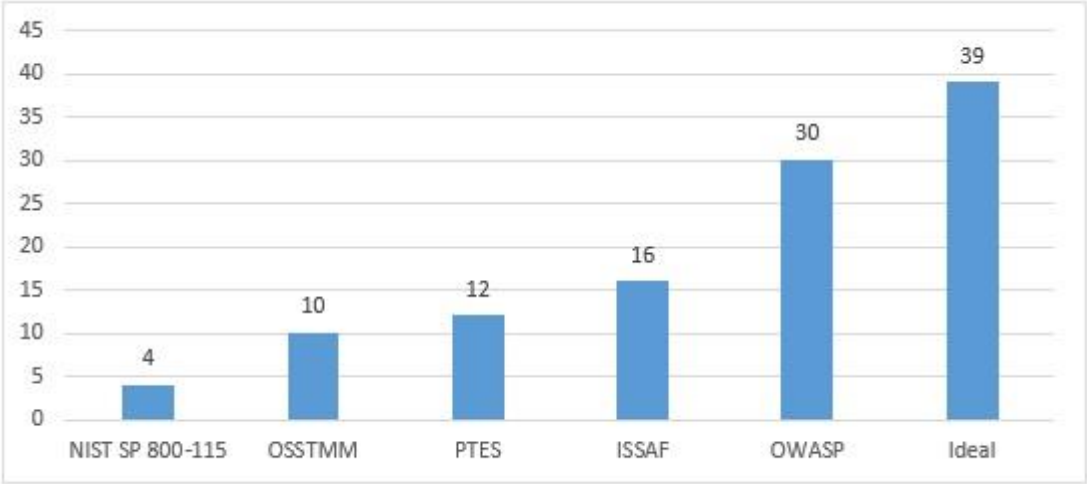


Figura – Capacidad para la realización de pruebas de penetración. Fuente: Gonzales, Brito (2018). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones.

Sin embargo, si se compara con otras metodologías de pruebas de penetración, OWASP trata poco el tema de la gestión del proceso. También es muy repetitivo en su framework, donde se encuentran el mismo tipo de pruebas de seguridad repetidas en diferentes renglones. Tampoco se mencionan aspectos organizativos como por ejemplo las actividades de establecimientos de alcances

y contratos de confidencialidad entre las partes o procesos de planificación y seguimiento y control.

4. CONCLUSIÓN

La ciberseguridad de es un elemento importante del aseguramiento de calidad de ‘software’. Los defectos relacionados con la seguridad deben considerarse como un problema de control de calidad, debido a que las aplicaciones con defectos y fallos de calidad son más propensas a las amenazas de seguridad, conduciendo a un comportamiento impredecible. Para el usuario, se puede manifestar como una no conformidad o mala usabilidad. Mientras que, para un atacante, proporciona una ventana de oportunidad para atacar el sistema de maneras inesperadas.

En un mundo cada vez enfocado en el uso de las tecnologías de la información para realizar tareas diarias, las vulnerabilidades de seguridad, al ser explotadas, pueden empañar seriamente la reputación de una empresa, e incluso ocasionarle pérdidas millonarias. Una vulnerabilidad de seguridad podría resultar en robo de información confidencial, como datos de los empleados (seguro social), información personal del cliente y hasta del mismo negocio, como datos de las cuentas bancarias y propiedades intelectuales. Por estas razones, el aseguramiento de calidad se considera la principal línea de defensa en lo que a seguridad se refiere, a pesar de la creencia generalizada de que la seguridad y el control de calidad están divorciados.

Armados con una comprensión del propio producto y metodologías de para asegurar la seguridad de las aplicaciones de ‘software’, como el modelo OWASP para pruebas de penetración, los ingenieros de aseguramiento de calidad añaden valor al negocio y garantizan su calidad, disminuyendo las vulnerabilidades de seguridad.

Con el diseño presentado en el capítulo tres, se fue capaz de crear un modelo que utiliza estándares internacionales para asegurar la calidad de las aplicaciones web de la compañía WebArtRD. Los ejecutivos de esta han decido implementar nuestra p en un ambiente isolado de prueba como veto para verificar si se puede implementar de manera permanente en la empresa. Para

5. RECOMENDACIONES

- Se recomienda que los equipos de desarrollo y de calidad trabajen de la mano con el equipo de seguridad de 'software', en aquellas empresas donde exista este último equipo, para que se pueda tener un panorama general de la seguridad de las aplicaciones del negocio. Esto ayudara a todos a estar en la misma sintonía, logrando que las vulnerabilidades puedan ser detectadas mucho más rápido.

