UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de Investigación previo a la obtención del título de Ingeniero en Sistemas y Computación

TRABAJO DE TITULACIÓN

METODOLOGÍA OWASP EN EL DESARROLLO DE UN WEBSITE PARA VOTO ELECTRÓNICO. CASO PRÁCTICO: SISTEMA DE ELECCIONES ASOCIACIÓN DE ESTUDIANTES TI-UNACH.

Autores:
Miriam Elizabeth Morocho Buñay
Fabricio Andrés Tasan Garcés

Tutora:
Ing. Lorena Paulina Molina Valdiviezo

Riobamba - Ecuador
Año 2020

PÁGINA DE ACEPTACIÓN

Los miembros del tribunal de Graduación del proyecto de investigación de título: **"METODOLOGÍA OWASP EN EL DESARROLLO DE UN WEBSITE PARA VOTO ELECTRÓNICO. CASO PRÁCTICO: SISTEMA DE ELECCIONES ASOCIACIÓN DE ESTUDIANTES TI-UNACH"**, presentado por los estudiantes Srta. Miriam Elizabeth Morocho Buñay y el Sr. Fabricio Andrés Tasan Garcés, dirigido por la PhD. Lorena Molina Valdiviezo.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación escrito, con fines de graduación en el cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

PhD. Lorena Molina
Tutora del Proyecto

……
Firma

PhD. Fernando Molina
Miembro del Tribunal

……
Firma

Mg. Henry Villa
Miembro del Tribunal

……
Firma

DERECHO DE AUTORÍA

La responsabilidad del contenido de este proyecto de graduación corresponde exclusivamente a: Miriam Elizabeth Morocho Buñay y Fabricio Andrés Tasan Garcés bajo la dirección de la Ing. Lorena Molina Valdiviezo, y al patrimonio intelectual de la Universidad Nacional de Chimborazo.

Autores

…….

Miriam Elizabeth Morocho Buñay
060508877-2

…….

Fabricio Andrés Tasan Garcés
060506155-5

Directora del proyecto

Ing. Lorena Molina, PhD.
060322815-6

DEDICATORIA

El trabajo de investigación va dedicado a Dios por brindarme fuerza, sabiduría y salud necesaria para seguir adelante con mi formación personal y académica.

A mis padres Francisca Buñay y Gerardo Morocho, quienes confiaron en mí y en todo momento tuve el apoyo incondicional, y aunque mi padre esté en el cielo sé que nunca me dejó sola.

A mis hermanos quienes de una u otra manera me ayudaron y acompañaron durante todo el proceso de estudios.

Y finalmente a mis compañeros de clases con quienes compartimos momentos únicos e inolvidables, y sobre todo a Fabricio Tasan que en el encontré un amigo incondicional.

Miriam Elizabeth Morocho Buñay

El presente trabajo investigativo está dedicando principalmente a mi mamá Alexandra Garcés Montero quien con su amor, confianza, paciencia y esfuerzo me ha permitido llegar a cumplir hoy un sueño de los más anhelados.

También está dedicado a mi abuelita, mis tíos/as y cada una de las personas que han estado para mí de una u otra forma, quienes que con sus consejos y palabras de aliento hicieron de mí una mejor persona y de una u otra forma me acompañan en todos mis sueños y metas.

A mis amigos con los cuales compartí grandes momentos en las aulas de la universidad en especial a mi compañera y gran amiga Elizabeth Morocho quien con su apoyo y motivación logramos alcanzar esta meta profesional.

Fabricio Andrés Tasan Garcés

AGRADECIMIENTO

En el presente trabajo de investigación queremos agradecer a Dios por darnos la salud, vida y permitirnos cumplir una de las metas propuestas en nuestras vidas.

Expresamos nuestro sincero agradecimiento a nuestra querida institución Universidad Nacional de Chimborazo quien se convirtió en nuestro segundo hogar, en la cual obtuvimos grandes experiencias y conocimientos.

Un agradecimiento especial a nuestra tutora de tesis Ing. Lorena Molina que con su apoyo, guía y motivación nos ayudó a culminar con éxito la investigación planteada.

También queremos agradecer al Dr. Antonio Meneses, Ing. Henry Villa e Ing. Fernando Molina, quienes con sus conocimientos aportaron en el desarrollo de la investigación.

Por otro lado, agradecemos a todos los docentes de la carrera de Sistemas y Computación, quienes día a día impartieron sus conocimientos para la preparación de nuestra vida profesional, en especial al Ing. Jorge Delgado quien estuvo presto para brindarnos su colaboración en todo momento.

Miriam Elizabeth Morocho Buñay

Fabricio Andrés Tasan Garcés

ÍNDICE GENERAL

LISTA DE TABLAS

LISTA DE FIGURAS

RESUMEN

La seguridad es importante en el desarrollo de websites y más aun tratándose del voto electrónico, así pues, el propósito de esta investigación fue sistematizar el proceso de votaciones para la elección de Asociación de Estudiantes de la carrera de Tecnología de la Información de la Universidad Nacional de Chimborazo.

Se desarrolló dos websites, un website aplicando la metodología OWASP y otro desarrollado vertiginosamente, cumpliendo los siguientes requerimientos: registro de partidos políticos con sus candidatos, sufragio, entrega del certificado, contabilización de votos, y finalmente, la emisión de resultados.

Posteriormente, se realizó pentesting a los dos websites utilizando Kali Linux evaluando: disponibilidad, confidencialidad, integridad y autenticidad, al analizar los resultados obtenidos se evidenció un 19.15% de seguridad en el website desarrollado vertiginosamente, a diferencia del website desarrollado con la metodología OWASP que alcanzo un 91.75% de seguridad, de esta manera se demostró que la metodología planteada permite desarrollar un website con menos vulnerabilidades.

Finalmente, se realizaron las votaciones participando 65 estudiantes de la carrera de TI, una vez terminado dicho proceso, el website con la metodología OWASP contabilizó 65 votos, correspondientes al total de votantes, a diferencia del website desarrollado de forma vertiginosa que contabilizó 78 votos, pues, algunos sufragantes pudieron votar más de una vez, por lo tanto se concluye que la metodología OWASP permitió mejorar la seguridad en el desarrollo de un website para el proceso de votación electrónica.

Palabras Clave: OWASP, Pentesting, Seguridad, Voto electrónico, Website.

ABSTRACT

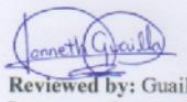
Security is essential in the development of websites, and even more so when it comes to electronic voting. Thus, this research aimed to systematize the voting process for the election of the Student Association of the Information Technology career of the National University of Chimborazo.

Developed two websites, a website applying the OWASP methodology and another developed vertiginously, fulfilling the following requirements: registration of political parties with their candidates, suffrage, delivery of the certificate, counting of votes, and finally, the issuance of results.

Subsequently, testing was made to the two websites using Kali Linux evaluating: availability, confidentiality, integrity, and authenticity. When analyzing the results obtained was evidence a 19.15% of security in the website developed vertiginously. Unlike the website developed with the OWASP methodology that reached 91.75% security, in this way, it demonstrated that the proposed methodology allows developing a website with fewer vulnerabilities.

Finally, voting was carried by participating 65 students of the IT career; once that process was complete, the website with the methodology; OWASP counted 65 votes, corresponding to voters' total, Unlike the website developed in a dizzying way that counted 78 votes. Some voters were able to vote more than once; therefore, it concluded that the OWASP methodology allowed to improve the security in developing a website for the electronic voting process.

Keywords: OWASP, Pentesting, Security, Electronic Voting, Website.



Reviewed by: Guailla, Janneth
Language Center Teacher

INTRODUCCIÓN

En la última década los sitios web se han convertido en puertas para el desarrollo tecnológico debido a la creciente presencia del internet, sin embargo, la información en internet debe estar segura [1]. En Perú, se analizaron el cumplimiento de políticas de seguridad, comprendiendo la estructura de riesgos que puedan afectar a los sistemas informáticos, enfocándose también en el comportamiento humano, tomando en cuenta la información como el activo más importante a proteger [2].

La seguridad en sitios web tiene la finalidad de minimizar vulnerabilidades que puedan afectar al funcionamiento correcto de dichos sitios web, para ello, se analizan técnicas, metodologías para mantener las características esenciales que son: integridad, confidencialidad y disponibilidad [3]. Es por lo que, la metodología OWASP facilita el *top ten* de detección de vulnerabilidades para un sitio web, permitiendo mejorar la seguridad [4].

Por consiguiente, en Cuba se realizaron investigaciones acerca de la metodología OWASP, demostrando que es más completa para disminuir vulnerabilidades en sitios web, con un 76% de satisfacción frente a otras como ISSAF, OSSTM, PTES y NIST SP 800-115, asimismo, se tomaron en cuenta vulnerabilidades como: inyección de código, perdida de autentificación y gestión de sesiones, secuencia de comandos cruzados con sus siglas (XSS), entidades externas de Lenguaje de Marcado Extensible (XML), registro y monitoreo insuficiente, entre otros, [5].

Es así, en México, Argentina, Venezuela y Estados Unidos desarrollaron prototipos de voto electrónico, en los cuales, las vulnerabilidades de los sistemas no fueron controladas, teniendo impactos negativos al evaluar los resultados de las votaciones [6].

El proceso de votación para elección de asociación de estudiantes de la Carrera de Ingeniería en Tecnologías de la Información (TI) de la Universidad Nacional de Chimborazo (UNACH) es llevado en forma manual, dicho proceso, conlleva tiempo en difundir los resultados de la votación, por tal motivo, se requirió sistematizar el proceso a través de un website seguro.

Por consiguiente, surgió la investigación en la que se desarrolló dos sitios web, en el primer caso, un website aplicando la metodología OWASP y en el segundo caso, un

website desarrollado vertiginosamente. Se realizó un pentesting con el fin de probar que el sitio web con la metodología planteada es más segura.

Los websites se desarrollaron en base a la información de los estudiantes, dicho sistema permitió el: registro de información de partidos políticos con sus respectivos candidatos, sufragio y emisión del certificado, contabilización de votos, y finalmente, la difusión de resultados, los cuales permitieron cumplir con los requerimientos de proceso de votación para la elección de Asociación de Estudiantes de la carrera de TI de la Universidad Nacional de Chimborazo.

El siguiente documento obedece a la siguiente estructura: En el capítulo 1, se aborda el planteamiento del problema y los objetivos, en el capítulo 2 se ve plasmado el estado del arte, en el capítulo 3 se puede visualizar la metodología, el capítulo 4 describe los resultados y discusión, finalmente, se cierra con las conclusiones, recomendaciones y anexos.

CAPÍTULO I

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Problema y Justificación

En la actualidad, los sitios web son utilizados por la mayoría de las personas, siendo un medio de comunicación, sin embargo, la seguridad de estos se ven vulnerados, pues, surgen riesgos que atentan contra la disponibilidad, confidencialidad, e integridad de la información que viaja por internet [7].

Los websites de votación electrónica a igual que otros sitios se ven amenazados por distintos ataques tales como: suplantación de identidad, adquisición de credenciales de un votante, manipulación de software, votación más de una vez, entre otros [8].

De esta manera, los atacantes buscan beneficiarse de las vulnerabilidades del sistema, permitiendo que el proceso de votación electrónica no sea íntegro, aquellas vulnerabilidades son por la falta de técnicas criptográficas, políticas y procedimientos de seguridad en el sitio web [9].

La metodología OWASP puede dar solución a este problema, pues, proporciona un *top ten* de detección de vulnerabilidades, ayudando a mejorar los procedimientos y técnicas de seguridad en el desarrollo de sitios web, siendo importante en un sistema de voto electrónico [4].

En la Carrera de Ingeniería en Tecnologías de la Información de la UNACH, el proceso de votación para elección de asociación de estudiantes es llevado en forma tradicional, aquel proceso, conlleva tiempo en difundir los resultados de la votación, por tal motivo, se requirió sistematizar el proceso a través de un website seguro.

Con la investigación planteada se pretende desarrollar dos sitios web de votación electrónica, en el primer caso, un website aplicando la metodología OWASP y en el segundo caso, un website desarrollado en forma vertiginosa, además, se realizará un pentesting en los sitios web con el fin de comparar resultados y seleccionar el sitio web que brinde mayor seguridad a los usuarios.

Con dicho website se beneficiará a los estudiantes, optimizando tiempo en los procesos de votación, brindando disponibilidad, confidencialidad, integridad, y la confianza de utilizar el website seguro para las elecciones de la Asociación de Estudiantes de Tecnologías de la Información de la Universidad Nacional de Chimborazo.

1.2 Objetivos

1.2.1 General

Desarrollar un website para el voto electrónico de las elecciones para la asociación de estudiantes de la Carrera de Ingeniería en Tecnologías de la Información utilizando la metodología OWASP.

1.2.2 Específicos

- Estudiar las guías de desarrollo de OWASP para proporcionar controles de seguridad primordiales en el sistema.
- Analizar, diseñar y crear un website aplicando la metodología OWASP para el voto electrónico.
- Validar el resultado obtenido a través de un pentesting del website.

CAPÍTULO II

2 ESTADO DEL ARTE RELACIONADO A LA TEMÁTICA

2.1 Sitios Web

Hoy en día, el internet es muy importante para la comunicación, de tal forma que, han dado paso a los sitios web, pues, brindan solución a problemas de acuerdo con las necesidades y requerimientos de cada persona u organización, transmitiendo información entre cliente y servidor [10].

Según investigaciones realizadas en Chile, la selección de herramientas para el desarrollo de websites es importante, permitiendo: orden en el desarrollo, facilidad y sobre todo que trabajen de manera rápida dando resultados positivos [11].

Por otro lado, han surgido metodologías que ayudan al desarrollo de websites que tienen por finalidad resolver problemas existentes en el desarrollo del software, pero es difícil encontrar las ventajas y desventajas de una metodología frente a otra y la complejidad que tienen cada una de ellas, a pesar de aquello, la Metodología de Diseño Hipermedia Orientada a Objetos(OHDM) demuestra poseer un modelado fuerte, sin embargo, no se evalúan metodologías que aporten seguridad en estos sitios web [12].

2.2 Seguridad en sitios web

La seguridad se basa en tres principios como se muestra en la **Figura 1**, con ello, se garantiza: mejor funcionalidad, la información transferida es íntegra, confidencial y está disponible todo el tiempo, para certificar aquello, se debe realizar un análisis de vulnerabilidades del sistema [13]. Las vulnerabilidades están en la mayoría de los sitios web, por tal razón, realizan ataques exitosos diariamente, dichos ataques causan daños a empresas y usuarios al poner en riesgo la información [14].

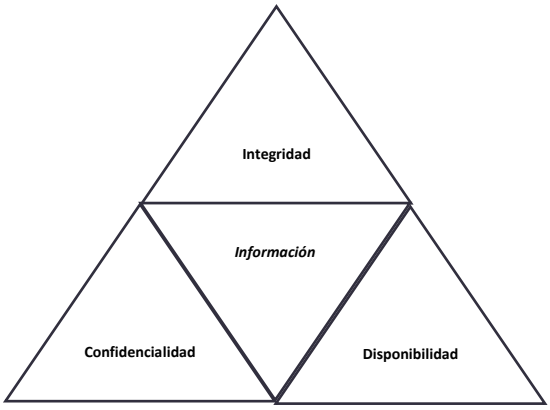


Figura 1: Principios de la seguridad informática

En la India consideraron, Cross-site scripting (XSS) como una vulnerabilidad porque permitió inyección de código JavaScript [14]. En Indonesia analizaron SQL Injection y Cross Site Scripting, mencionaron que deben ser controladas para evitar ataques masivos, pues observaron que la seguridad está en la base de datos [15].

2.3 Metodologías de Seguridad

En Cuba fueron analizados los sitios web, determinando las vulnerabilidades que tenían, el objetivo era verificar que los procedimientos se lleven correctamente, para ello, se basaron en distintas metodologías como: Information Systems Security Assessment Framework (ISSAF), Open Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP), Penetration Testing Execution Standard (PTES) y National Institute of Standards and Technology (NIST SP 800-115), lo cual, permitió enfocarse en la ciberseguridad en el área de desarrollo y mantenimiento de websites [5].

Se tomaron en cuenta distintas vulnerabilidades de las aplicaciones web, entre ellas, inyección de código, pérdida de autenticación, gestión de sesiones, exposición de datos sensibles, falsificación de peticiones en sitios cruzados, entre otras, [5].

En la **Figura 2**, se plasma los resultados correspondientes al porcentaje de seguridad obtenido por cada metodología, después de haber sido evaluadas.

