



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TÍTULO DEL TRABAJO DE TITULACIÓN

“Diseño de una metodología de seguridad informática basada en OWASP
para evaluar vulnerabilidades del sistema ViTECED”

AUTOR

Angel Tomala, Cinthia Vanessa

PROYECTO DE UNIDAD DE INTEGRACIÓN CURRICULAR

Previo a la obtención del grado académico en
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN

TUTOR

Ing. Comp. Marcia Bayas Sampedro, PhD.

Santa Elena, Ecuador

Año 2026



UPSE

**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

TRIBUNAL DE SUSTENTACIÓN

Ing. José Sánchez Aquino, Mgtr.
DIRECTOR DE LA CARRERA

Ing. Comp. Marcia Bayas S. PhD.
TUTOR

Ing. Lidice Haz López, Mgtr.
DOCENTE ESPECIALISTA

Ing. Marjorie Coronel Suárez, Mgtr.
DOCENTE GUÍA UIC



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por **Angel Tomala Cinthia Vanessa**, como requerimiento para la obtención del título de Ingeniero en Tecnologías de la Información.

Santa Elena, a los 15 días del mes de agosto de 2026

TUTOR



Ing. Comp. Marcia Marisol Bayas Sampedro, Ph.D.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
DECLARACIÓN DE RESPONSABILIDAD**

Yo, **Angel Tomala Cinthia Vanessa**

DECLARO QUE:

El Trabajo de Titulación, titulado “Diseño de una metodología de seguridad informática basada en OWASP para evaluar vulnerabilidades del sistema ViTECED”, fue desarrollado como requisito para la obtención del título de Ingeniero en Tecnologías de la Información, respetando los derechos intelectuales de terceros debidamente citados. Este trabajo forma parte del proyecto de investigación “Investigación de las aplicaciones de IoT en la adquisición de señales bioeléctricas”, financiado por la Universidad Estatal Península de Santa Elena, código CUP 91870000.0000.389571, bajo la dirección del Ph.D. Ronald Rovira. La idea y el diseño conceptual fueron propuestos por mi tutora, la Ph.D. Marcia Bayas, y mi contribución se centró en el desarrollo de la metodología basada en OWASP para la evaluación de vulnerabilidades de ViTECED. En virtud de lo expuesto, asumo la responsabilidad por el contenido, veracidad y alcance del presente Trabajo de Titulación.

Santa Elena, a los 15 días del mes de agosto de 2026

EL AUTOR

Cinthia Vanessa Angel Tomala



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado Diseño de una metodología de seguridad informática basada en OWASP para evaluar vulnerabilidades del sistema ViTECED, presentado por el estudiante, Angel Tomala Cinthia Vanessa fue enviado al Sistema Antiplagio, presentando un porcentaje de similitud correspondiente al 10%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



Certificado de análisis
Compilatio Magister+ | UPSE-ECU

Proyecto Angel Tomala a

ID : a4acc8f4ef6e80c77022e9ee367af1e45ead8bbf



10%
Textos sospechosos

Nombre del fichero : Proyecto Angel Tomala a.txt
Tamaño del archivo original : 6,44 MB
Número de palabras : 33.814

Depositante : MARCIA MARISOL BAYAS SAMPEDRO
Fecha de depósito : 15 de agosto de 2026
Tipo de carga : interface
fecha de fin de análisis : 15 de agosto de 2026

TUTOR



**Marcia Marisol
Bayas Sampedro**



Ing. Comp. Marcia Marisol Bayas Sampedro, Ph.D



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES**

AUTORIZACIÓN

Yo, **Angel Tomala Cinthia Vanessa**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del trabajo de titulación con fines de difusión pública, dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor.

Santa Elena, a los 15 días del mes de agosto de 2026

EL AUTOR

Cinthia Vanessa Angel Tomala

AGRADECIMIENTO

Agradezco, en primer lugar, a Dios, por brindarme la sabiduría, la fortaleza y la oportunidad de alcanzar este importante logro en mi vida.

A mi familia; padres y hermanos por su amor, consejos y respaldo incondicional que me impulsaron a seguir adelante y a nunca dejar de creer en mis sueños. A Javier Panchana, por su paciencia, confianza, amor y apoyo incondicional durante este proceso, siendo una fuente constante de motivación para alcanzar esta meta.

Asimismo, expreso mi más sincero agradecimiento a mi tutora, Ing. Comp. Marcia Bayas S. Ph.D., por su guía, paciencia, experiencia y apoyo incondicional durante el desarrollo de esta investigación. También, extendiendo mi gratitud al grupo de investigación TECED, especialmente a la Ing. Angela Parra, por el apoyo en el acceso a la plataforma informática durante el desarrollo de este proyecto.