- En aquellas empresas donde no exista un equipo dedicado a la seguridad, como **WebArtRD**, se recomienda a los ejecutivos realizar una inversión para la creación de un equipo de seguridad dedicado, o como mínimo, que se capacite a los equipos de calidad y desarrollo en temas de seguridad, para que estos se hagan cargo de estas tareas.

- Se recomienda que definan los requisitos funcionales y no-funcionales de seguridad que la aplicación deberá cumplir.

- Se recomiendan considerar las pruebas de seguridad automatizadas como un componente principal en el proceso de desarrollo y mantenimiento, ya que permiten cubrir un número mayor de pruebas en menor tiempo, en comparación con las pruebas manuales. Esto conlleva la posible inversión en herramientas profesionales de automatización.

- Se recomienda que las empresas desarrolladoras de aplicaciones web, como WebArtRD, entrenen a su personal no informático en temas de seguridad e ingeniería social, para reducir el riesgo de que un empleado sea víctima de alguno

de los numerosos ataques como ‘ransomware’ o ‘phishing’. Esta recomendación es válida, además, para cualquier empresa no importa a lo que se dedique.

- Se recomienda habilitar ambientes únicos de pruebas y no probar en producción.
- Se recomienda a la empresa WebArtRD incrementar su personal de calidad, debido a que consideramos que es muy pequeño, resultando en una carga de trabajo mayor a los encargados de las pruebas, resultando así que todas las pruebas no se hagan con calidad, resultando en muchas omisiones por falta de tiempo.
- Se recomienda la realización de cursos y certificaciones en el tema de ciberseguridad al personal de tecnología y desarrollo, así como al resto del personal.

6. REFERENCIAS

¿Qué es un scanner de vulnerabilidades web? – Acunetix. (30 de junio de 2020). Obtenido de Cyberseguridad: https://www.cyberseguridad.com.mx/que-es-un-scanner-de-vulnerabilidades-web-acunetix/

About NIST. (s.f.). Recuperado el 31 de Enero de 2021, de https://www.nist.gov/about-nist.

About the OWASP Foundation [Sobre la Fundacion OWASP]. (s.f.). Obtenido de OWASP: https://owasp.org/about/

Anonymous 2020: ¿quiénes son y por qué son tendencia? (2 de junio de 2020). Recuperado el febrero de 2021, de El País: https://www.elpais.com.uy/vida-actual/anonymous-quienes-son-son-tendencia.html

Arroyo Guardeño, D., Gayoso Martínez, V., & Hernández Encinas, L. (2020). *Ciberseguridad.* MAdrid, España: CSIC Consejo Superior de Investigaciones Científicas. Recuperado el enero de 2021, de https://elibro.net/es/lc/unapec/titulos/172144

Arroyo Guardeño, D., Gayoso Martínez, V., & Hernández Encinas, L. (2020). *Ciberseguridad.* MAdrid, España: CSIC Consejo Superior de Investigaciones Científicas. Recuperado el enero de 2021, de https://elibro.net/es/ereader/unapec/172144

Atlassian. (s.f.). *JIRA SOFTWARE*. Obtenido de JIRA SOFWARE: https://www.atlassian.com/es/software/jira

Avast Business Team. (28 de noviembre de 2018). *3 cybersecurity tests and tools for protecting your small business*. Obtenido de Avast: https://blog.avast.com/cybersecurity-tests

Báez Pérez, C. I., & Suárez Zarabanda, M. I. (2013). *Proceso de desarrollo de software: basado en la articulación de RUP y CMMI priorizando su calidad* (Primera ed.). Tunja, Boyacá, Colombia: Universidad de Boyacá. Obtenido de https://elibro.net/es/lc/unapec/titulos/129062

Calidad de Software: Argentum. (s.f.). Obtenido de Argentum: https://www.argentuminc.com/calidad-de-software/

Campderrich Falgueras, B. (2013). *Ingeniería del software.* Editorial UOC. Obtenido de https://elibro.net/es/lc/unapec/titulos/56294

Capgemini. (20 de noviembre de 2020). Obtenido de Capgemini, Sogeti and Micro Focus Research: World Quality Report 2020-21: https://www.capgemini.com/news/world-quality-report-2020/