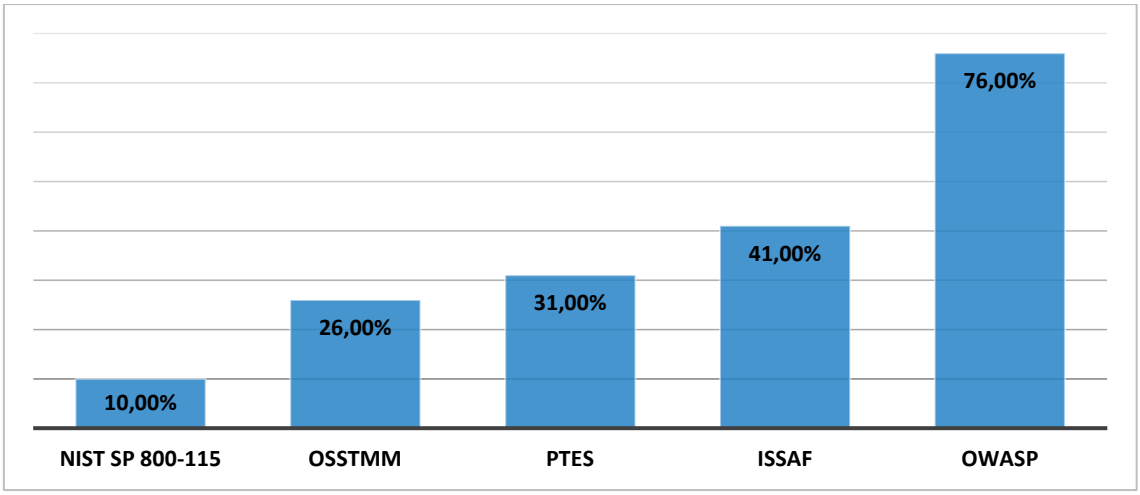


Figura 2: Porcentaje de Seguridad de las Metodologías

2.4 Seguridad con OWASP

Según la página oficial de OWASP emite el top 10 de las vulnerabilidades más graves de aplicaciones web, por lo cual, propone educar a las organizaciones que hacen uso de las TICs sobre las consecuencias de las vulnerabilidades de seguridad en sus aplicaciones web. Estas 10 vulnerabilidades se plasman en la **Figura 3**.

1	• Inyección
2	• Pérdida de Autenticación
3	• Exposición de datos sensibles
4	• Entidades Externas XML (XXE)
5	• Pérdida de Control de Acceso
6	• Configuración de Seguridad Incorrecta
7	• Secuencia de Comandos en Sitios Cruzados (XSS)
8	• Deserialización Insegura
9	• Componentes con vulnerabilidades conocidas
10	• Registro y Monitoreo Insuficientes

Figura 3: Top Ten OWASP

Dentro de este contexto, OWASP como metodología se basa en una guía top ten de vulnerabilidades que se evalúan a través de pruebas llamadas Pentesting centrándose en mejorar la seguridad de sitios web [4].

2.5 Seguridad en la votación electrónica

Según Schmidt y Gutiérrez, en un estudio realizado entre los años 2015 y 2016, desarrollaron un prototipo de sistema de voto electrónico para Costa Rica en las que se analizó que el sistema era vulnerable y susceptible a fraudes, además de los errores, se tienen que enfrentar a posibles ataques internos que traten de esconder un fragmento de código malicioso [6].

Existen algunos casos en donde se vieron atacados los sistemas de votación electrónica, entre los principales están: Estados Unidos, Argentina, y otros países [16]. Los acontecimientos se detallan a continuación:

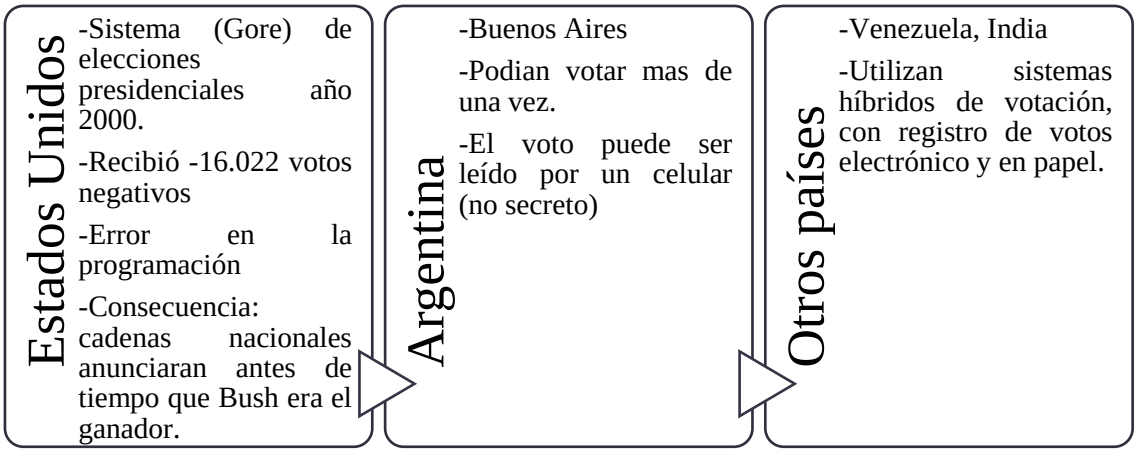


Figura 4: Voto electrónico en diferentes países

2.6 Pentesting

El objetivo de los pentesting es simular el comportamiento de un atacante a un sitio web, de manera que permita descubrir vulnerabilidades, instalando parches de seguridad, modificando permisos, y políticas de usuario, es así como, en Popayán-Colombia, se identificaron las vulnerabilidades más comunes, pues, analizaron 5 sitios web, con 4 herramientas especializadas como son: Acunetix WVS, Nessus, Netsparker, y OWASP ZAP, en las cuales, obtuvieron 12 vulnerabilidades en común. De esta forma, probaron que los pentesting son una alternativa eficiente para medir el nivel de seguridad e integridad de las aplicaciones web [17].

2.7 Kali Linux

Forma parte de una distribución de Debian, el cual integra herramientas que ayudan a realizar pentesting, así pues, en Manabí-Ecuador se utilizó Kali Linux para determinar

vulnerabilidades, se basaron en la metodología Ethical Hacking, centrándose en un entorno virtual en la que lograron obtener acceso a dispositivos debido a la vulnerabilidad que hay en ellos, para aquello, utilizaron algunas herramientas como: Maltego, Set Toolkit, Nmap, Armitage, Metasploit, etc., [18].

Figura 5: Logo de KALI Linux

2.7.1 SlowHttpTest

En una conferencia realizada en Jaipur-India, algunos investigadores mencionan que la herramienta slowhttptest es de relevancia para realizar ataques DoS, de tal forma, enviar cientos de paquetes HTTP a la nube para que la pagina se vuelva lenta, pues, pusieron a prueba la disponibilidad de un servicio web [19].

2.7.2 BurpSuite

Con la plataforma BurpSuite se puede acceder a herramientas como, proxy, spider, intruder, scanner, sequencer, repeater, decoder, comparer, y extender, muy importantes para el escaneo de vulnerabilidades, es por ello que, en la University of defence in Belgrade, utilizaron esta herramienta para la detección de vulnerabilidades en su aplicación web, configuraron el proxy respectivo y luego realizaron ataques con la herramienta intruder demostrando la efectividad de estos componentes, pues, lograron descifrar usuarios y contraseñas en los inicios de sesión [20].

Figura 6: Logo de BurpSuite

2.7.3 Wireshark

La herramienta se enfoca al análisis de tráfico de paquetes en tiempo real, proporcionando datos minuciosos como el decifrado de información de estos paquetes, con ello se puede ver credenciales del usuario, es así como, en Wadhwan una ciudad de la India, fue útil para el análisis de vulnerabilidades crónicas, donde se encontraron fugas, permitiendo observar la información confidencial que viaja en la red [21].

Figura 7: Logo de Wireshark

CAPÍTULO III

3 METODOLOGÍA

3.1 Hipótesis

Hipótesis Nula (Ho): La metodología OWASP no permite mejorar la seguridad en el desarrollo de un website para el proceso de votación electrónica.

Hipótesis Alternativa (Ha): La metodología OWASP permite mejorar la seguridad en el desarrollo de un website para el proceso de votación electrónica.

3.2 Identificación de variables

3.2.1 Variable Independiente

Metodología OWASP

3.2.2 Variable Dependiente

Mejora de la seguridad en el website.

3.3 Tipo de Estudio

El tipo de estudio fue cuasi experimental, pues, se requirió evaluar la seguridad en los dos sitios web. Fue prospectivo, porque se tomó mediciones propias, las mismas que se evaluaron de manera transversal, puesto que, la investigación se realizó en un mismo lapso de tiempo. También fue analítico, porque se planteó una hipótesis que fue puesta a prueba.

3.3.1 Enfoque Cuantitativo

Porque a través de los pentesting realizados y votos registrados en los dos websites, se realizó conclusiones generales, sobre la seguridad en los sitios web.

3.3.2 Según la fuente de investigación

Fue una investigación Bibliográfica dado que se recopiló información en libros, revistas científicas y artículos.

3.4 Unidad de análisis

La unidad de análisis fueron los estudiantes de la Carrera de Ingeniería en Tecnologías de la Información de la Universidad Nacional de Chimborazo matriculados en el periodo Mayo – Octubre del 2020.

3.5 Población de estudio

La población de estudio fueron 231 estudiantes de la carrera de TI.

3.6 Tamaño de muestra

El tamaño de la muestra se realizó a través de muestreo aleatorio simple, utilizando un margen de error de 10%, con un nivel de confianza de 90%, obteniendo un total de 53 estudiantes.

3.7 Técnicas de recolección de datos

Técnica de Observación: Se obtuvo los registros de la base de datos de las votaciones realizadas por los estudiantes para luego ser analizada.

3.8 Técnicas de Análisis e interpretación de la información

La investigación se evaluó en base a estadísticas, para ello, se utilizó el software R en la recopilación y análisis de los datos.

De este modo se puso a prueba la hipótesis, en el indicador de disponibilidad se utilizó el Test de Wilcoxon, de igual manera, para medir la confidencialidad, integridad, autenticidad, y confiabilidad se utilizó el Test de Chi Cuadrado.

3.9 Operacionalización de variables

Tabla 1: Operacionalización de variables

Variable	Tipo	Definición Conceptual	Dimensión	Indicadores
Metodología OWASP	Independiente	Metodología de seguridad para el desarrollo de aplicaciones web.	Seguridad	• Confiabilidad

Variable	Tipo	Definición Conceptual	Dimensión	Indicadores
Mejora de la seguridad en el website	Dependiente	Prevención de vulnerabilidades para reducir riesgos.	Fiabilidad	• Tiempo de Disponibilidad • Porcentaje de Integridad • Porcentaje de Confidencialidad • Criterios que garanticen autenticidad

3.10 Procesamiento y Análisis

3.10.1 Desarrollo de los websites

Para el desarrollo de los website se definieron parámetros de la siguiente manera:

- **Website 1:** con la metodología OWASP.
- **Website 2:** de forma vertiginosa.

WEBSITE 1

a. Análisis de requerimientos

El website cumplió los requisitos de acuerdo con el diagrama de flujo presentado a continuación:

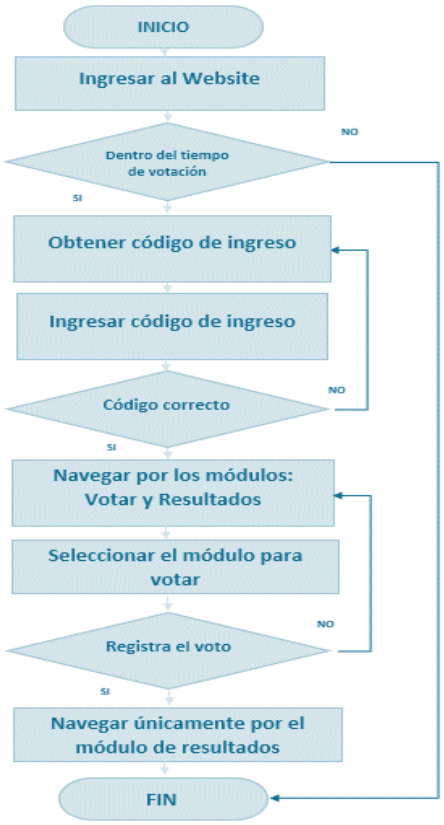


Figura 8: Flujo de procedimiento de votación website 1

Los requisitos funcionales y no funcionales se detallan en el **Anexo 1**.

b. Diseño de Arquitectura

La **Figura 9** muestra la arquitectura utilizada para el desarrollo del website 1, la cual se basó en la arquitectura N-Capas con Orientación al Dominio, por lo tanto las capas necesarias fueron: presentación, aplicación, dominio, e infraestructura, teniendo en cuenta la seguridad en todo el desarrollo.

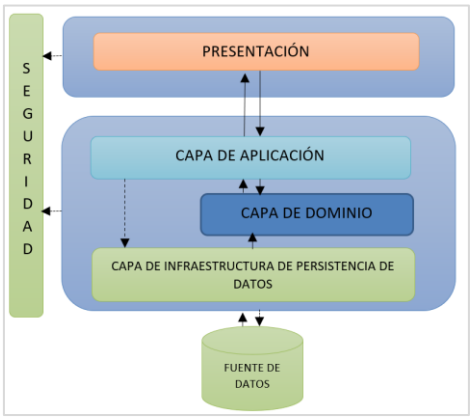


Figura 9: Arquitectura N-Capas con Orientación al Dominio

c. Diagrama de Base de Datos

De acuerdo con los requisitos analizados se creó la base de datos, la misma que se plasma en la **Figura 10**, satisfaciendo los requerimientos del voto electrónico para la elección de Asociación de Estudiantes de la carrera de TI.

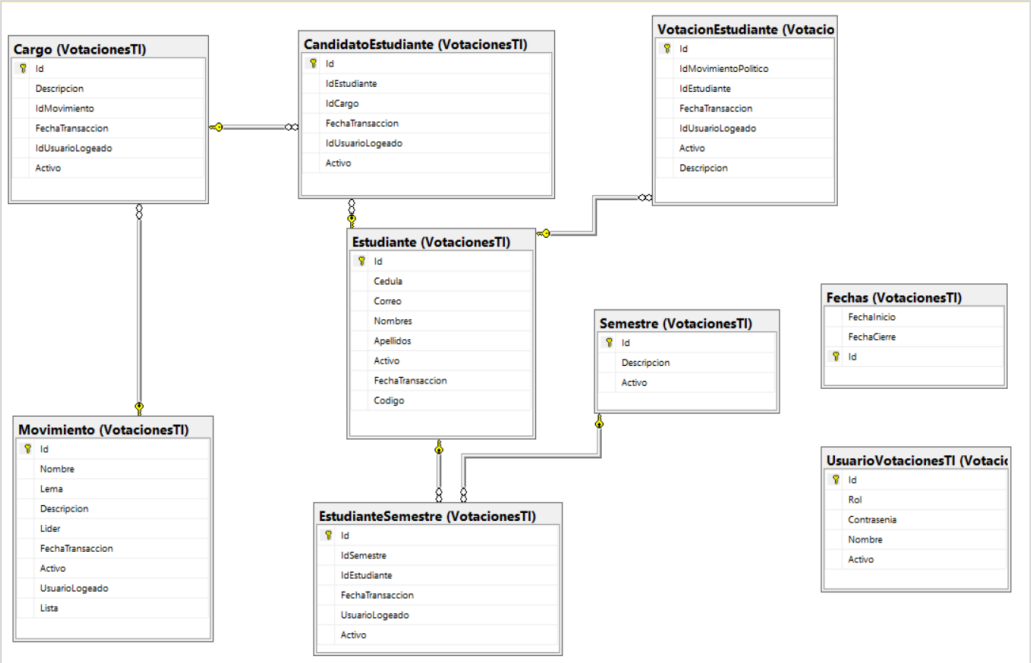


Figura 10: Diagrama de Base de Datos

d. Desarrollo del website aplicando OWASP

El website se desarrolló en Visual Studio con ASP.NET y base de datos SQL Server, de este modo, fueron implementados7 módulos los mismos que se detallan a continuación:

- Ajustes (Configuración del sistema)
- Estudiantes
- Partidos Políticos
- Cargos
- Candidatos
- Votaciones
- Resultados

Finalmente, se analizó y aplicó los criterios de seguridad plasmados en la **Tabla 2** basándose en el Top Ten de OWASP.

Tabla 2: Metodología OWASP en base al caso práctico

OWASP TOP 10 2017	REQUERIMIENTOS PARA EL SISTEMA	CRITERIOS APLICADOS EN EL WEBSITE
Inyección	Evitar que un atacante pueda enviar información maliciosa a través del sistema.	• Se realizó el cifrado de las contraseñas al ingresar al sistema utilizando el algoritmo SHA-256. • Se aplicó certificado SSL para tener conexión segura.
Pérdida de Autenticación	No permitir que el atacante ingrese al sistema con combinaciones de credenciales conocidas.	• Se implementó autenticación doble factor. Primero se registra con el usuario y contraseña, luego el votante recibe en su correo un código de acceso al sistema. • Para la gestión de la aplicación se estableció variables de sesión, las mismas que fueron cifradas con el algoritmo SHA-256.
Exposición de datos sensibles	Prevenir el robo de información confidencial.	• En este ítem el dato más sensible a proteger fue el número de cedula, el cual fue cifrado. • Se eliminó las variables de sesión en la website
Entidades Externas XML (XXE)	Prevenir que realicen ataques de denegación de servicio.	• En el inicio de sesión de la página se implementó un captcha.
Pérdida de Control de Acceso	Evitar que un usuario no autorizado acceda a los distintos módulos.	• Se realizó restricciones, en caso del administrador tuvo acceso a los 7 módulos del sistema, sin embargo, el votante solo los resultados y votar.
Configuración de Seguridad Incorrecta	Evitar mostrar mensajes de error con contenido sensible en texto plano.	• Se utilizó instrucciones try-catch, pues, en caso de existir un error no mostrará el contenido.
Secuencia de Comandos en Sitios Cruzados (XSS)	Evitar que se envié datos a través de un navegador sin validar correctamente.	• Se realizó la validación de datos, en el caso de los votantes se validó la cedula y correo electrónico.
Deserialización Insegura	Impedir un ataque de denegación de servicio (DoS) y que los datos no confiables fuercen el correcto funcionamiento de la aplicación.	• Se utilizó el captcha para evitar ataques DoS.
Componentes con vulnerabilidades conocidas	Evitar el uso de componentes de origen no oficial en la aplicación.	• Se utilizó componentes necesarios de origen oficial, pues, para el sistema se descargaron paquetes Nuggets de Microsoft Visual Studio, es decir no se obtuvieron de otro sitio web.