De igual manera, expreso mi gratitud a la Ing. Marjorie Coronel S. y a todos los docentes de la carrera de Ingeniería en Tecnología de la Información, quienes contribuyeron con su experiencia y enseñanzas a mi formación académica y profesional.

Cinthia Vanessa, Angel Tomala

DEDICATORIA

Dedico este trabajo, en primer lugar, a Dios, por iluminarme mi camino, darme la fortaleza para superar cada desafío y permitirme cumplir una de las metas más importante de mi vida.

A mi familia, por ser mi mayor fuente de amor, inspiración y apoyo incondicional. Gracias por sus enseñanzas, consejos y por creer siempre en mí, incluso en los momentos más difíciles.

A Javier Panchana, por acompañarme con paciencia, amor y confianza durante este camino. Gracias por ser un apoyo constante, por celebrar conmigo cada pequeño logro que hizo posible alcanzar esta meta.

A mis amistades; Jordy Yagual, Ketty Salvador, Alex Alejandro y demás amistades que compartieron conmigo en esta etapa de aprendizaje, por sus palabras de aliento, su apoyo y los momentos que hicieron este recorrido más llevadero.

Finalmente, dedico este logro a mí misma, como reflejo del esfuerzo, la perseverancia y el compromiso que me permitieron convertir este sueño en una realidad.

Cinthia Vanessa, Angel Tomala

ÍNDICE GENERAL

TRIBUNAL DE SUSTENTACIÓN	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
DECLARO QUE:	IV
CERTIFICACIÓN DE ANTIPLAGIO	V
AUTORIZACIÓN	VI
AGRADECIMIENTO	VII
DEDICATORIA	VIII
ÍNDICE GENERAL	IX
INDICE DE TABLAS	XI
RESUMEN	XVIII
ABSTRACT	XVIII
INTRODUCCIÓN	20
CAPÍTULO I	23
1. FUNDAMENTACIÓN	23
1.1. Antecedentes	23
1.2. Descripción del Proyecto	26
1.3. Objetivos del Proyecto	29
1.4. Justificación del Proyecto	29
1.5. Alcance del Proyecto	31
1.6. Metodología del Proyecto	34
1.6.1. Metodología de la investigación	34
1.6.2. Beneficiarios del Proyecto	37
1.6.3. Variables	38

1.7.	Metodología de Desarrollo	42
CAPÍTULO II		45
2.	PROPUESTA	45
2.1.	Marco Contextual	45
2.2.	Marco Conceptual	45
2.2.1.	Seguridad Informática	45
2.2.2.	Pentesting o pruebas de penetración	46
2.2.3.	Herramientas para análisis de seguridad	47
2.2.4.	Herramientas para prueba de seguridad	49
2.2.5.	Prueba de Análisis Estadístico	52
2.3.	Marco Teórico	52
2.3.1.	Ataques informáticos dirigidos a los sistemas de salud.	52
2.3.2.	Principales vulnerabilidades identificadas en los sistemas de salud.	54
2.3.3.	Vulnerabilidades más comunes detectadas en aplicaciones web y móvil.	54
2.3.4.	Aplicación de OWASP en los sistemas informáticos	56
2.3.6.	Requerimientos no funcionales	58
2.4.	Componentes de la propuesta	59
2.5.	Arquitectura del sistema	59
2.5.1.	Componentes del sistema ViTECED	61
2.5.2.	Puntos críticos considerados para el análisis del sistema ViTECED	61
2.6.	Metodología basada en OWASP	72
RESULTADOS		79
CONCLUSIONES		167
RECOMENDACIONES		168
ANEXOS		189

INDICE DE TABLAS

Tabla 1 Componentes del Sistema ViTECED	36
Tabla 2 Módulos del Sistema.	36
Tabla 3 APIs principales del Sistema.	37
Tabla 4 Servicios del Sistema.	37
Tabla 5 Bases de datos del Sistema	37
Tabla 6 Usuarios considerados para la prueba	37
Tabla 7 Análisis de la variable	38
Tabla 8 Lista de vulnerabilidades más comunes en aplicaciones web y móvil según OWASP	55
Tabla 9 Requerimientos funcionales – Actividad de recopilación de información	56
Tabla 10 Requerimientos funcionales - Actividad de identificación de vulnerabilidades	57
Tabla 11 Requerimientos funcionales - Actividad de remediación	57
Tabla 12 Requerimiento funcionales - Actividad de reevaluación	58
Tabla 13 Requerimientos funcionales - Actividad de elaboración del informe	58
Tabla 14 Requerimientos no funcionales - Seguridad y cuidado de la información	59
Tabla 15 Requerimientos no funcionales - Claridad y presentación	59
Tabla 16 Componente del sistema ViTECED	61
Tabla 17 Punto críticos considerados para el análisis del sistema ViTECED	63
Tabla 18 Resultados general de la evaluación del modelo SAMM en el sistema ViTECED	63
Tabla 19 Resultados de evaluación de prácticas de seguridad y niveles de madurez en el sistema ViTECED mediante SAMM	64