Carrizo, D., & Alfaro, A. (Marzo de 2018). Método de aseguramiento de la calidad en una metodología de desarrollo de software: un enfoque práctico. *Ingeniare. Revista chilena de ingeniería, 26*(1), 114-119. doi:10.4067/S0718-33052018000100114

Chavarria, A. E., Oré, S. B., & Pastor, C. (diciembre de 2016). Aseguramiento de la Calidad en el Proceso de Desarrollo de Software utilizando CMMI, TSP y PSP. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, 1-16. doi:10.17013/risti.20.62-77

Cibersecurity Framework. (s.f.). Recuperado el 31 de Enero de 2021, de NIST: https://www.nist.gov/industry-impacts/cybersecurity-framework

CISA. (22 de octubre de 2009). *Avoiding Social Engineering And Phishing Attacks*. Obtenido de Cibersecurity & Infrastructure Security Agency (CISA): https://us-cert.cisa.gov/ncas/tips/ST04-014

CISA. (s.f.). *ABOUT CISA*. Obtenido de Cibersecurity & Infrastructre Security Agency: https://www.cisa.gov/about-cisa

CISCO. (s.f.). *What Is a Hacker?* Obtenido de CISCO: https://www.cisco.com/c/en/us/products/security/what-is-a-hacker.html#~more-resources

Clay, K. (2013). *Amazon.com Goes Down, Loses $66,240 Per Minute*. Recuperado el enero de 2021, de Amazon: https://www.forbes.com/sites/kellyclay/2013/08/19/amazon-com-goes-down-loses-66240-per-minute/?sh=562e4b57495c

Code Academy. (22 de julio de 2020). *What is a programming language?* Obtenido de CodeAcademy: https://news.codecademy.com/programming-languages/

Con Quien Trabajamos. (s.f.). Recuperado el 31 de Enero de 2021, de INCIBE: https://www.incibe.es/que-es-incibe/con-quien-trabajamos/membresias

Dell Corporation. (s.f.). *What is a Server?* Obtenido de Dell: https://www.dell.com/downloads/us/bsd/What_Is_a_Server.pdf

Diaz Diaz, S. (s.f.). *Pruebas de seguridad en aplicaciones web.* Obtenido de https://www.unab.edu.co/sites/default/files/MemoriasGrabadas/papers/capitulo7_paper_13.pdf

Dueñas Huaroto, J. J., & Jáuregui Montalva, C. M. (2014). Aseguramiento de la calidad en una empresa de desarrollo de software. *Tesis*. Lima, Peru: Universidad Peruana de Ciencias Aplicadas (UPC). Recuperado el febrero de 2021, de https://repositorioacademico.upc.edu.pe/handle/10757/324664

Ferrer Martínez, J. (2014). *Aplicaciones web* (Vol. 0). RA-MA Editorial. Recuperado el enero de 2021, de https://elibro.net/es/lc/unapec/titulos/106407

Folgar, O. F. (1997). *ISO 9000 - Aseguramiento de La Calidad folgar.* Macchi Grupo Editor. Recuperado el enero de 2021

Gallagher, M. (2019). *The Little Book of Cyber Scams 2.0.* Obtenido de London Metropolitan Police: https://www.met.police.uk/SysSiteAssets/media/downloads/central/advice/fraud/met/little-book-of-cyber-scams-2.0.pdf

García-Moran, J. P., Fernández Hansen, Y., & Martínez Sánchez, R. (2014). *Hacking y Seguridad en Internet: edición 2011.* RA-MA Editorial. Recuperado el enero de 2021, de https://elibro.net/es/lc/unapec/titulos/106415

Gonzáles Brito, H. R. (octubre de 2018). Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones we. *Revista Cubana de Ciencias Informaticas*. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000400005&lng=pt&nrm=iso

IBM Corporation. (s.f.). *What is software development*. Obtenido de IBM: https://www.ibm.com/topics/software-development

Imperva. (s.f.). *Penetration Testing*. Obtenido de Imperva: https://www.imperva.com/learn/application-security/penetration-testing/

INCIBE. (16 de enero de 2017). *¡Fácil y sencillo! Análisis de riesgos en 6 pasos*. Obtenido de INCIBE: https://www.incibe.es/protege-tu-empresa/blog/analisis-riesgos-pasos-sencillo

INCIBE. (8 de marzo de 2017). *Ransomware: una guía de aproximación para el empresario.* Obtenido de INCIBE: https://www.incibe.es/protege-tu-empresa/guias/ransomware-guia-aproximacion-el-empresario