OWASP TOP 10 2017	REQUERIMIENTOS PARA EL SISTEMA	CRITERIOS APLICADOS EN EL WEBSITE
Registro y Monitoreo Insuficientes	Verificar que acciones incorrectas y sospechosas se están registrando.	• Se generaron Logs para el registro de fallos en el inicio de sesión y funciones que realiza el sistema.

WEBSITE 2

a. Análisis de Requerimientos

El website desarrollado de forma vertiginosa (website 2) cumple los mismos requerimientos, sin embargo, corresponde otro diagrama de flujo, pues, no fue desarrollado con criterios que garanticen seguridad.

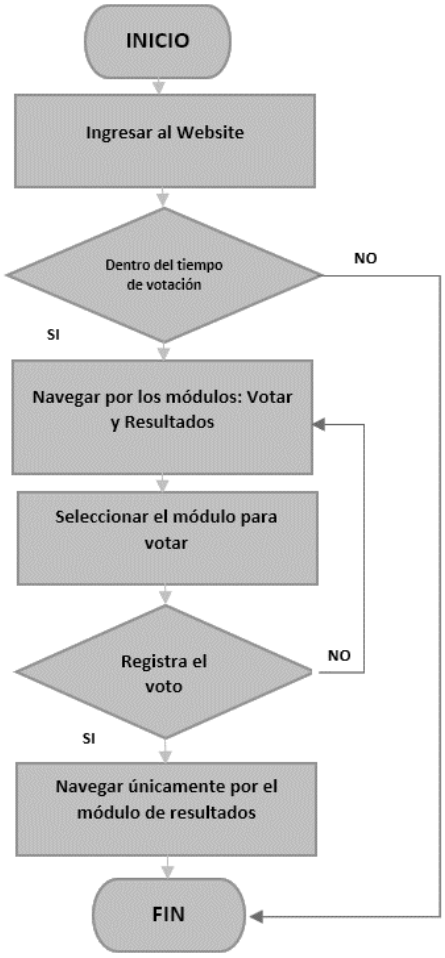


Figura 11: Flujo de procedimiento de votación website 2

b. Diseño de Arquitectura

El diseño de arquitectura utilizado en el website 2 constaron de 4 capas: presentación, lógica de negocios, acceso a datos, y base de datos, sin embargo no se considera la seguridad.

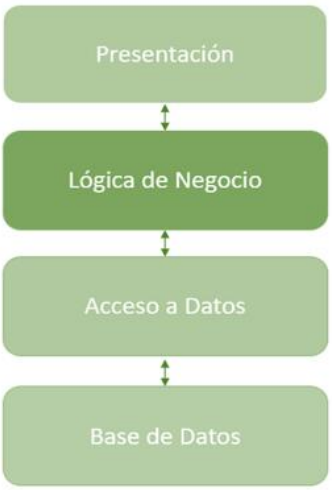


Figura 12: Arquitectura N-Capas

c. Diagrama de Base de Datos

Se utilizó el mismo diagrama de base de datos que en el website 1, como se muestra en la **Figura 10**.

d. Desarrollo del website de forma vertiginosa

De igual forma, se desarrolló en Visual Studio, con ASP.NET y SQL Server para la base de datos y se implementaron los 7 módulos correspondientes.

3.10.2 Pentesting en los dos websites

Los pentesting fueron realizados en Kali Linux en base a los indicadores planteados: disponibilidad, integridad, confidencialidad y autenticidad.

SlowHTTPTest: con la herramienta se realizaron ataques DoS lo que permitió medir la *disponibilidad* de los websites.

BurpSuite se configuró el proxy respectivo, así pues, se realizó las interceptaciones de acceso a diferentes módulos de acuerdo con los permisos establecidos, de esta forma se midió la *confidencialidad*.

Además, se utilizó la misma herramienta, para medir la *integridad*, pues, una vez accedido a páginas no autorizadas se procedió a modificar la información obtenida.

Para finalizar, se midió el indicador de *autenticidad*, para ello, en el website1 se implementaron los siguientes criterios:

- **Credenciales únicas:** se estableció el número de cedula para el ingreso al website.
- **Acceso con código único:** el votante accede con un código que es proporcionado a su correo electrónico.
- **Escoger entre roles:** en el inicio de sesión el votante debe seleccionar el tipo de usuario.
- **Captcha:** con el captcha se garantiza que el votante sea una persona y no un robot.
- **Registro del voto únicamente del usuario estudiante:** para el sufragio el website comprueba el tipo de usuario, si no es un usuario estudiante (votante), el sistema no registra el voto.
- **Protocolo https:** permite que la información viaje cifrada, y las credenciales no sean sustraídas.

Con estos criterios se garantizó que el votante no sea un impostor, pues, en las votaciones es primordial que no exista suplantación de identidad.

Por el contrario, en el website 2 solo se aplicó:

- El ingreso con el número de cedula como credencial única
- Selección de rol en el Login.

Además, se publicó bajo el protocolo http, permitiendo que se visualice las credenciales del votante y cualquier usuario pueda hacer uso de ella. Para comprobar aquello se utilizó la herramienta Wireshark, que permitió analizar el tráfico que transita en la red en el cual se pudo capturar el usuario y contraseña del usuario logeado.

3.10.3 Registro y certificado de votación

Los websites fueron publicados en un Servidor Virtual Privado (VPS) en Azure, de este modo los estudiantes de la carrera de Ingeniería en Tecnologías de la Información de la UNACH accedieron a los websites y registraron su voto, así pues, obtuvieron su respectivo certificado de votación enviado a su correo electrónico institucional.

CAPÍTULO IV

4 RESULTADOS Y DISCUSIÓN

4.1 Pentesting: Website 1

4.1.1 Disponibilidad

Para medir la disponibilidad, se realizó ataque DoS con conexiones mal formadas, a la página de Login, pues, la página de inicio es la más crítica.

Para la ejecución de ataques DoS se utilizó la herramienta SlowHTTPTest ejecutando el siguiente comando:

slowhttptest -c 1000 -X -g -o slowhttp_login -i 10 -r 200 -t GET -u ttps://votacionesti2.tk/Sandbox/Loguin1/Loguin -x 24 -p 3

Los parámetros y significado de dicho comando se ven plasmados en la **Tabla 3**.

Tabla 3: Parámetros establecidos con la herramienta slowhttptest

Parámetro	Significado	Valor
-c	Número de ataques	1000
-X	Modo de prueba	Slow Read
-g, -o	Genera estadísticas, y guarda la salida en un archivo	slowhttp_login
-i	Intervalo de tiempo	10
-r	Velocidad de conexiones	200
-t	Tipo de solicitud	GET
-u	Url de destino	https://votacionesti2.tk Sandbox/Loguin1/Loguin
-x	Longitud máxima (en bytes)	24
-p	Segundos de tiempo de espera para esperar la respuesta HTTP	3

Figura 13: Ejecución de comando para ataque al website 1

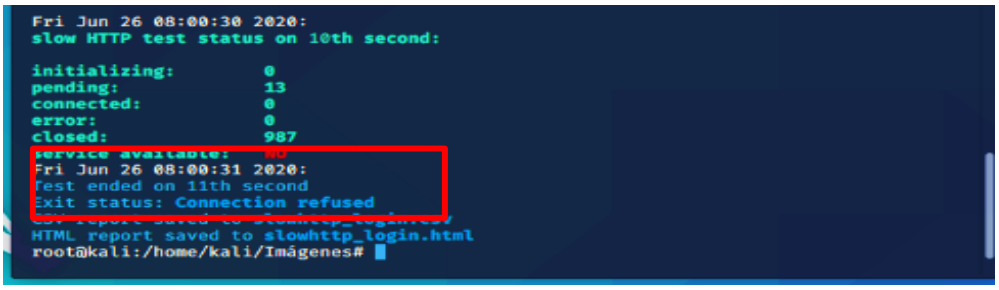


Figura 14: Disponibilidad del website 1

La **Figura 14** muestra el tiempo de respuesta de las 1000 conexiones realizadas por el ataque, el cual finalizó en 11 segundos y se observó que la conexión fue rechazada, es decir, no aceptó las conexiones mal formadas que se envió con la herramienta SlowHTTPTest.

De igual manera se realizó el mismo procedimiento en 30 periodos de tiempo, los resultados se visualizan en la **Tabla 19** del **Anexo 2**.

4.1.2 Confidencialidad

Para comprobar la confidencialidad se puso a prueba el nivel acceso a los módulos, pues solo el usuario administrador tiene acceso a todos los módulos (Ajustes, Estudiantes, Partidos Políticos, Cargos, Candidatos, Votaciones y Resultados), sin embargo, el votante únicamente tiene acceso a los módulos de Votaciones y Resultados.

El pentesting se realizó con la herramienta BurpSuite interceptando peticiones (url) del usuario, los pasos se detallan a continuación:

Paso 1: Iniciar sesión como usuario estudiante (votante).

Figura 15: Inicio de sesión en el website con la metodología OWASP

Figura 16: Código para ingresar al website

Figura 17: Página principal del usuario Estudiante

La **Figura 17** muestra la página principal del website con los módulos permitidos para el votante.

Paso 2: Abrir la herramienta BurpSuite

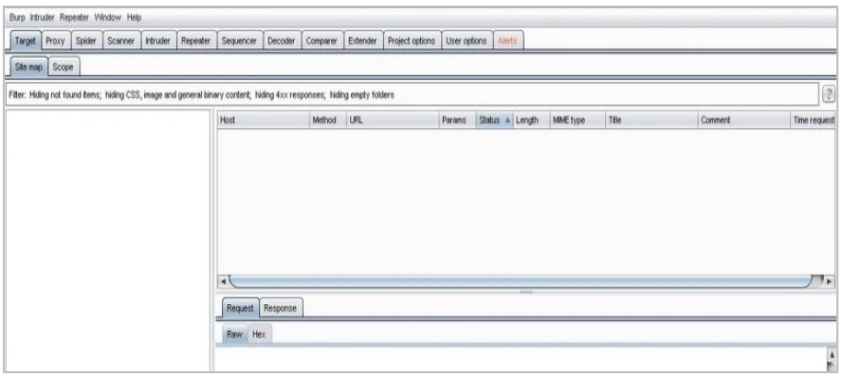


Figura 18: Inicio de la herramienta BurpSuite

Paso 3: Dar clic en el módulo de votaciones para obtener la url.

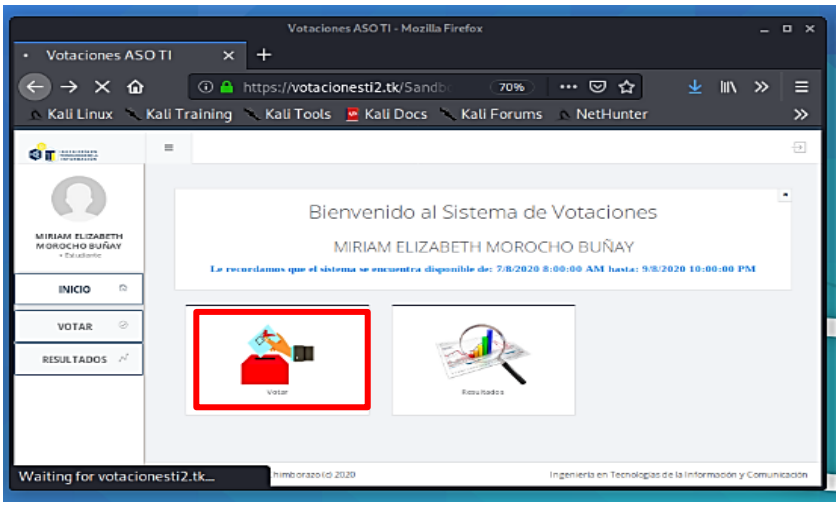


Figura 19: Petición del estudiante al módulo de votar en el website 1

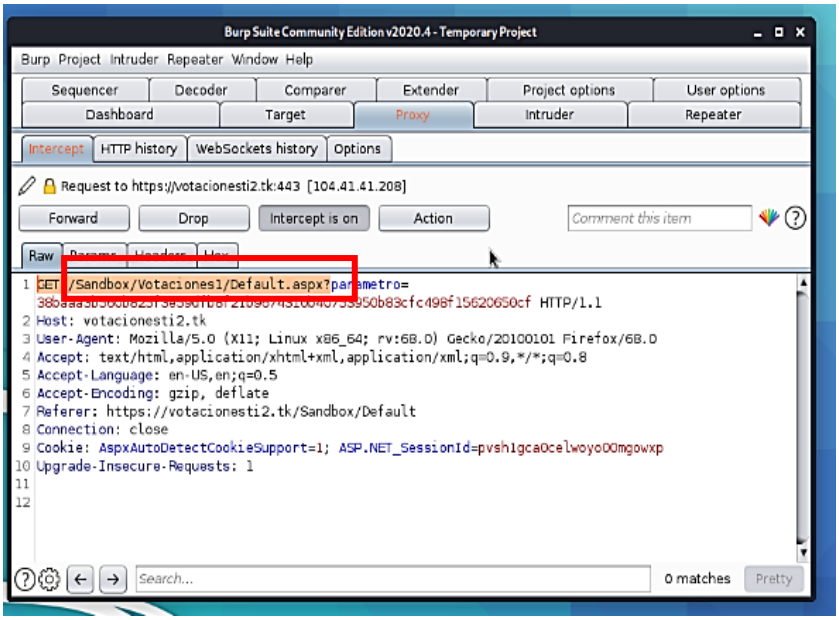


Figura 20: Herramienta BurpSuite con la url peticionada

La **Figura 20** muestra la url peticionada por el estudiante(votante) la misma que fue interceptada con la herramienta BurpSuite.

Paso 4: Editar la url interceptada para redireccionarle al módulo de Ajustes.

Figura 21: Edición de la url en la herramienta BurpSuite

Paso 5: Cerrar la herramienta BurpSuite.

Como se puede ver en el sitio web, la petición que se editó fue redireccionada a una página Oops, pues, el sistema generó un error de acceso a un módulo no permitido, en efecto, el website 1 si controló esta vulnerabilidad y cuando esto sucede, los códigos de inicio de sesión se eliminan, por lo tanto, el usuario deberá iniciar sesión nuevamente, dando clic en un botón como se observa en la **Figura 22**.

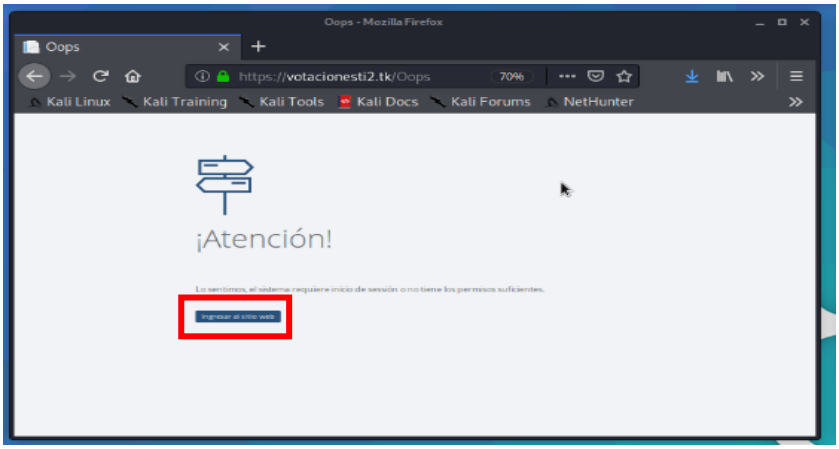


Figura 22: Página de acceso denegado en la interceptación

Para verificar que un votante no tenga acceso a módulos no permitidos, se realizó los mismos pasos con los módulos restantes, los resultados obtenidos se muestran en la **Tabla 20** del **Anexo 2**.

4.1.3 Integridad

Para medir la integridad se puso a prueba que un usuario no permitido no deba alterar información en los módulos, así pues, en el website 1 un votante no pudo editar información del módulo de Ajustes, pues, anteriormente en la confidencialidad se les negó el acceso a este módulo, sin embargo, al realizar pentesting hacia los demás módulos se logró interceptar el módulo de Candidatos, el mismo que resulto ser vulnerable a modificación de la información, los resultados obtenidos hacia los demás módulos se visualizan en la **Tabla 21** del **Anexo 2**.

4.1.4 Autenticidad

Se utilizó la herramienta Wireshark permitiendo verificar que la información confidencial ingresada en el sitio web no sea visible para los demás, de tal forma que las credenciales no sean sustraídas y por ende no puedan acceder personas impostoras.

Al navegar en el website 1 se capturó los paquetes del tráfico de red, se utilizó el comando *urlencoded-form* para visualizar paquetes que contengan información de credenciales, así

pues, al ser implementado con protocolo https no se observa que las credenciales hayan sido capturadas como se muestra en la **Figura 23**.