Tabla 20 Prueba conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información	73
Tabla 21 Prueba huellas digitales en el servidor web	73
Tabla 22 Prueba revisión de los metas archivos del servidor web en busca de fugas de información	74
Tabla 23 Prueba enumere las aplicaciones en el servidor web	74
Tabla 24 Prueba identificar puntos de entrada de la aplicación web	75
Tabla 25 Prueba de definición de Roles	75
Tabla 26 Prueba de enumeración y adivinanzas de cuentas de usuario	76
Tabla 27 Prueba de transporte de credenciales en canal encriptado	76
Tabla 28 Pruebas para buscar la debilidad de memoria caché del navegador	77
Tabla 29 Prueba de credenciales por defecto	77
Tabla 30 Pruebas para determinar el escalamiento de privilegios	77
Tabla 31 Pruebas de los atributos de las cookies	78
Tabla 32 Prueba para falsificación de petición de sitio cruzado (CSRF)	78
Tabla 33 Pruebas de errores de código	79
Tabla 34 Pruebas para el envío de información sensible por canales sin encriptar	79
Tabla 35 Resultados de la prueba conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información	81
Tabla 36 Tablas de resultados de puertos abiertos	86
Tabla 37 Registro de endpoints identificados mediante solicitudes GET	91
Tabla 38 Registro de endpoints identificados mediante solicitudes POST	92
Tabla 39 Parámetros identificados en la petición GET	93
Tabla 40 Parámetros identificados en las peticiones POST	98
Tabla 41 Encabezados adicionales identificados - OWASP ZAP	100

Tabla 42 Resultados de la evaluación de métodos HTTP potencialmente peligrosos en la API de la plataforma	108
Tabla 43 Resultados de la evaluación de métodos HTTP arbitrarios en la API y la aplicación web	109
Tabla 44 Resultado de Control de Acceso mediante el Método HEAD	110
Tabla 45 Prueba HTTP Strict Transport Security	111
Tabla 46 Roles identificados del sistema	113
Tabla 47 Matriz de roles y permisos	117
Tabla 48 Verificación de funcionalidades del perfil Super Administrador	118
Tabla 49 Verificación de funcionalidades del perfil Administrador	119
Tabla 50 Verificación de funcionalidades del perfil Medico	120
Tabla 51 Verificación de funcionalidades del perfil Asistente	120
Tabla 52 Verificación de funcionalidades del perfil Paciente	120
Tabla 53 Comparación de respuestas del mecanismo de autenticación ante diferentes escenarios de credenciales	122
Tabla 55 Direcciones identificadas en el sistema	153
Tabla 56 Interpretación de resultados	156
Tabla 57 Nivel de gravedad del riesgo	163
Tabla 58 Determinación de la gravedad de Riesgo	163
Tabla 59 Frecuencia de los resultados de las pruebas	164
Tabla 60 Estado Cantidad Porcentaje sobre hallazgos	165
Tabla 61 Indicadores obtenidos	166

ÍNDICE DE FIGURAS

Figura 1 Metodología de pruebas de penetración con OWASP.....	43
Figura 2 Triada CID: Confidencialidad, Integridad y Disponibilidad.....	46
Figura 3 Componentes funcionales y prácticas de seguridad del modelo SAMM.....	48
Figura 4 Riesgo de seguridad en Sistemas.....	53
Figura 5 Arquitectura del sistema ViTECED	60
Figura 6 Análisis gráfico de las funciones comerciales y prácticas de seguridad evaluadas en el sistema ViTECED mediante el modelo SAMM.	64
Figura 7 identificaron mensajes de error visibles en motores de búsqueda.....	82
Figura 8 Solicitud con métodos inexistentes	83
Figura 9 Verificación del archivo robost.txt mediante wget	85
Figura 10 Encabezados de respuesta HTTP identificados durante la evaluación - OWASP ZAP.....	100
Figura 12 Inspección cookies en el código HTML.....	102
Figura 13 Inspección del código fuente HTML del sistema ViTECED.....	103
Figura 14 Resultado del análisis de Wappalyzer sobre el dominio principal del sistema	104
Figura 15 Resultado de la enumeración de directorios con Gobuster sobre el dominio del sistema.....	105
Figura 16 Análisis de métodos HTTP OPTIONS en la API Mongo - Curl.....	106
Figura 17 Análisis de métodos HTTP OPTIONS en la API REST – Curl.....	106
Figura 18 Análisis del método HEAD en la API REST – Curl.....	110
Figura 19 Interfaz del perfil Administrador.....	114
Figura 20 Interfaz del perfil Super Administrador.	114
Figura 21 Interfaz del perfil Paciente.	114