INCIBE. (s.f.). *Ayuda ransomware*. Obtenido de INCIBE: https://www.incibe.es/protege-tu-empresa/herramientas/servicio-antiransomware

ISO 9000. (s.f.). Recuperado el enero de 2021, de ISO.ORG: https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en

Jabaloyes Vivas, J., Carot Sierra, J. M., & Carrión García, A. (2020). *Introducción a la gestión de la calidad.* Valencia, España: Editorial de la Universidad Politécnica de Valencia. Recuperado el febrero de 2021, de https://elibro.net/es/lc/unapec/titulos/165233

JEST. (s.f.). *JEST*. Obtenido de JEST: https://jestjs.io/

Jimeno Muñoz, J. (2019). *Derecho de daños teconológicos, ciberseguridad e insurtech.* Madrid, España: Dykinson. Obtenido de https://elibro.net/es/lc/unapec/titulos/118410

Konitz, M. (25 de October de 2019). *Quality assurance trends for 2020*. Obtenido de Codilime: https://codilime.com/quality-assurance-trends-for-2020/

Laravel LLC. (s.f.). *Laravel*. Obtenido de Laravel: https://laravel.com/

Laskowski, J. (2011). *Why software quality assurance and IT security need to work together: IBM*. Recuperado el 2021, de IBM: https://www.ibm.com/developerworks/rational/library/software-quality-assurance-IT-security/index.html#:~:text=The%20alliance%20of%20quality%20assurance,are%20concerned%20with%20removing%20risks.

Lemonnier, J., & Latto, N. (2 de enero de 2020). *¿Qué es el spyware?* Obtenido de AVG: https://www.avg.com/es/signal/what-is-spyware

liam. (2016). *Extensible Markup Language (XML)*. Obtenido de W3School: https://www.w3.org/XML/

Lizarzaburu, E., Chávez, M., & Barriga, G. (2018). *Gestión de Operaciones y Calidad.* Pearson Educación. Recuperado el febrero de 2021, de https://elibro.net/es/lc/unapec/titulos/136611

López, A. (15 de octubre de 2014). *OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web*. Recuperado el enero de 2021, de INCIBE-CERT: https://www.incibe-cert.es/blog/owasp-4

Machaca, A. (s.f.). *ANÁLISIS DE RIESGOS APLICANDO LA METODOLOGIA OWASP.* Obtenido de OWASP: https://owasp.org/www-pdf-archive/Analisis_de_riesgo_usando_la_metodologia_OWASP.pdf

Martínez, C. (18 de agosto de 2017). *¿Cuál es el origen y la utilidad de un sistema de gestión de calidad?* Recuperado el febrero de 2021, de Revistadigital INESEM: https://revistadigital.inesem.es/gestion-integrada/sistema-gestion-calidad/

Mikhalchuk, O. (2 de October de 2020). *What to expect in 2021:7 top trends in quality assurance and testing*. Obtenido de Forté Group: https://fortegrp.com/latest-innovation-trends-in-software-testing-technologies/

MINTIC. (s.f.). *Guía Metodológica de Pruebas de Efectividad.* Obtenido de Ministerio de Tecnologías de la Información y las Comunicaciones: https://www.mintic.gov.co/gestionti/615/articles-5482_G1_Metodologia_pruebas_efectividad.pdf

Mozilla. (s.f.). *¿Qué es una URL?* Obtenido de Developer Mozilla: https://developer.mozilla.org/es/docs/Learn/Common_questions/What_is_a_URL

Mozilla. (16 de julio de 2020). *Seguridad de Sitios Web*. Obtenido de Mozilla: https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Website_security

Mozilla. (s.f.). *Frameworks Web de lado servidor*. Obtenido de Mozilla: https://developer.mozilla.org/es/docs/Learn/Server-side/First_steps/Web_frameworks

Mozilla. (s.f.). *HTML*. Obtenido de Developer Mozilla: https://developer.mozilla.org/en-US/docs/Glossary/HTML

Mozilla. (s.f.). *La web y los estándares web*. Obtenido de Developer Mozilla: https://developer.mozilla.org/es/docs/Learn/Getting_started_with_the_web/The_web_and_web_standards

NIST Mission, Vision, Core Competencies, and Core Values. (s.f.). Recuperado el 31 de Enero de 2021, de NIST: https://www.nist.gov/about-nist/our-organization/mission-vision-values