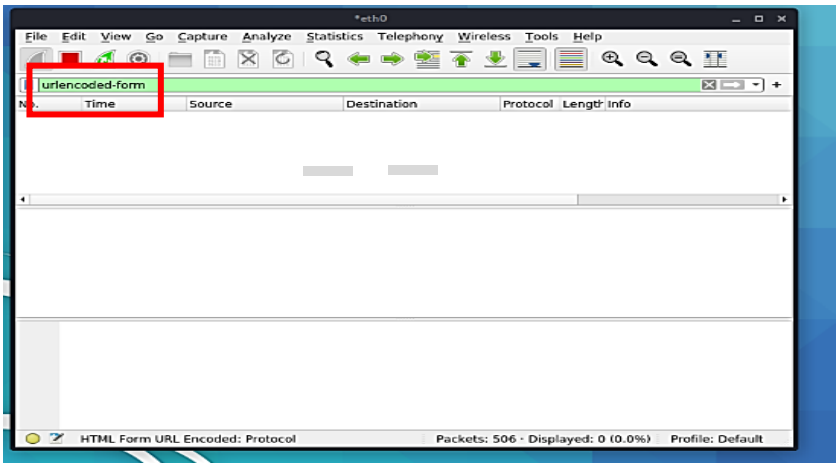


Figura 23: Ataque negado para visualizar las credenciales del website 1

4.2 Pentesting: Website 2

4.2.1 Disponibilidad

Para el pentesting del website 2 se ejecutó el comando con los mismos parámetros que en el website 1, como se visualiza en la **Tabla 3**. En este apartado se modificó la url hacia dónde va dirigido el ataque.

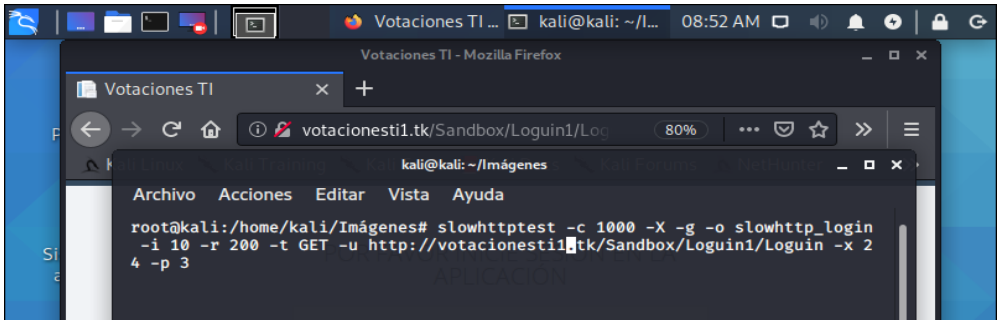


Figura 24: Ejecución de comando para ataque al website 2

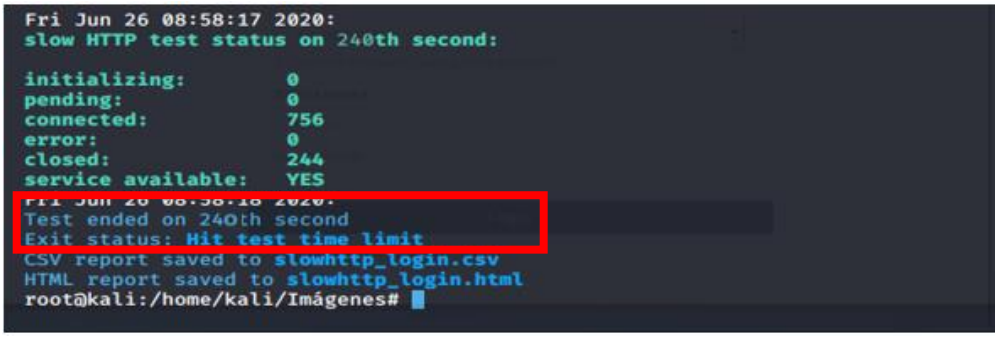


Figura 25: Disponibilidad del website 2

La **Figura 25** muestra el tiempo en que finalizó el ataque, el cual terminó en 240 segundos alcanzando su límite de tiempo, es decir, el website 2 procesó las conexiones mal

formadas que se envió con la herramienta SlowHTTPTest, por lo tanto, en estos 240 segundo el sistema tardó más tiempo en resolver las peticiones.

De igual forma, se realizó este mismo procedimiento en 30 periodos de tiempo, los resultados de los tiempos obtenidos se observan en la **Tabla 19** del **Anexo 2**.

4.2.2 Confidencialidad

Del mismo modo que en el website 1, se puso a prueba el nivel acceso a los módulos iniciando sesión con credenciales de un estudiante (votante), a continuación se detallan los pasos.

Paso 1: Ingreso al website con las respectivas credenciales, y abrir la herramienta BurpSuite.

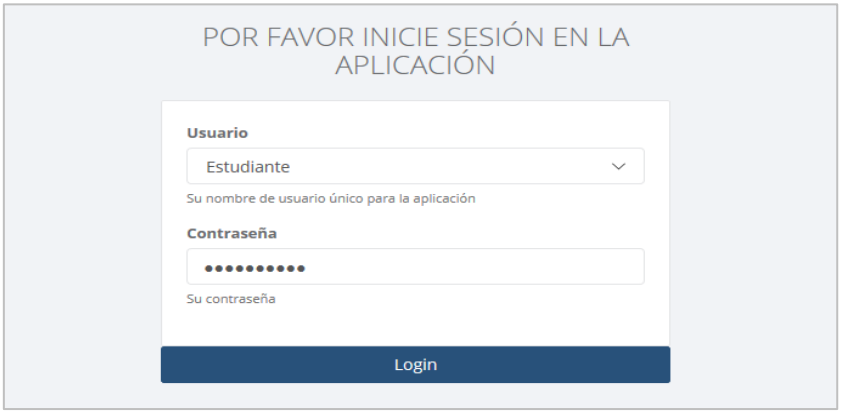


Figura 26: Inicio de sesión al website desarrollado de forma vertiginosa

Paso 2: Dar clic en el módulo de votaciones para obtener la url.

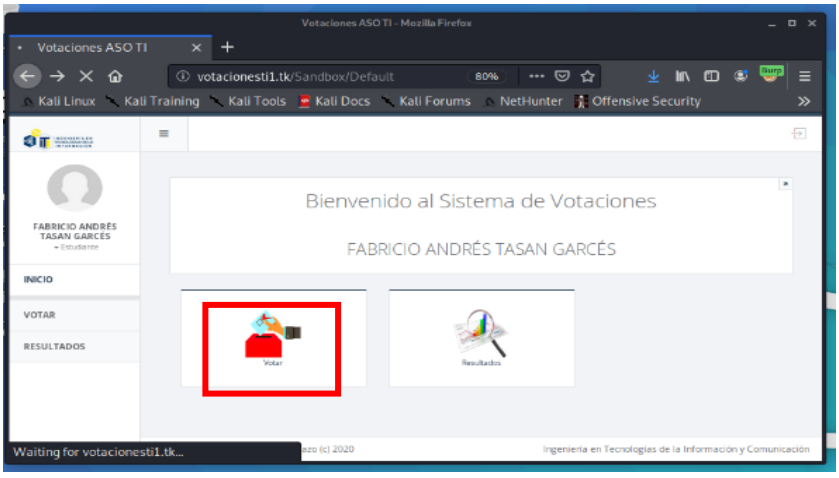


Figura 27: Petición del estudiante al módulo de votar en el website 2

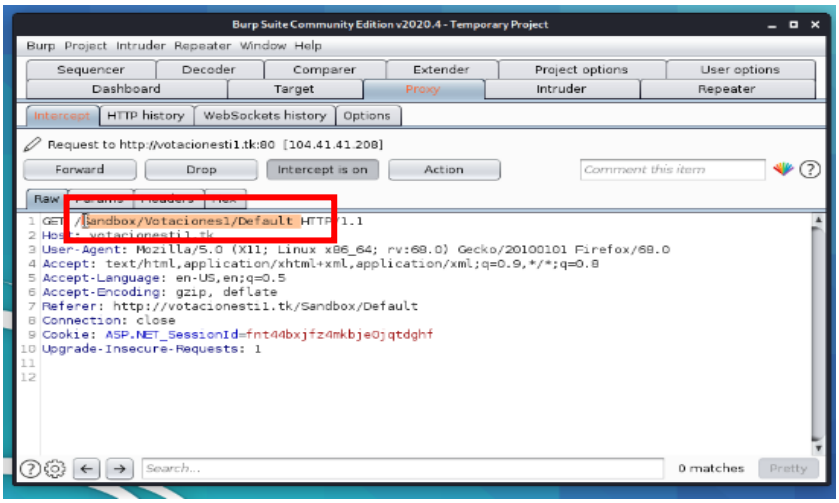


Figura 28: Url peticionada por un votante

La **Figura 28** muestra la url peticionada por el estudiante la misma que fue interceptada con la herramienta BurpSuite.

Paso 3: Editar la url redireccionándole al módulo de Ajustes.

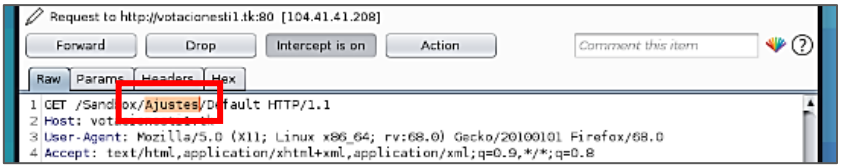


Figura 29: Edición de la url del website 2

Paso 4: Cerrar la herramienta BurpSuite y visualizar el sitio web.

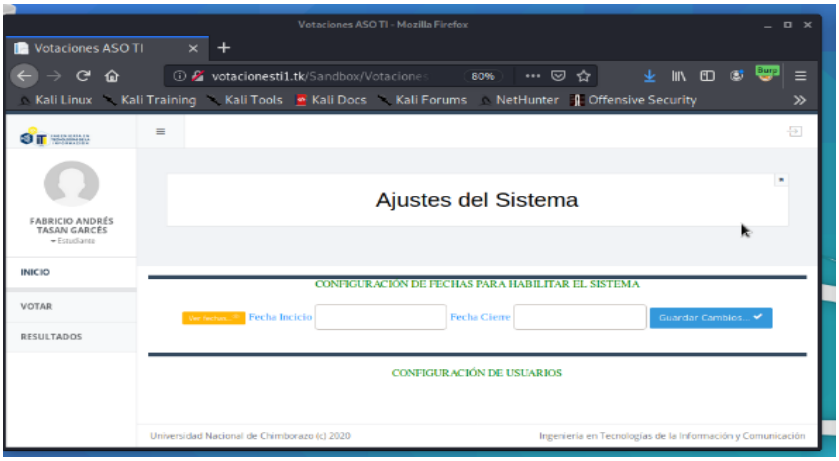


Figura 30: Interceptación exitosa en el website 2

La **Figura 30** muestra la interceptación exitosa a un módulo en el cual un votante no tenía acceso, de esta forma se demuestra que en el website 2 existe una vulnerabilidad no controlada.

De igual manera se evaluaron los demás módulos del website para verificar si se interceptan o no, los resultados obtenidos se visualizan en la **Tabla 20** del **Anexo 2**.

4.2.3 Integridad

Una vez vulnerado la confidencialidad se pudo acceder a los distintos módulos, de tal forma el votante pudo navegar por los demás módulos digitando directamente la url.
Así pues, se procede a modificar o alterar la información del módulo de Ajustes como se visualiza a continuación:

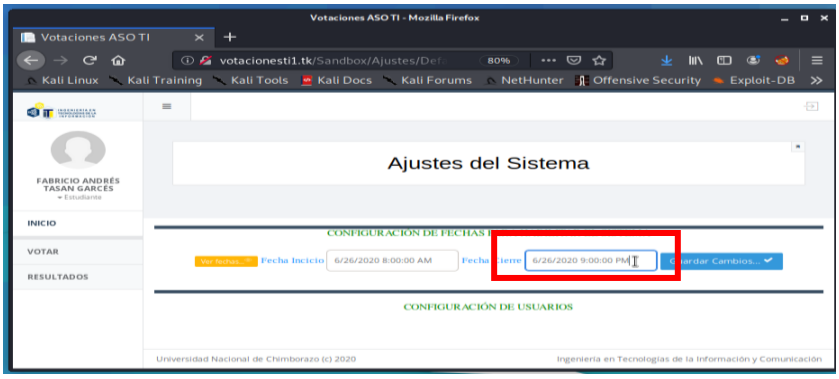


Figura 31: Modificación en las fechas de votación del website 2

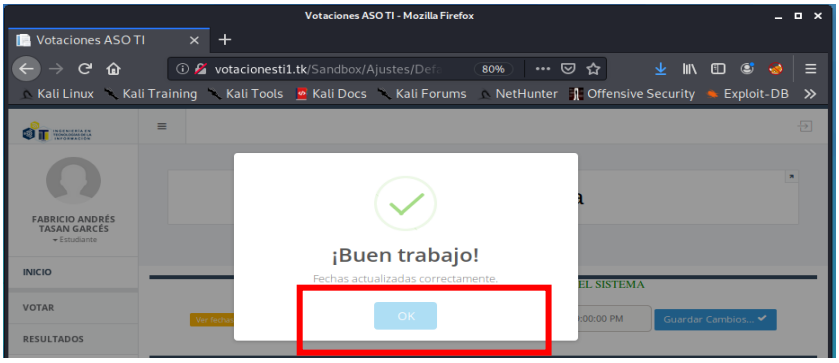


Figura 32: Modificación exitosa de un usuario no permitido del website 2

La **Figura 32** muestra la modificación exitosa por parte de un usuario no permitido, sin embargo solo el administrador debe realizarlas, de este modo se demostró que en el website desarrollado vertiginosamente no se controla esta vulnerabilidad.
De igual forma se realizó este proceso para verificar que módulos tienen esta vulnerabilidad, los resultados se plasman en la **Tabla 21** del **Anexo 2**.

4.2.4 Autenticidad

El website 2 fue desplegado en el servidor con protocolo http, teniendo la vulnerabilidad que la información proporcionada en el sitio web sea sustraída. Para comprobar aquello se realizó un pentesting con la herramienta Wireshark capturando paquetes del tráfico de red.

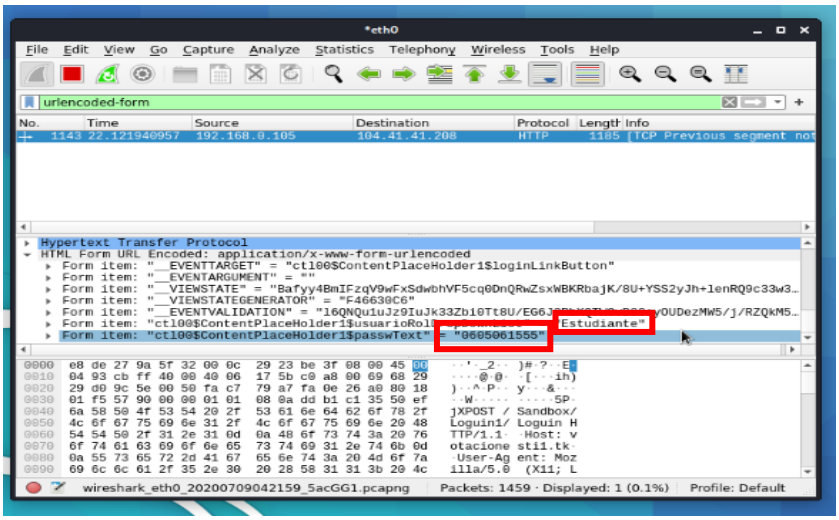


Figura 33: Visualización de credenciales en el website 2

La **Figura 33** muestra los resultados obtenidos al ejecutar el comando *urlencoded-form* el cual permitió verificar que en uno de sus paquetes contenía el usuario y contraseña digitados anteriormente al iniciar sesión en el sitio web.

4.3 Resultados Obtenidos de los Pentesting

4.3.1 Tiempo de Disponibilidad

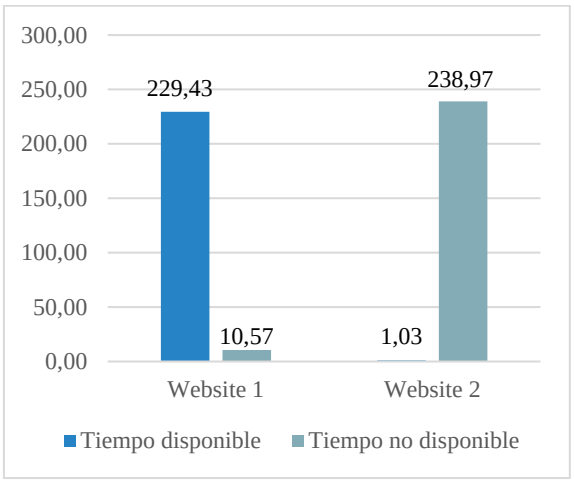


Figura 34: Resultado de disponibilidad de los websites

La **Figura 34** detalla el promedio de los tiempos de disponibilidad de los dos websites, mencionados en la **Tabla 19** del **Anexo 2**, en el ataque configurado con 1000 conexiones en un tiempo de 240 segundos, el website 1 mostró un tiempo promedio de 229,43 segundos de disponibilidad, mientras que, el website 2 mostró 1,03 segundos, es decir, el website 1 rechazó las conexiones mal formadas que se realizó con el ataque, sin embargo, el website 2 procesó estas conexiones, por lo tanto, fue más susceptible a ataques DoS.

4.3.2 Porcentaje de Confidencialidad

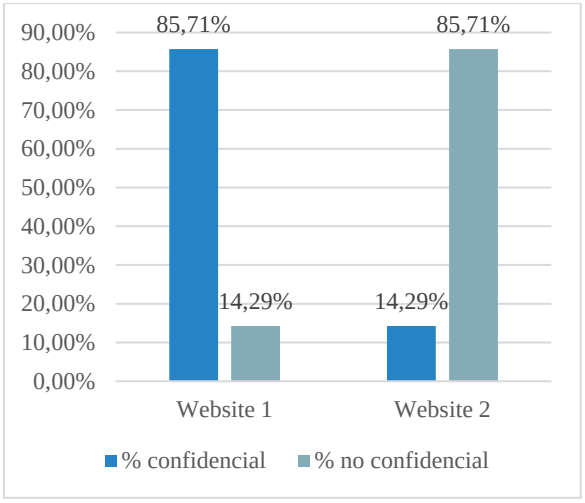


Figura 35: Resultados de confidencialidad de los websites

La **Figura 35** plasma el porcentaje de confidencialidad de los dos websites, en base a la información de la **Tabla 20** del **Anexo 2,** a través de la herramienta BurpSuite en el website 1 solo se logró interceptar un módulo y los 6 restantes no, por lo tanto existió un 85,71% de confidencialidad.

Por el contrario, el website 2 obtuvo el 14,29% de confidencialidad, pues, se logró interceptar 6 módulos de los 7 existentes, es decir, existió mayor vulnerabilidad porque se pudo acceder a información no autorizada.

4.3.3 Porcentaje de Integridad

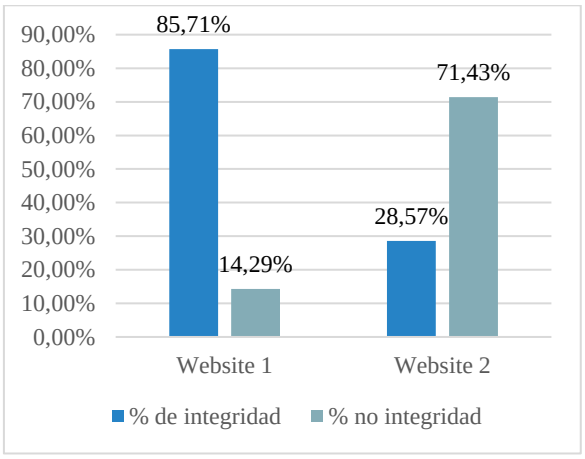


Figura 36: Resultados de integridad de los websites

La **Figura 36** muestra los porcentajes de integridad de los dos websites de acuerdo con los resultados plasmados en la **Tabla 21** del **Anexo 2.**

El website 1 muestra un porcentaje de 85,71% de integridad, pues de los 7 módulos solo uno resulto ser susceptible a modificaciones o alteraciones de información, por el contrario, el website 2 tiene un 28,57% porque 5 de los 7 módulos fueron vulnerados, por lo tanto se verifica que el website 1 tiene mayor porcentaje de protección en los datos que el website 2.

4.3.4 Criterios que garanticen Autenticidad

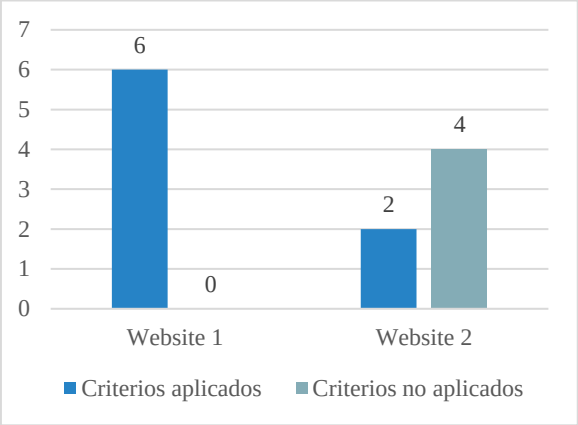


Figura 37: Resultados de autenticidad de los websites

La **Figura 37** muestra los criterios implementados en los websites para garantizar la autenticidad de los usuarios, los mismos que se detallan en la **Tabla 22** del **Anexo 2**.

De acuerdo con la metodología OWASP se analizaron 6 criterios para la autenticación de usuarios, los mismos que fueron implementados en el website 1, caso contrario, en el website 2 solo se implementaron dos de los 6 criterios, pues, fue desarrollado de forma vertiginosa. Por lo tanto, el website 1 tuvo más posibilidad que los que registren el voto sean estudiantes y no impostores con credenciales hurtadas o sustraídas.

4.3.5 Tabla resumen de los pentesting

Tabla 4: Tabla resumen de los pentesting

Indicadores	**Website con la metodología OWASP**	**Website desarrollado vertiginosamente**
Disponibilidad	95,60 %	0,43 %
Confidencialidad	85,71 %	14,29 %
Integridad	85,71 %	28,57 %
Autenticidad	100 %	33,33 %
Total	**91, 75 %**	**19,15 %**

La **Tabla 4** muestra el porcentaje de seguridad de los websites, basándose en los resultados de cada indicador plasmados en el **Anexo 2**.

El website desarrollado con la metodología OWASP tuvo un 91,75% de seguridad, sin embargo el website desarrollado vertiginosamente solo alcanzó un 19,15% de seguridad, es decir, en el website 2 se identificaron más fallos de seguridad que son consecuencia de vulnerabilidades no controladas.

4.4 Resultados de votaciones

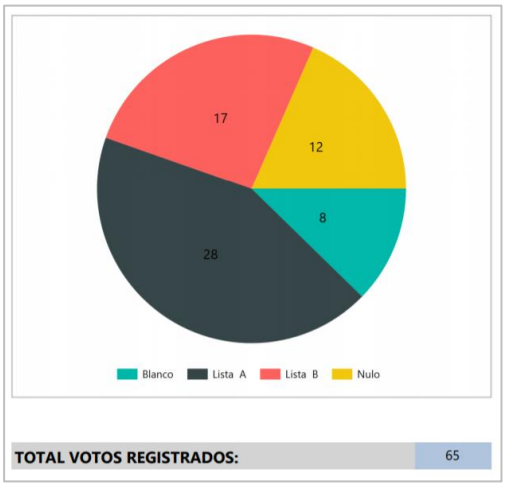


Figura 38: Resultados de votaciones del website 1

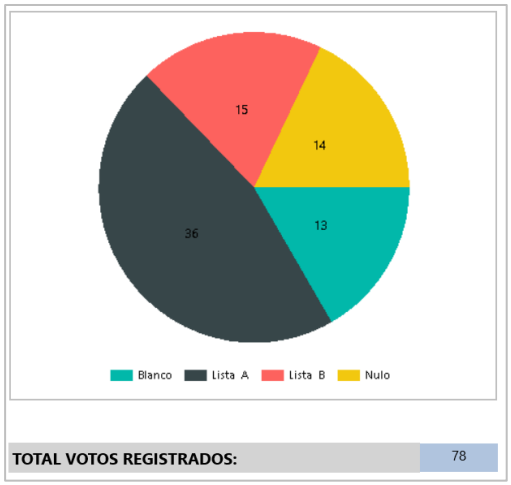


Figura 39: Resultados de votaciones del website 2

La **Figura 38** y **Figura 39** muestra los resultados obtenidos de las votaciones realizadas por 65 estudiantes de la carrera de Tecnologías de la Información de la Universidad Nacional de Chimborazo, como se puede observar en el website 1 se contabilizó 65 votos correspondientes al total de votantes, sin embargo, en el website 2 se contabilizó 78 votos, pues, el sistema fue vulnerable y algunos votantes sufragaron más de una vez permitiendo que haya fraude en los resultados de la votación.

4.5 Comprobación de hipótesis por indicador

Para comprobar la hipótesis se utilizó los datos obtenidos en los pentesting realizados, los mismos que son mencionados en la **Tabla 19**, **Tabla 20**, **Tabla 21**, **Tabla 22** y **Tabla 23** del **Anexo 2**.

Se puso a prueba la hipótesis nula y la hipótesis alternativa, para ello, se aplicó Test de Wilcoxon y Test de Chi Cuadrado con el software R.

4.5.1 Indicador: Tiempo de Disponibilidad

Se comparó los tiempos de disponibilidad obtenidos en los ataques realizados al website 1 como al website 2, plasmados en la **Tabla 19** del **Anexo 2**, en primer lugar, se aplicó un test de normalidad a las variables (TiempoDisponible_w1 y TiempoDisponible_w2), con un p-valor menor al 5% en las dos variables como se visualiza en el **Anexo 4**, el test de Shapiro-Wilk afirmó que dichas variables no se distribuyen normalmente, por lo tanto no se pudo utilizar test paramétricos para la comparación de los tiempos.

Por tal razón se utilizó un test no paramétrico de Wilcoxon para datos emparejados, en el cual se obtuvo los siguientes datos.

Tabla 5: Resultados del Test de Wilcoxon del indicador disponibilidad

V	496
p-valor	0.0000009443

La **Tabla 5** plasma el resultado del test, con un estadístico V= 496 y un *p-valor* menor que el 5 % (0.05) se afirma que los tiempos del website 1 comparado con el website 2 son diferentes , es decir, existe una diferencia significativa al 95% de confianza, por lo tanto, se rechaza la hipótesis nula (Ho) y la hipótesis alternativa (Ha) se valida, lo cual significa que el website 1 es más seguro que el website 2 con respecto a la disponibilidad del servicio, pues, el website 1 fue menos vulnerable a ataques DoS.

4.5.2 Indicador: Porcentaje de Confidencialidad

Se analizó los datos obtenidos en el pentesting realizado a cada uno de los módulos visualizados en la **Tabla 20** del **Anexo 2**, en este apartado, se aplicó el Test de Chi Cuadrado, puesto que se trata de variables cualitativas y con el software estadístico R se obtuvo los siguientes resultados.

Tabla 6: Resultados del Test de Chi Cuadrado del indicador confidencialidad

X-squared	7.1429
p-valor	0.007526

La **Tabla 6** muestra el valor estadístico de 7.1429 y un p-valor de 0.007526, siendo este menor al 5% se rechaza la hipótesis nula (Ho) y se valida la hipótesis alternativa (Ha), así pues, se comprueba que el website 1 es más seguro que el website 2 con respecto a la confidencialidad de sus módulos.

4.5.3 Indicador: Porcentaje de Integridad

Se comparo los resultados obtenidos del pentesting realizado al website 1 como al website 2 plasmados en la **Tabla 21** del **Anexo 2,** de igual manera, haciendo uso del software estadístico R y aplicando el Test de Chi Cuadrado se obtuvo los siguientes datos.

Tabla 7: Resultados del Test de Chi Cuadrado del indicador integridad

X-squared	4.6667
p-valor	0.03075

La **Tabla 7** plasma el resultado del test, donde el valor estadístico es de 4.6667 y un *p-valor* de 0.03075 el mismo que es menor al 5%, por ende se rechaza la hipótesis nula (Ho) y se valida la hipótesis alternativa (Ha), determinando que el website 1 es más seguro que el website 2 con respecto a la protección de datos al mantenerlos íntegros en cada uno de sus módulos.

4.5.4 Indicador: Criterios que garanticen autenticidad

Se analizó los criterios establecidos en el website 1 como en el website 2 como se muestra en la **Tabla 22** del **Anexo 2**, haciendo uso del software estadístico R y aplicando Test de Chi Cuadrado se obtuvo los siguientes datos.

Tabla 8: Resultados del Test de Chi Cuadrado del indicador autenticidad

X-squared	6
p-valor	0.01431

La **Tabla 8** plasma el resultado del test, con un valor estadístico de 6 y *p-valor=0.01431* siendo este menor a 0.05, se rechaza la hipótesis nula (Ho) y se aprueba la hipótesis alternativa (Ha), pues, existe diferencia significativa entre los dos websites, de este modo, se determina que el website 1 es más seguro que el website 2 con respecto a la autenticidad, asegurando que el votante sea un estudiante y no un impostor.

4.5.5 Indicador: Confiabilidad

Para la comprobación de hipótesis de éste indicador se basó en datos de la **Tabla 23** del **Anexo 3** se realizó el Test de Chi Cuadrado y se obtuvieron los siguientes resultados.

Tabla 9: Resultados del Test de Chi Cuadrado del indicador confiabilidad

X-squared	11.917
p-valor	0.0005563

La **Tabla 9** plasma el resultado del test, con valor estadístico de 11.917 y con un *p-valor* menor al 5% se rechaza la hipótesis nula (Ho) y se aprueba la hipótesis alternativa (Ha), comprobando que el website 1 es más seguro que el website 2 con respecto a la confiabilidad de los resultados obtenidos en las votaciones registradas.

En resumen, al evaluar cada indicador se verificó que el website desarrollado con la metodología OWASP obtuvo mayor porcentaje de seguridad que el website desarrollado vertiginosamente.

CONCLUSIONES

- La seguridad en los websites debe ser considerado como un requisito mas no como una opción, por tal motivo, OWASP como metodología ofrece una guía de desarrollo conocida como OWASP TOP 10, en la que menciona una lista de vulnerabilidades críticas y frecuentes, de este modo, provee criterios, sugerencias y técnicas primordiales que se deben aplicar en todo el ciclo de vida del proyecto, basándose en la prevención de riegos y vulnerabilidades que puedan afectar al sistema de información.
- A través de la metodología OWASP se logró mayor nivel de seguridad para el website de votación electrónica, pues, se aplicaron criterios basándose en su *Top Ten* de detección de vulnerabilidades haciendo énfasis en aspectos fundamentales que garantizaron: disponibilidad, confidencialidad, integridad, autenticidad y confiabilidad, a diferencia del website desarrollado vertiginosamente que solo se enfocó en la funcionalidad mas no en la seguridad, por ende existió más vulnerabilidades.
- Al realizar los pentesting con Kali Linux se identificó vulnerabilidades en los dos websites, como resultado se obtuvo que el website desarrollado con la metodología OWASP es menos susceptible ante ataques informáticos, pues alcanzo un 91,75% de seguridad a diferencia del website desarrollado vertiginosamente que logró solo un 19,15% de seguridad.
- El website con la metodología OWASP brinda mayor seguridad para el proceso de votaciones, pues, al analizar los resultados se contabilizó 65 votos correspondientes al total de votantes, a diferencia del website desarrollado vertiginosamente que se contabilizo 78 votos, es decir, existieron votantes que sufragaron más de una vez.

RECOMENDACIONES

- Utilizar la metodología OWASP para el desarrollo de sitios web, puesto que, brinda directrices para la detección y prevención de vulnerabilidades, los mismos que se pueden implementar tanto en software libre como privado.
- Analizar los criterios de la metodología OWASP Top Ten, para luego implementarlos de forma correcta basándose en los requerimientos del sistema de votación electrónica.
- Aplicar estándares de seguridad en websites de acuerdo con las nuevas tendencias y tecnologías que van mejorando en el transcurso del tiempo.
- Utilizar Kali Linux para realizar pentesting, pues, incorpora herramientas muy útiles que permiten determinar las vulnerabilidades existentes en los sitios web.

BIBLIOGRAFÍA

[1] N. Lagreca, «MODELO DE AUDITORIA PARA SERVICIOS TELEMÁTICOS DE LA UNIVERSIDAD SIMÓNBOLÍVAR.,» *Télématique,* pp. 79-95, 2017.

[2] J. Altamirano , «Políticas de Seguridad de la Información: Revisión Sistemática de las Teorías que Explican su Cumplimiento,» *REVISTA IBÉRICA DE SISTEMAS E TECNOLOGIAS DE INFORMAÇÃO* , pp. 112-134, 2017.

[3] S. Quiroz y D. Macías, «Seguridad en informatica: consideraciones,» *DOMINIO DE LAS CIENCIAS,* pp. 676-688, 2017.

[4] O. W. A. S. P. OWASP, «Fundación OWASP,» 18 11 2017. [En línea]. Available: https://wiki.owasp.org/images/5/5e/OWASP-Top-10-2017-es.pdf.

[5] H. R. González y R. Montesino, «Capacidades de las metodologías de pruebas de penetración para detectar vulnerabilidades frecuentes en aplicaciones web,» *Revista Cubana de Ciencias Informáticas,* pp. 52-65, 2018.

[6] J. S. Peralta y J. Gutiérrez, «Hacia el desarrollo de un prototipo de sistema de voto electrónico para Costa Rica,» *Tecnología en Marcha,* pp. 146-158, 2016.

[7] A. Hernández y J. Porven, «Procedimiento para la seguridad del proceso de despliegue de aplicaciones web,» *Revista Cubana de Ciencias Informáticas,* pp. 42-56, 2016.

[8] Y. La Rosa, «Seguridad web/ Web security,» *TINO,* p. 28, 2016.

[9] V. M. Morales, «Amenazas y vulnerabilidades de los sistemas de,» *Elecciones(ONPE),* pp. 119-136, 2014.

[10] G. Vega y R. A. Ramos, «Vulnerabilidades y amenazas a los servicios web de la intranet de la Universidad Técnica de Babahoyo,» *3C Tecnología,* pp. 53-66, 2017.

[11] C. Vidal, L. López, J. Rojas y M. Castro, «Web System Development for Competences Personnel Selection of Directives by PHP CodeIgniter 3.0,» *CIT-Centro de Información Tecnológica,* pp. 2-9, 2017.

[12] F. G. García, J. M. Contento, M. P. Zea y J. R. Molina, «Metodologías de desarrollo en aplicaciones web,» *3C Tecnología,* pp. 54-71, 2017.