Nuñez Fernández, E. (2007). *Archivos y normas ISO.* (Á. Díaz Huici, Ed.) España: Ediciones Trea. Recuperado el febrero de 2021, de https://elibro.net/es/lc/unapec/titulos/60537

Ousterhout K., J. (s.f.). Obtenido de Web Standord: https://web.stanford.edu/~ouster/cgi-bin/papers/scripting.pdf

Outsourcing: FL Betances. (s.f.). Obtenido de FL Betances: https://www.flbetances.com/servicio-de-outsourcing--personal

OWASP Testing Guide v4.0. Guia de seguridad en aplicaciones Web. (s.f.). Obtenido de INCIBE: https://www.incibe-cert.es/blog/owasp-4

Pantaleón Contreras, D., & Laureano Ramirez, J. A. (2011). Estudio de auditoría de sistemas en el área de aseguramiento de calidad de software. Estudio de caso : Solutech Sistemas Informáticos. Distrito Nacional, Santo Domingo, República Dominicana: Universidad APEC. Recuperado el enero de 2021, de https://eds.a.ebscohost.com/eds/detail/detail?vid=0&sid=c42bf091-a0c7-4f60-9f75-c8074b94a8e0%40sdc-v-sessmgr01&bdata=Jmxhbmc9ZXMmc2l0ZT1lZHMtbGl2ZQ%3d%3d#AN=dbu.32671&db=cat07658a

Pola Maseda, Á. (2009). *Gestión de la calidad.* Marcombo. Recuperado el febrero de 2021, de https://elibro.net/es/lc/unapec/titulos/45847

Polanco, W. (28 de julio de 2020). Webinar: "Ecosistemas inteligentes potenciados mediante analítica de datos en tiempo real". Recuperado el febrero de 2021, de https://www.youtube.com/watch?v=QIFJPTA811o&t=2845s

Polanco, W. (s.f.). *Aseguramiento de la Calidad del Software Impulsada por Metodologías Ágiles:UNIBE*. Recuperado el enero de 2021, de UNIBE: https://www.unibe.edu.do/aseguramiento-de-la-calidad-del-software-impulsada-por-metodologias-agiles/

Pressman, R. S. (2010). *Ingeniería de Software: Un Enfoque Practico* (Septima ed.). (V. Campos Olguín, & J. E. Brito, Trads.) Connecticut: MCgraw Hill. Recuperado el 30 de enero de 2021, de https://www.academia.edu/15231805/Ingenieria_de_software_enfoque_practico_7ed_Pressman_PDF

Proyectos Agiles. (s.f.). *Qué es SCRUM*. Obtenido de Proyectos Agiles: https://proyectosagiles.org/que-es-scrum/

Public-Private Partnership for Semiconductor Research. (s.f.). Recuperado el 31 de Enero de 2021, de NIST: https://www.nist.gov/industry-impacts/public-private-partnerships-semiconductor-research

Quality Assurance, Quality Control and Testing — the Basics of Software Quality Management. (s.f.). Obtenido de Alterxsoft: https://www.altexsoft.com/whitepapers/quality-assurance-quality-control-and-testing-the-basics-of-software-quality-management/

Qué es INCIBE. (s.f.). Obtenido de INCIBE: https://www.incibe.es/que-es-incibe

Que es INCIBE: INCIBE. (s.f.). Recuperado el 31 de enero de 2021, de INCIBE: https://www.incibe.es/que-es-incibe

Ramos Alcazar, G. M. (s.f.). *HISTORIA Y EVOLUCIÓN DEL CONCEPTO DE GESTIÓN DE CALIDAD*. Recuperado el febrero de 2021, de Sutori: https://www.sutori.com/story/historia-y-evolucion-del-concepto-de-gestion-de-calidad--VMb6P4wrEX1F3M7fgKtHtjRr

ReYDeS. (s.f.). *El Proceso de Pruebas de Penetración*. Obtenido de Reydes: http://www.reydes.com/d/?q=El_Proceso_para_Pruebas_de_Penetracion

Reyes, C. (9 de enero de 2018). *Confirman hackearon página web de Cámara de Diputados*. Recuperado el febrero de 2021, de Diario Libre: https://www.diariolibre.com/actualidad/confirman-hackearon-pagina-web-de-camara-de-diputados-FL8940236