[13] A. L. Hernández Saucedo y J. Mejia Miranda, «Guía de ataques, vulnerabilidades, técnicas y herramientas para aplicaciones Web,» *Revista Electronica de Computación, Informática, Biomédica y Electrónica,* pp. 1-9, 2015.

[14] I. Hydara, A. Bakar Sultan, H. Zulzalil y N. Indriaty Admodisastro, «Removing Cross-Site Scripting Vulnerabilities from Web Applications using the OWASP ESAPI Security Guidelines,» *Indian Journal of Science and Technology,* pp. 1-5, 2015.

[15] R. Memen Akbar y M. A. Fadhly Ridha, «SQL Injection and Cross Site Scripting Prevention Using OWASP,» *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION,* pp. 286-292, 2018.

[16] L. González, «VOTO ELECTRÓNICO POR INTERNET Y RIESGOS PARA LA DEMOCRACIA,» *UNED,* pp. 213-249, 2015.

[17] A. Muñoz, S. Pérez y S. Donado, «Pentesting sobre aplicaciones web basado en la metodología OWASP utilizando un cluster conformado por dispositivos SBC de bajo costo.,» *Revista Ibérica de Sistemas e Tecnologias de Informação,* pp. 1-14, 01 Enero 2018.

[18] J. Veloz, A. Alcivar, G. Salvatierra y C. Silva, «Ethical hacking, una metodología para descubrir fallas de seguridad en sistemas informáticos mediante la herramienta KALI-LINUX,» *Revista de Tecnología de la Informática y las Comunicaciones*, 2017.

[19] A. Dar, B. Habib, F. Khurshid y T. Banday, «Experimental analysis of DDoS attack and it's detection in Eucalyptus private cloud platform,» de *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, 2016.

[20] I. Babincev y D. Vuletić, «WEB APPLICATION SECURITY ANALYSIS USING THE KALI LINUX OPERATING SYSTEM,» *MILITARY TECHNICAL COURIER*, p. 513–531, 2016.

[21] A. Parmar y K. Pattani, «Sniffing GSM Traffic Using RTL-SDR And Kali Linux OS,» *International Research Journal of Engineering and Technology (IRJET)*, pp. 1637-1642, 2017.

[22] M. Montes, D. Penazzi y N. Wolovick, «CONSIDERACIONES SOBRE EL VOTO ELECTRONICO,» *SEDIC*, pp. 297-307, 2016.

[23] P. Hidalgo Romero, «Seguridad en la web. Criptografía de llave pública,» *Pasaje a la ciencia*, pp. 49-56, 2017.

ANEXOS

Anexo 1: Requisitos funcionales y no funcionales

Tabla 10: Requisito funcional 1

Identificación del requerimiento:	**RF01**
Nombre del Requerimiento:	Autenticación
Actores Involucrados	Todos los Usuarios
Descripción del requerimiento:	El sistema debe permitir la autenticación de los usuarios del sistema
Prioridad del requerimiento: **Alta**	

Tabla 11: Requisito funcional 2

Identificación del requerimiento:	**RF02**
Nombre del Requerimiento:	Configurar Sistema
Actores Involucrados	Administrador
Descripción del requerimiento:	El sistema debe permitir ser configurado.
Prioridad del requerimiento: **Alta**	

Tabla 12: Requisito funcional 3

Identificación del requerimiento:	**RF03**
Nombre del Requerimiento:	Registrar
Actores Involucrados	Administrador
Descripción del requerimiento:	El sistema debe aceptar el registro de Votantes y candidatos.
Prioridad del requerimiento: **Alta**	

Tabla 13: Requisito funcional 4

Identificación del requerimiento:	**RF04**
Nombre del Requerimiento:	Realizar Voto
Actores Involucrados	Estudiante
Descripción del requerimiento:	El sistema debe permitir registrar el voto
Prioridad del requerimiento: **Alta**	

Tabla 14: Requisito funcional 5

Identificación del requerimiento:	**RF05**
Nombre del Requerimiento:	Resultados
Actores Involucrados	Administrador, Estudiante
Descripción del requerimiento:	El sistema debe mostrar los resultados de las votaciones.
Prioridad del requerimiento: **Alta**	

Tabla 15: Requisito no funcional 1

Identificación del requerimiento:	**RNF01**
Nombre del Requerimiento:	Sistema de votación electrónica
Características:	El entorno de ejecución debe ser a través de la computadora incluido internet.
Descripción del requerimiento:	El sistema debe estar en línea y ser de fácil uso, intuitiva y sencilla.
Prioridad del requerimiento: **Alta**	

Tabla 16: Requisito no funcional 2

Identificación del requerimiento:	**RNF02**
Nombre del Requerimiento:	Uso de un código de ingreso
Características:	Para la autentificación
Descripción del requerimiento:	El sistema debe implementar un código para que se compruebe la identidad del estudiante.
Prioridad del requerimiento: **Alta**	

Tabla 17: Requisito no funcional 3

Identificación del requerimiento:	**RNF03**
Nombre del Requerimiento:	Seguridad
Características:	Metodología OWASP
Descripción del requerimiento:	El sistema debe tener controles de seguridad para obtener a disponibilidad, la integridad y confidencialidad
Prioridad del requerimiento: **Alta**	

Tabla 18: Requisito no funcional 4

Identificación del requerimiento:	**RNF04**
Nombre del Requerimiento:	Certificado de votación
Características:	Correo con archivo adjunto
Descripción del requerimiento:	El sistema debe enviar un correo electrónico con el certificado de votación.
Prioridad del requerimiento: **Alta**	

Anexo 2: Resultados de los pentesting de cada indicador

Tabla 19: Resultado de tiempos de disponibilidad de los websites

Tiempo (en segundos)	Website 1		Website 2	
	Tiempo disponible_w1	Tiempo no disponible	Tiempo disponible_w2	Tiempo no disponible
t1	229	11	0	240
t2	229	11	0	240
t3	229	11	3	237
t4	230	10	1	239
t5	231	9	1	239
t6	229	11	0	240
t7	229	11	1	239
t8	229	11	1	239
t9	229	11	1	239
t10	229	11	1	239
t11	230	10	2	238
t12	229	11	1	239
t13	232	8	0	240
t14	229	11	1	239
t15	230	10	2	238
t16	229	11	3	237
t17	229	11	0	240
t18	229	11	0	240
t19	230	10	2	238
t20	229	11	2	238
t21	229	11	0	240
t22	229	11	1	239
t23	229	11	0	240
t24	229	11	1	239
t25	230	10	1	239
t26	229	11	2	238
t27	230	10	0	240
t28	230	10	3	237
t29	230	10	1	239
t30	229	11	0	240
Promedio	**229,43**	**10,57**	**1,03**	**238,97**
%	**95,60%**	**4,40%**	**0,43%**	**99,57%**

Tabla 20: Resultados de pentesting con BurpSuite hacia los módulos para medir el indicador de confidencialidad

Módulos	Website 1		Website 2	
	Confidencial	**No confidencial**	**Confidencial**	**No confidencial**
Configuración del sistema	x			x
Estudiantes	x		x	
Partidos Políticos	x			x
Cargos	x			x
Candidatos		x		x
Votaciones	x			x
Reportes	x			x
Total	**6**	**1**	**1**	**6**
%	85,71%	14,29%	14,29%	85,71%

Tabla 21: Resultados de pentesting con BurpSuite hacia los módulos para medir el indicador de integridad

Módulos	Website 1		Website 2	
	Integro	**No integro**	**Integro**	**No integro**
Configuración del sistema	x			x
Estudiantes	x		x	
Partidos Políticos	x			x
Cargos	x			x
Candidatos		x		x
Votaciones	x			x
Reportes	x		x	
Total	**6**	**1**	**2**	**5**
%	85,71%	14,29%	28,57%	71,43%

Tabla 22: Criterios que garanticen autenticidad

Criterios	Website 1		Website 2	
	Aplicados	**No aplicados**	**Aplicados**	**No aplicados**
Credenciales únicas (#cedula)	x		x	
Acceso con código único	x			
Escoger entre roles	x		x	
Captcha	x			
Protocolo HTTPS	x			
Registro del voto únicamente del usuario estudiante	x			
Total	**6**	**0**	**2**	**4**
%	**100%**	**0%**	**33,33%**	**66,67%**

Anexo 3: Resultado de Votaciones

Tabla 23: Resultados de votaciones para medir el indicador de confiabilidad

	Website 1		**Website 2**	
	Válidos	**Duplicados**	**Válidos**	**Duplicados**
Votos	65	0	65	13
Total	**65**	**0**	**65**	**13**

Anexo 4: Prueba de normalidad

```
> normalityTest(~tiempo_disponible_w2, test="shapiro.test", data=Dataset)

        Shapiro-Wilk normality test

data:  tiempo_disponible_w2
W = 0.84235, p-value = 0.0003497


> normalityTest(~tiempo_disponible_w1, test="shapiro.test", data=Dataset)

        Shapiro-Wilk normality test

data:  tiempo_disponible_w1
W = 0.64799, p-value = 0.0000002161
```

Figura 40: Prueba de normalidad, tiempos de Disponibilidad
Fuente: Software Estadístico R

Anexo 5: Resultados de los test de cada indicador

```
> with(Dataset, median(tiempo_disponible_w1 - tiempo_disponible_w2, na.rm=TRUE)) # median difference
[1] 228

> with(Dataset, wilcox.test(tiempo_disponible_w1, tiempo_disponible_w2, alternative='two.sided', paired=TRUE))

        Wilcoxon signed rank test with continuity correction

data:  tiempo_disponible_w1 and tiempo_disponible_w2
V = 496, p-value = 0.0000009443
alternative hypothesis: true location shift is not equal to 0
```

Figura 41: Test de Wilcoxon del criterio disponibilidad
Fuente: Software Estadístico R

```
> .Table  # Counts
                Websites
Módulos          Website 1 Website 2
  Confidencial           6         1
  No Confidencial        1         6

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

        Pearson's Chi-squared test

data:  .Table
X-squared = 7.1429, df = 1, p-value = 0.007526
```

Figura 42: Test de Chi Cuadrado del criterio confidencialidad
Fuente: Software Estadístico R

```
> .Table  # Counts
            Websites
Módulos      Website 1 Website 2
  Integro            6         2
  No Integro         1         5

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

        Pearson's Chi-squared test

data:  .Table
X-squared = 4.6667, df = 1, p-value = 0.03075
```

Figura 43: Test de Chi Cuadrado del criterio integridad
Fuente: Software Estadístico R

```
> .Table  # Counts
              Websites
Criterios      Website 1 Website 2
  Aplicados            6         2
  No Aplicados         0         4

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

        Pearson's Chi-squared test

data:  .Table
X-squared = 6, df = 1, p-value = 0.01431
```

Figura 44: Test de Chi Cuadrado del criterio autenticidad
Fuente: Software Estadístico R

```
> .Table  # Counts
              Websites
Votos          Website 1 Website 2
  Válidos             65        65
  Duplicados           0        13

> .Test <- chisq.test(.Table, correct=FALSE)

> .Test

         Pearson's Chi-squared test

data:  .Table
X-squared = 11.917, df = 1, p-value = 0.0005563
```

Figura 45: Test de Chi Cuadrado del criterio confiabilidad
Fuente: Software Estadístico R

Anexo 6: Manual de Usuario

MANUAL DE USUARIO

WEBSITE DE VOTACIONES

RIOBAMBA-ECUADOR
2020

UNIVERSIDAD NACIONAL DE CHIMBORAZO
Creado por:
Miriam Elizabeth Morocho Buñay
Fabricio Andrés Tasan Garcés

INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN

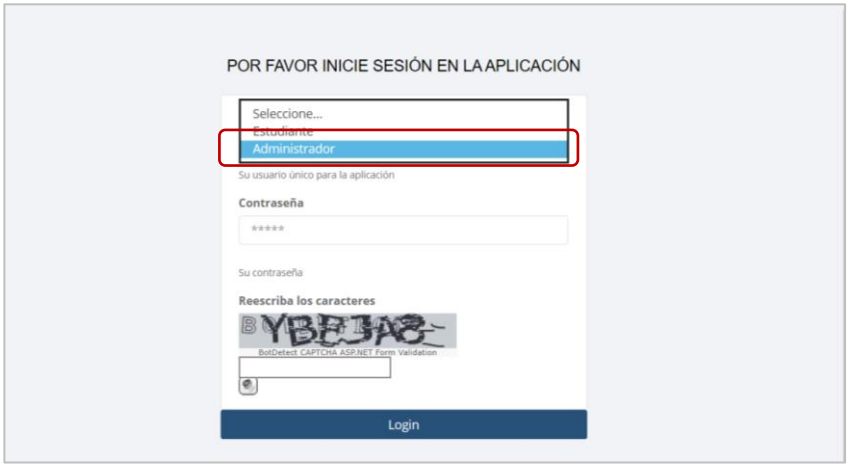
USUARIO ADMINISTRADOR

A continuación se describe los pasos que debe realizar el administrador para registrar informacion del proceso de votaciones.

Paso 1	• Ingresar al sitio web con la url : https://votacionesti2.tk • En la parte superior derecha se podrá visualizar dos opciones de ingreso y manual de usuario.

Paso 2	Dar clic en *Ingreso*

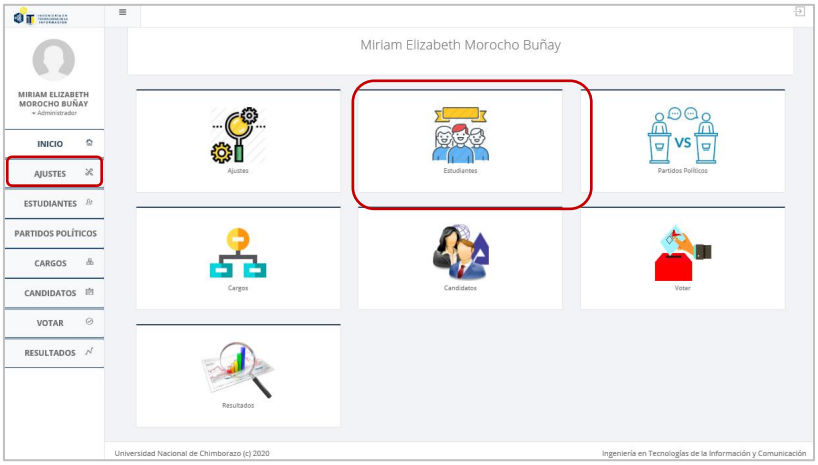
Paso 3	• Luego se mostrará la página de Login. • Desplegar, y seleccionar ○ Usuario: Administrador



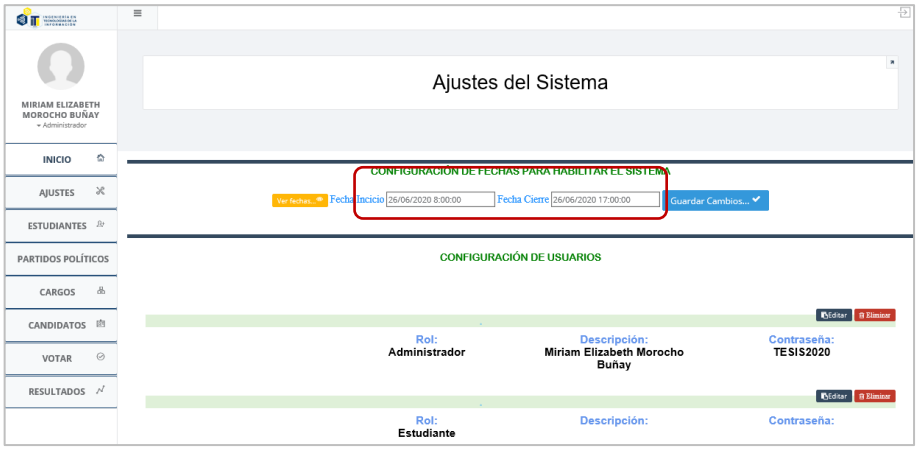
Paso 4	• Colocar su contraseña • Ingresar el captcha respectivo • Dar clic en *Logi*n

Módulo de Ajustes

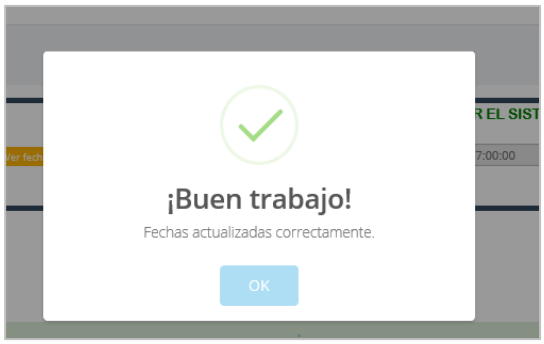
Paso 5	• Dar clic en el menú Ajustes, para ello, tiene dos opciones como se visualiza en la imagen.



Paso 6	• Aparecerá las fechas de apertura del sistema. • Llenar las fechas establecidas para la votación. • Dar clic en guardar cambios.