Servicios:Newtech. (s.f.). Obtenido de Newtech: https://newtech.software/en/home/#servicios

Shaukat, K., Faisal, A., Masood, R., Usman, A., & Shaukat, U. (2016). Security quality assurance through penetration testing. *19th International Multi-Topic Conference (INMIC)*, (págs. 1-5). doi:10.1109/INMIC.2016.7840115

SQL. (s.f.). Obtenido de Developer Mozilla: https://developer.mozilla.org/en-US/docs/Glossary/SQL

Tester-h. (2019). *¿Qué es un tester y a qué se dedica?* Obtenido de Tester House: https://testerhouse.com/teoria-testing/que-es-un-tester-y-a-que-se-dedica/

Urcuqui López, C. C., García Peña, M., Osorio Quintero, J. L., & Navarro Cadavid, A. (2018). *Ciberseguridad: un enfoque desde la ciencia de datos.* Editorial Universidad Icesi. Obtenido de https://elibro.net/es/lc/unapec/titulos/120435

Valderrama, Á. (25 de Diciembre de 2014). *¿Quién causó realmente el ataque a Sony?* Recuperado el enero de 2021, de CNN Español: https://cnnespanol.cnn.com/2014/12/25/quien-causo-realmente-el-ataque-a-sony/

Villalba Fernández, A. (2015). La ciberseguridad en España 2011-2015: una propuesta de modelo de organización. Madrid, España: D - UNED - Universidad Nacional de Educación a Distancia. Obtenido de https://elibro.net/es/lc/unapec/titulos/86621

W3School. (s.f.). *PHP Tutorial*. Obtenido de W3School: https://www.w3schools.com/php/

W3Schools. (s.f.). *What is Full Stack?* Obtenido de w3schools: https://www.w3schools.com/whatis/whatis_fullstack.asp

What is a web server? (s.f.). Obtenido de Developer Mozilla: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/What_is_a_web_server

Wisseman, S. (s.f.). *Application security and QA: Why they are better together: TechBeacon*. Recuperado el 2021, de TeachBeacon: https://techbeacon.com/app-dev-testing/application-security-qa-why-they-are-better-together

Woody, C., Ellison, R. J., & Nichols, W. (2014). *Predicting Software Assurance Using Quality and Reliability Measures.* Software Engineering Institute, Pittsburgh. doi:10.1184/R1/6582113.v1

7. ANEXOS

ANEXO 1. Entrevista al señor Yoel Leonardo

La siguiente entrevista fue realizada al fundador y Desarrollador en Jefe de WebArtRD

- **Entrevistador:** Para empezar, cuéntenos sobre usted - su nombre, nivel de estudios y a que se dedica.

- ***Ing. Leonardo:*** *Mi nombre es Yoel Leonardo. Soy Ingeniero en Sistemas graduado de la Universidad UNAPEC y me dedico al desarrollo de aplicaciones web.*

- **Entrevistador:** ¿Cuántos años de experiencia tiene en el área de desarrollo web?

- ***Ing. Leonardo:*** *Aproximadamente 10 años.*

- **Entrevistador:** ¿Es empleado público o privado?

- ***Ing. Leonardo:*** *Cuento con mi propia empresa, WebArtRD.*

- **Entrevistador:** Cuéntenos un poco sobre WebArtRD. ¿Como surge?

- ***Ing. Leonardo:*** *Surgió como un proyecto personal en el cual desarrollaba páginas web, alrededor del 2008, en un pequeño cubículo de la oficina de mi madre, quien se dedica a los bienes raíces. Desde entonces, la empresa ha crecido un poco y ya cuenta con 10 empleados, incluyéndome a mí.*

- **Entrevistador:** ¿Esta WebArtRD constituida como una empresa formal?

- ***Ing. Leonardo:*** *Sí.*

- **Entrevistador:** ¿Dónde está ubicada actualmente?

- ***Ing. Leonardo:*** *En la 27 de febrero casi con esquina Tiradentes.*

- **Entrevistador:** ¿Se dedican estrictamente al desarrollo web?

- ***Ing. Leonardo:*** *Sí.*

- **Entrevistador:** Si puede responder, ¿Qué proyectos ha realizado la empresa y/o a cuáles empresas les han trabajado? dígase, les han desarrollado páginas web.

- ***Ing. Leonardo:*** *No puedo dar muchos detalles de los proyectos en sí por cuestiones de confidencialidad. Pero, algunas de ellas son la empresa Fusion Multimedia, el Ministerio de Relaciones Exteriores (de República Dominicana) y el Ministerio de Educación (de República Dominicana). Y de los clientes internacionales, puedo mencionar MidInfinity, la cual es una compañía canadiense a la cual le trabajamos actualmente como contratistas.*

- **Entrevistador:** ¿Cómo está estructurada la empresa? Me refiero a la estructura organizacional. Puestos… etc.

- ***Ing. Leonardo:*** *Estoy yo como fundador y Desarrollador en Jefe a la vez, y diría que en la misma rama se encuentra Jenny, mi esposa quien es la Directora Administrativa que se encarga de las cuestiones administrativas, valga la redundancia, como la contratación y asuntos de recursos humanos. Luego tenemos la división de desarrollo que cuenta con gerente de proyectos, la cual levanta los requerimientos de 'software', dos desarrolladores web, un analista programador, un diseñador gráfico y dos*

analistas de calidad que hacen las pruebas. También contamos con una publicista que responde directamente al área administrativa.

- **Entrevistador:** A su esposa Jenny. Bien. Ahora, si puede contarnos un poco sobre el proceso de desarrollo. ¿Qué metodología utilizan y cómo funciona?

- ***Ing. Leonardo:*** *Utilizamos la metodología SCRUM para desarrollo ágil. Básicamente el proceso comienza cuando el gerente de proyecto se reúne con el cliente para hacer un levantamiento de los requerimientos. Luego, basados en ellos el analista programador y el diseñador gráfico diseñan el sistema (el diseñador gráfico diseña la interfaz y el analista programador el funcionamiento de la lógica del negocio). El documento de diseño se le muestra al cliente y cuando este lo aprueba se empieza el desarrollo hasta que pasa la certificación de QA y se entrega al cliente.*

- **Entrevistador:** ¿Qué sucede si el cliente no aprueba el documento?

- ***Ing. Leonardo:*** *Se realizan las revisiones de lugar hasta que este apruebe el documento.*

- **Entrevistador:** Entiendo. Ahora quisiera que nos cuente como es el proceso de aseguramiento de la calidad. Dijo que contaban con dos analistas…

- ***Ing. Leonardo:*** *Sí. Como utilizamos la metodología SCRUM, la cual es cíclica… es decir mientras se va haciendo el desarrollo los analistas de calidad van realizando las pruebas. Se sostienen reuniones periódicas para ver el avance y como está el proyecto.*

- **Entrevistador:** ¿Qué tipos de pruebas realizan?

- ***Ing. Leonardo:*** *Pruebas unitarias.*

- **Entrevistador:** ¿Tienen alguna herramienta para el reporte de defectos y para realizar los planes de prueba?

- ***Ing. Leonardo:*** *Sí. Jira y Confluence*

- **Entrevistador:** ¿En qué lenguajes desarrollan?

- ***Ing. Leonardo:*** *Para las interfaces, HTML. Para el backend, php y Javascript.*

- **Entrevistador:** Dentro de las pruebas, ¿incluyen pruebas de seguridad web?

- ***Ing. Leonardo:*** *Actualmente no. Pero uno de nuestros clientes nos ha pedido que las incluyamos por un incidente que sucedió en producción.*

- **Entrevistador:** ¿Puede darnos detalles?

- ***Ing. Leonardo:*** *Lamentablemente no.*

- **Entrevistador:** Entiendo, ya que es un riego dar detalles de vulnerabilidades de seguridad. Estaría poniendo en riesgo su empresa y a su cliente.

- ***Ing. Leonardo:*** *Exacto! … Tal vez pueda darte algunos detalles, pero necesito que queden fuera de la entrevista y de la documentación que va a realizar.*

- **Entrevistador:** Oh, ok.

- ***Ing. Leonardo:*** *… Por ahora me limitare a decir que fue un incidente donde alguien tuvo acceso a información a la que no estaba autorizado por una falla en el código.*

- **Entrevistador:** ¿Esto puedo ponerlo en la entrevista?

- ***Ing. Leonardo:*** *… Sí. Pero otros detalles que te daré, no por lo que te comenté.*

- **Entrevistador:** Entiendo. Tiene mi palabra de que seré prudente. También quisiera saber si puedo contactarlo cuando necesite más información o que me aclare algo, por ejemplo, del proceso de desarrollo y de aseguramiento de la calidad.

- ***Ing. Leonardo:*** *Claro.*