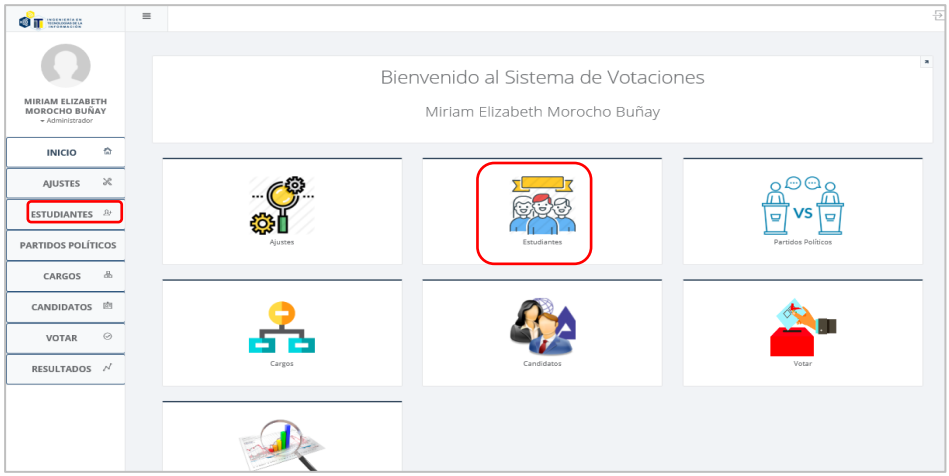


Paso 7	• Aparecerá un mensaje, de la actualización de fechas.

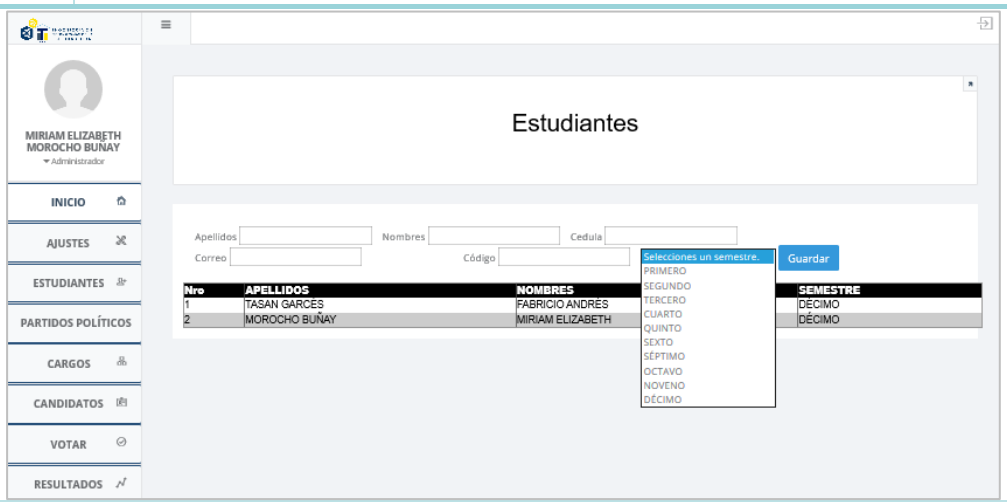


Módulo de Estudiantes

Paso 8	• Dar clic en el menú Estudiantes.

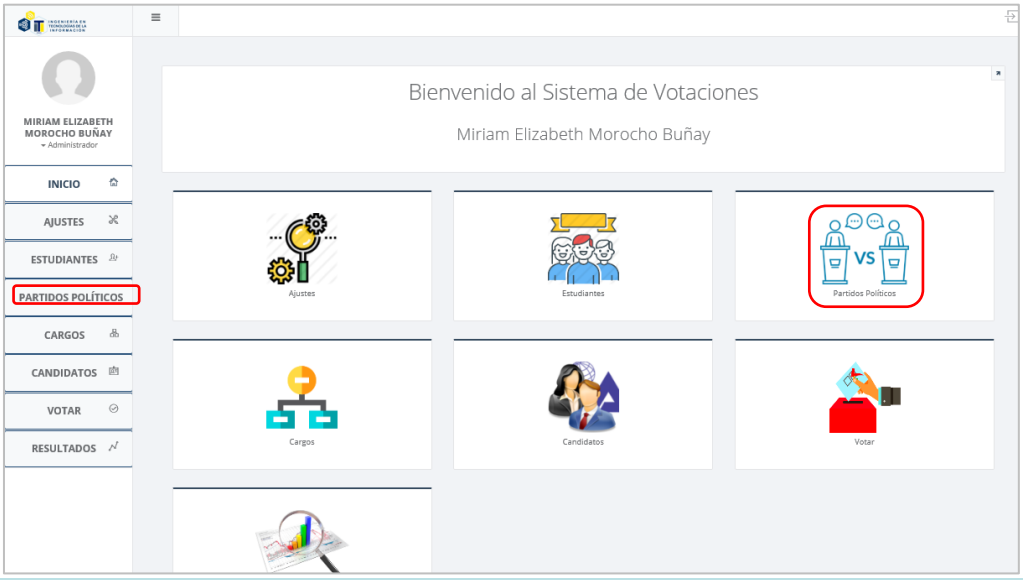


Paso 9	• Llenar los datos correspondientes a cada estudiante, los mismos que, se irán listando.



Módulo de Partidos Políticos

Paso 10	• Dar clic en el módulo de Partidos Políticos.



Paso 11	• Dar clic en Nuevo.

Paso 12	• Registrar los datos requeridos • Dar clic en Guardar • Si no está seguro de ingresar los datos, dar clic en Cerrar.

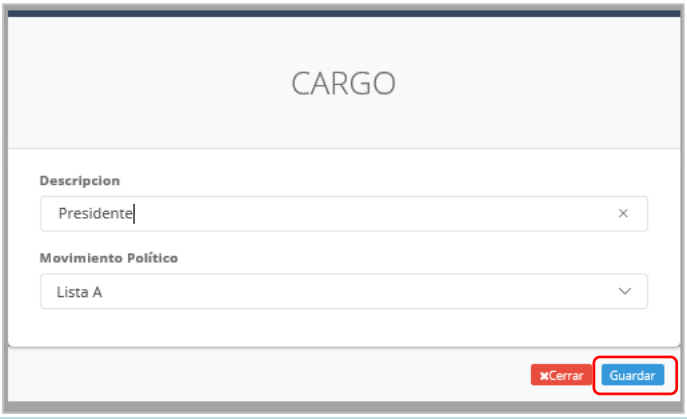
Paso 13	• Se visualizará la lista ingresada, en el cual tiene las opciones de actualizar, o eliminar. • Del mismo modo se podrá ir añadiendo más partidos políticos (Listas).

Módulo de Cargos

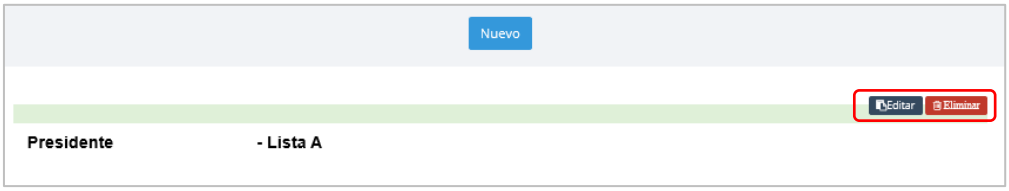
Paso 14	• Dar clic en el menú de Cargos.

Paso 15	• Dar clic en Nuevo.

Paso 16	• Registrar los cargos que vaya a tener la Lista anteriormente ingresada. • Dar clic en Guardar

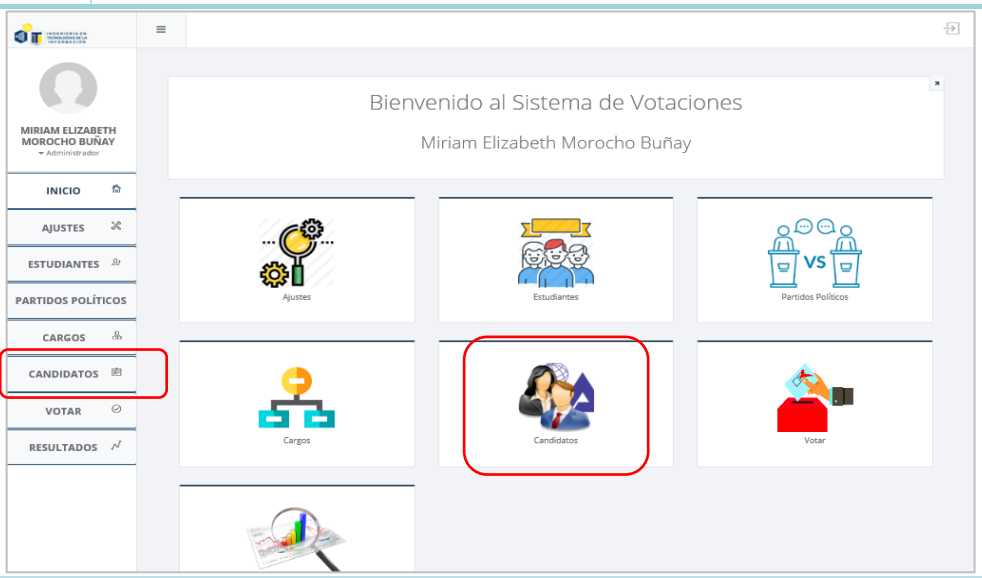


Paso 17	• Se listará el cargo correspondiente a la lista, en la misma que, podrá editar o eliminar. • De la misma manera se podrá seguir añadiendo más cargos.



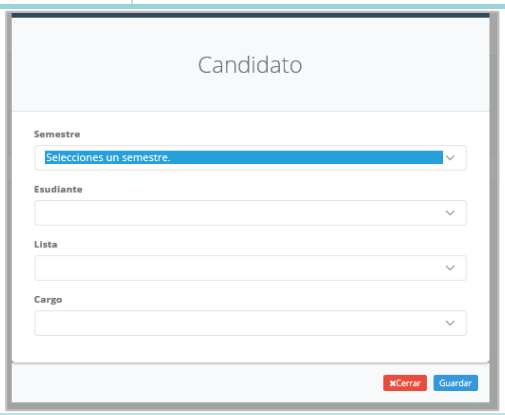
Módulo de Candidatos

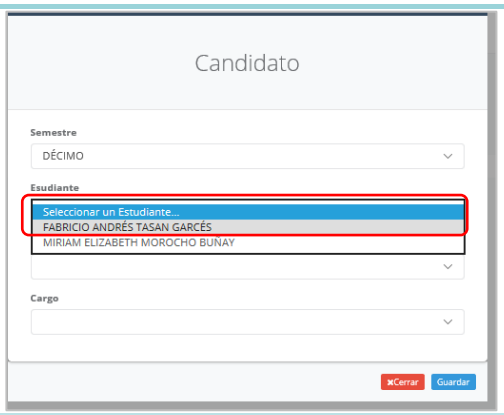
Paso 18	• Dar clic en el menú de Candidatos.



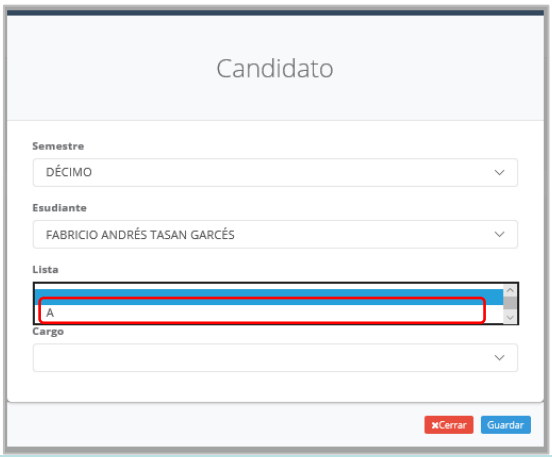
Paso 19	• Dar clic en nuevo

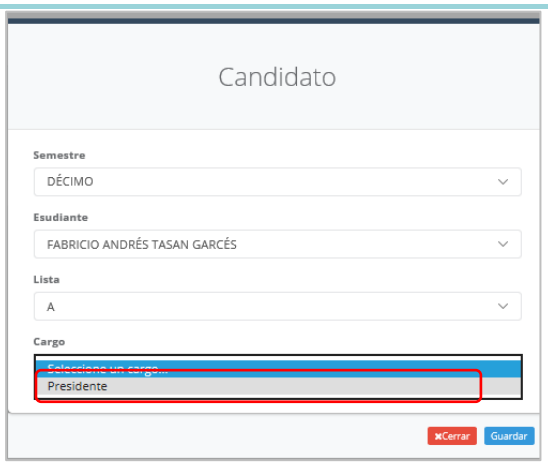
Paso 20	• Seleccionar el semestre al cual pertenece el candidato. • Seleccionar el estudiante candidato.



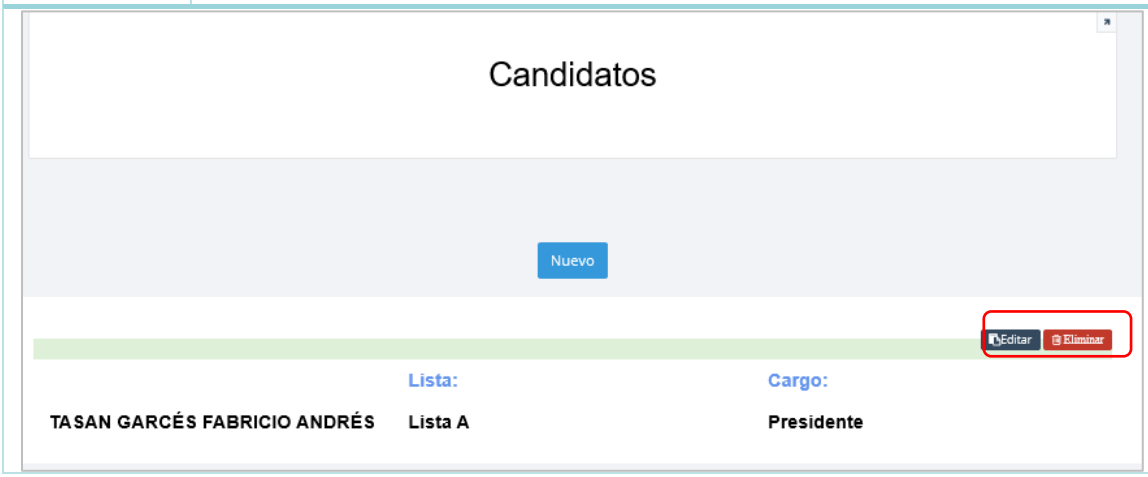


Paso 21	• Seleccionar la Lista y Cargo correspondiente. • Dar clic en Guardar.



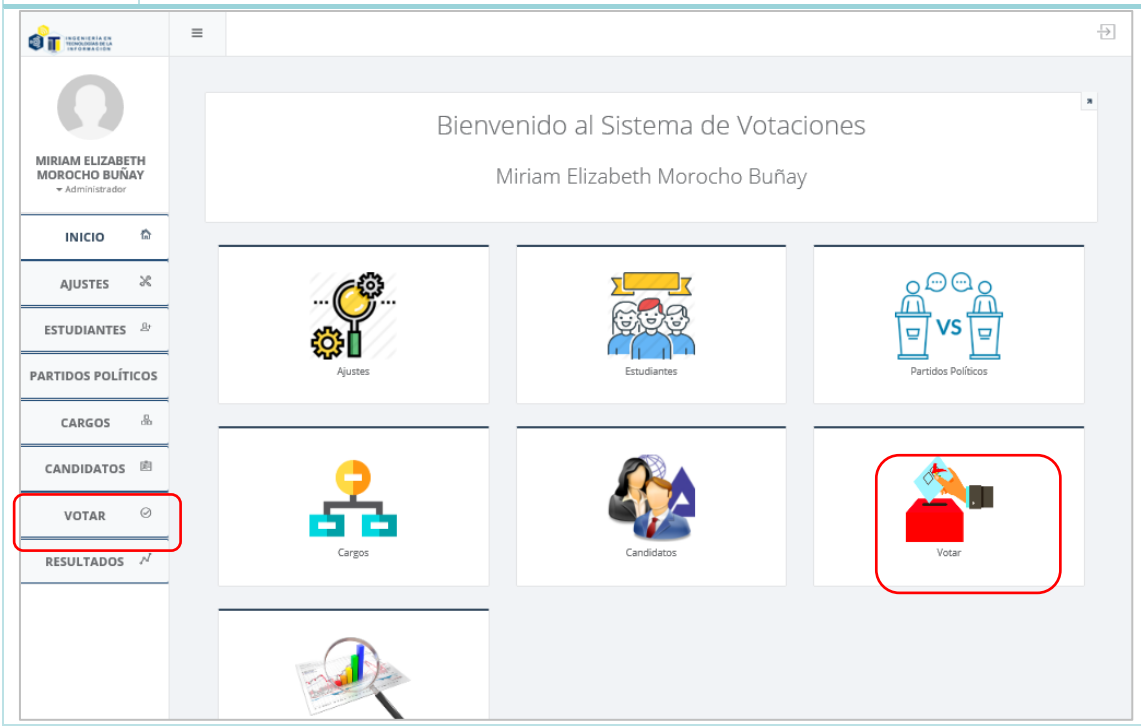


Paso 22	• Se listará el candidato registrado, el mismo que tiene la opción de editarlo o eliminarlo.



Módulo de votaciones

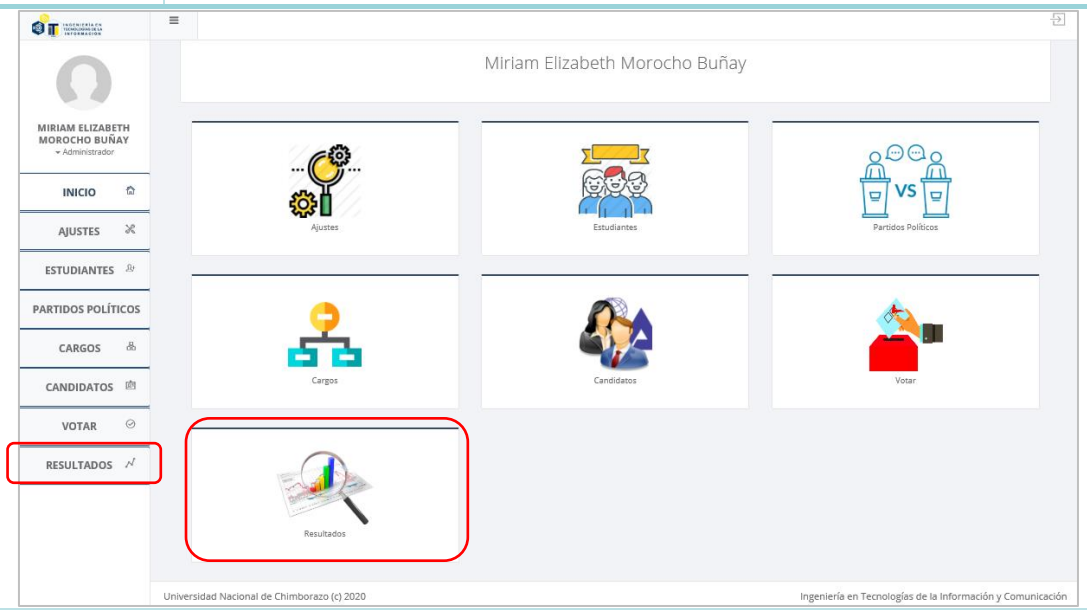
Paso 23	• Dar clic en el menú de Votaciones • El administrador solo puede visualizar más no registrar el voto.



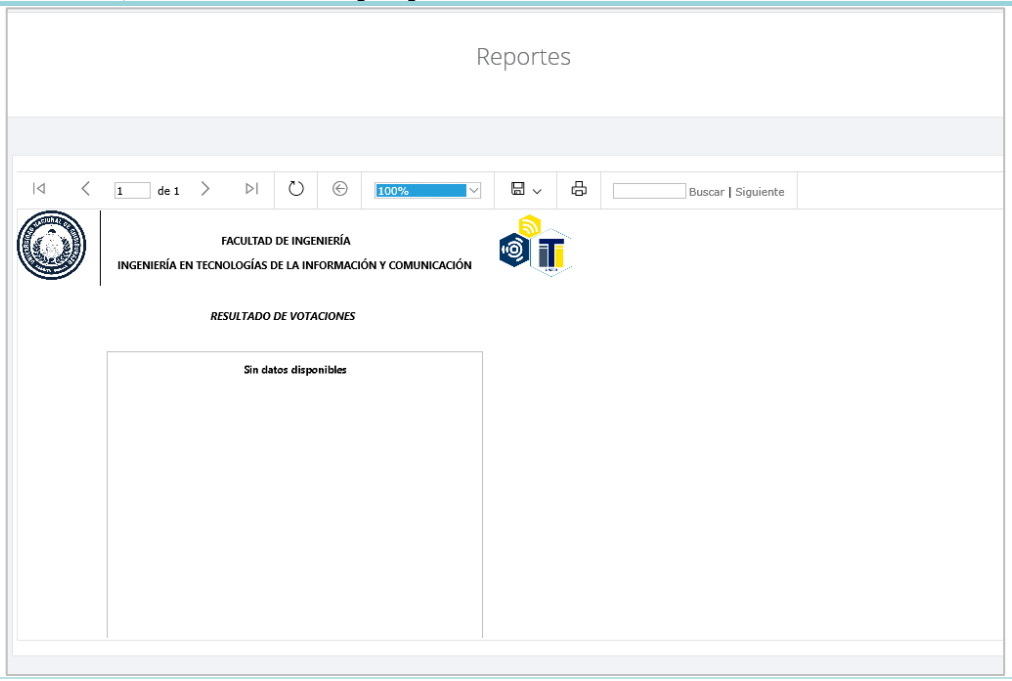
Paso 24	• Se visualizará la interfaz del módulo de votar, en los cuales, el voto nulo y voto blanco, ya vienen insertados por defecto.

Módulo de Resultados

Paso 25	• Dar clic en el menú de Resultados.



Paso 26	• Del mismo modo, tiene acceso para visualizar el módulo de reportes. • Como se visualiza no existen datos, pues, aunó se han registrado los votos por parte del estudiante.



Paso 27	• Para salir, tiene dos opciones.
	  ○ Desplegar en la parte superior izquierda, y dar clic en Cerrar sesión.
	  ○ O, dar clic en la parte superior derecha en el icono señalado.

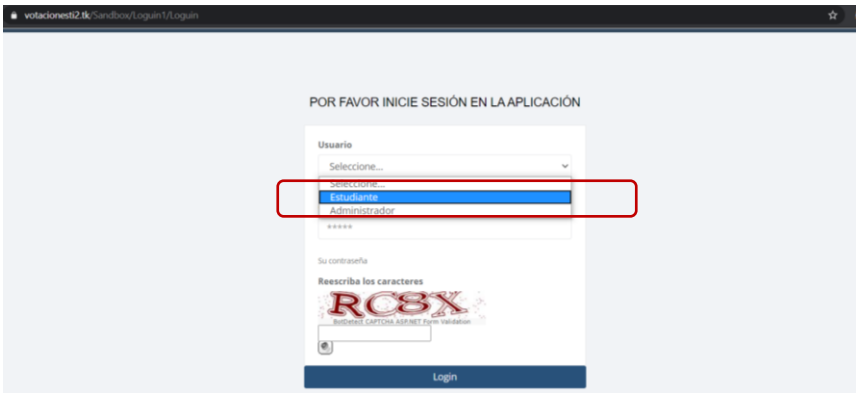
USUARIO ESTUDIANTE

A continuación se describe los pasos para que el estudiante de la carrera de Tecnologías de la Información pueda acceder a votar en el website con mayor nivel de seguridad.

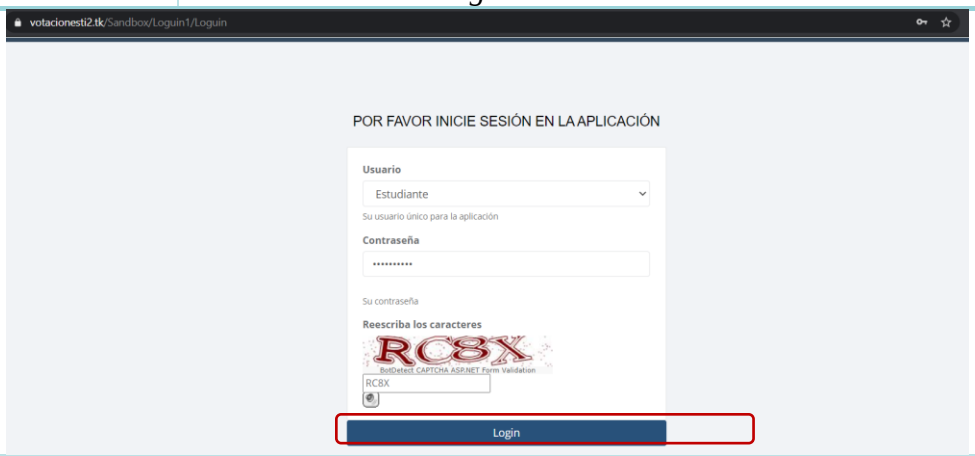
Paso 3	• Ingresar al sitio web con la url : https://votacionesti2.tk • En la parte superior derecha se podrá visualizar dos opciones de ingreso y manual de usuario.

Paso 4	Dar clic en *Ingreso*

Paso 3	• Luego se mostrará la página de Login. • Desplegar, y seleccionar ○ Usuario: Estudiante



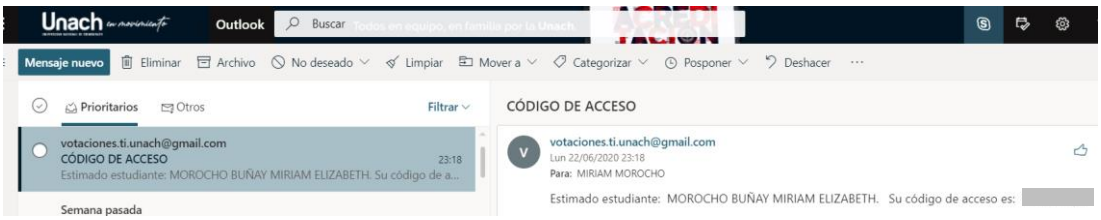
Paso 4	• Colocar su contraseña ○ Contraseña: su número de cedula (sin guion) • Ingresar el captcha respectivo • Dar clic en *Logi*n



Paso 5	• Para ingresar al sistema, debe obtener el código, para ello debe dar clic en *Obtener código,* y esperar un momento hasta que el código se envié a su correo proporcionado.

Paso 6	• Se notificará la dirección de correo, al que se ha enviado su código de acceso. • NOTA: De tener problemas al obtener su código, comunicarse a votaciones.ti.unach@gmail.com

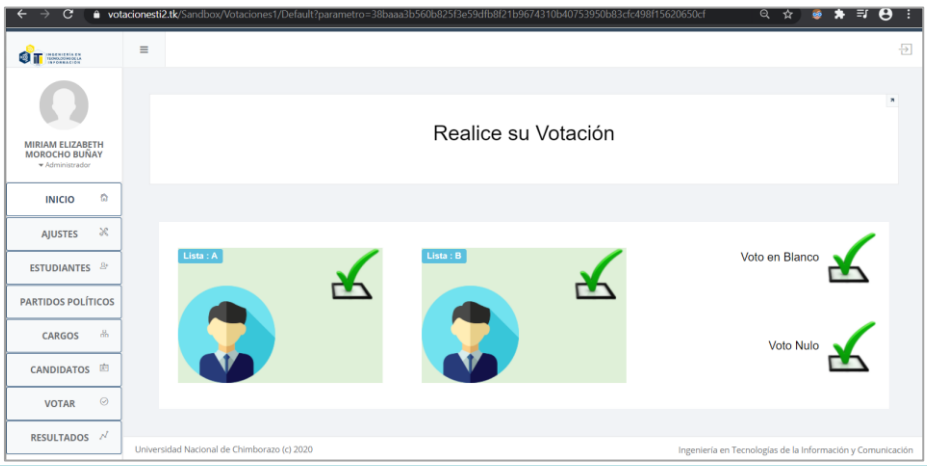
Paso 7	• Verificar que el código ha sido enviado al correo.



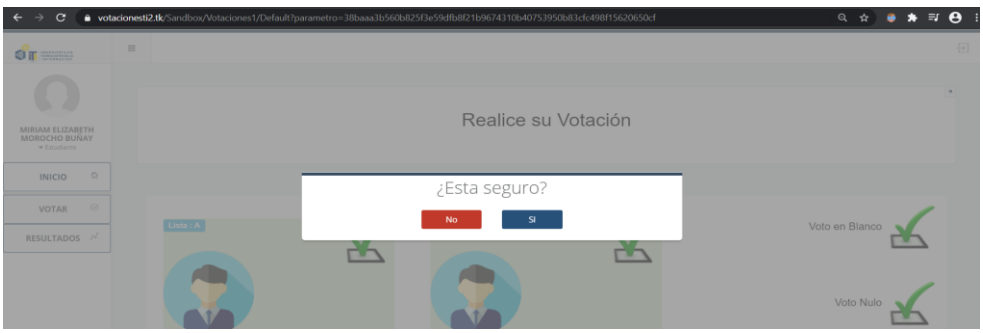
Paso 8	• Colocar su código de acceso, y dar clic en ingresar.

Paso 9	• Al ingresar se mostrará la hora de inicio y cierre del sistema, así como también a los dos módulos disponibles. • Dar clic en *Votar* para registrar su voto, tiene dos opciones, en el menú ingreso, o en la pantalla principal.

Paso 10	• Se mostrará las opciones, usted deberá elegir si votar por una lista, votar en blanco, o anular el voto.

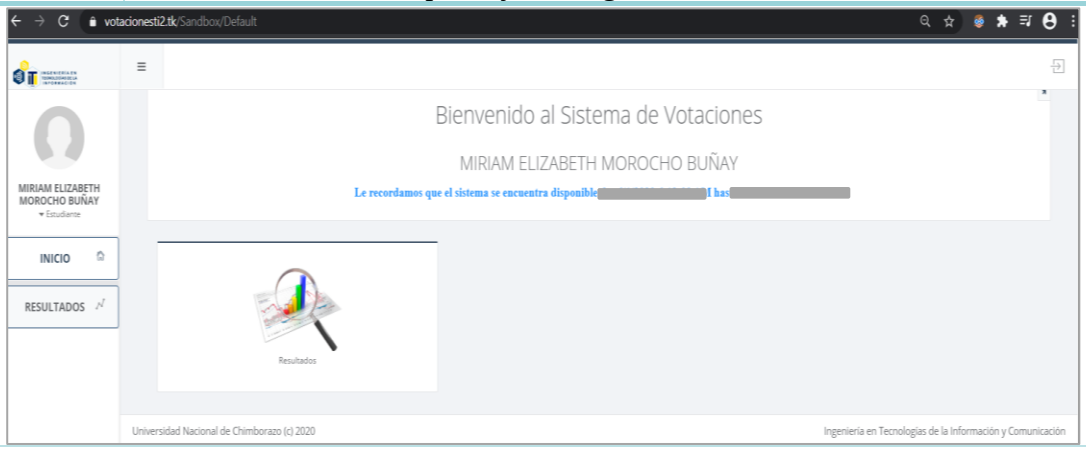


Paso 11	• Escoger la opción *SI*, para votar por la lista u opción seleccionada. • NOTA: Si escoge la opción NO, podrá seguir navegando hasta que este seguro, de votar.



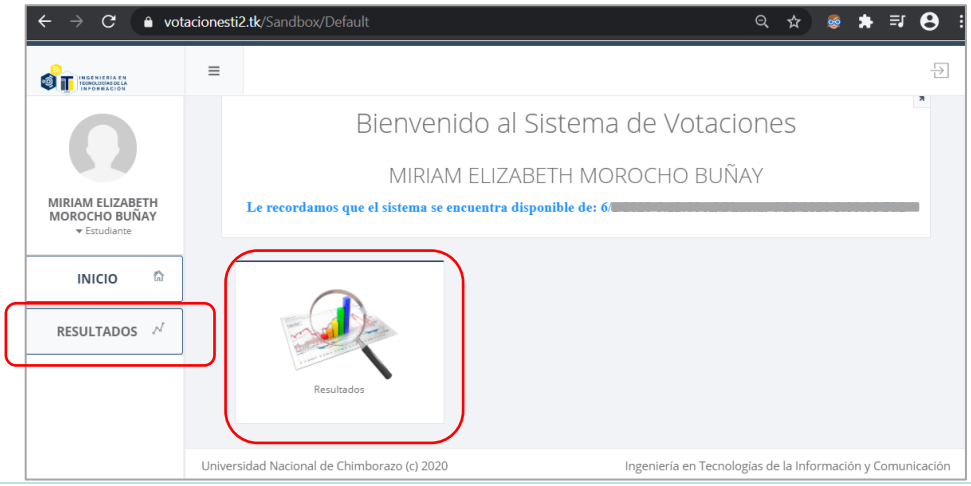
Paso 12	• Esperar hasta que el voto se registre.

Paso 13	• Luego se redireccionará a la página principal, solo con la opción de los resultados, pues, ya ha registrado el voto.

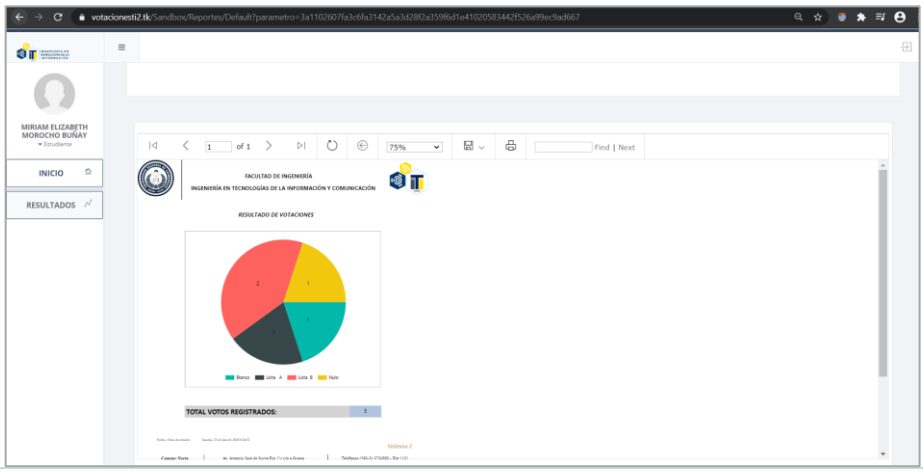


Paso 14	• Para constatar que el voto se registró exitosamente, debe comprobar en su correo electrónico (al mismo al que se le envió el código de acceso), pues se le notificará con su respectivo certificado.

Paso 15	• Para visualizar los votos que se van generando hacer clic, en *Resultados.*



Paso 16	• Si se requiere se puede descargar el reporte generado.



Paso 17	• Para salir, tiene dos opciones. ○ Desplegar en la parte superior izquierda, y dar clic en Cerrar sesión o dar clic en la parte superior derecha en el icono señalado.