Figura 22 Interfaz del perfil Medico.....	115
Figura 23 Interfaz del perfil Asistente.	115
Figura 24 Intento de inicio de sesión con Usuario válido y contraseña incorrecta.	121
Figura 25 Intento de inicio de sesión con credenciales invalidas.	122
Figura 26 Análisis del mensaje de error tras inicio de sesión con credenciales invalidas.	123
Figura 27 Título de la página durante la autenticación con usuario válido y contraseña incorrecta.	124
Figura 28 Verificación de cuenta mediante la respuesta del servidor ante un correo inexistente - Burp Suite.....	125
Figura 29 Verificación de cuenta mediante correo electrónico existente.....	126
Figura 30 Verificación de cuenta mediante la respuesta del servidor ante un correo existente- Burp Suite.....	126
Figura 31 Código backend para la emisión de tokens de sesión y manejo de mensajes de respuesta según el estado del usuario.....	127
Figura 32 Análisis integral de la gestión de respuestas y mensajes de error en el proceso de autenticación de la plataforma.	128
Figura 33 <i>Análisis de un mensaje recibido de una instalación de recuperación en el código fuente.</i>	129
Figura 34 Análisis del título de la Página Web en el código fuente.....	129
Figura 35 Envío de datos con el método POST a través de HTTP.....	130
Figura 36 Envío de datos con el método POST a través de HTTPS.	131
Figura 37 Envío de datos con el método POST a través de HTTPS en una página accesible a través de HTTP.....	131
Figura 38 Envío de datos con el método GET a través de HTTPS.	132

Figura 39 Resultado al presionar "Atrás" tras cerrar sesión - pantalla de "Acceso Denegado".	134
Figura 40 Verificación de nombres de usuario y contraseñas administrativas comunes.	135
Figura 41 Verificación de formato de credenciales.	136
Figura 42 Verificación del mecanismo de generación del nombre de usuario.	136
Figura 43 Validación de parámetros en el endpoint de autenticación ante el envío de un campo de contraseña vacío.	138
Figura 44 Solicitud de crear un usuario administrador con token de paciente.	140
Figura 45 Verificación del atributo de cookie - OWASP ZAP.	142
Figura 46 Solicitud legítima POST /editar_contra capturada mediante Burp Suite.	143
Figura 47 Solicitud generada desde el origen externo.	144
Figura 48 Proceso de autenticación exitoso y obtención del token JWT en ViTECED.	145
Figura 49 Solicitud autenticada mediante el encabezado Autorización.	146
Figura 50 Rechazo de una solicitud con un token JWT modificado.	146
Figura 51 Rechazo de una solicitud sin token de autenticación.	147
Figura 52 Análisis de manejo de error HTTP 404 Not Found.	148
Figura 53 Análisis de manejo de error HTTP 405 Method Not Allowed.	149
Figura 54 Respuesta HTTP 405 ante la solicitud evaluada para el manejo de errores HTTP 501 Method Not Implemented.	150
Figura 55 Análisis de manejo de error de conectividad con la base de datos.	150
Figura 56 Análisis de manejo de error de exposición de datos personales de usuario en stack trace.	151
Figura 57 Análisis de petición ante una solicitud sin token.	152

Figura 58 Análisis de solicitud ante un token modificado.....	152
Figura 59 Verificación de acceso al directorio /svg/.	154
Figura 60 Análisis de políticas de seguridad en la capa de transporte (HTTPS).....	155
<i>Figura 61 Verificación del uso del protocolo HTTPS.....</i>	<i>156</i>
Figura 62 Transmisión de credenciales durante el inicio de sesión.....	157
Figura 63 Procesamiento de solicitudes de inicio de sesión sin establecer una conexión segura	158
Figura 64 Análisis de información cifrada mediante Wireshark	158
Figura 65 Verificación de la protección del token mediante HTTPS.....	159
Figura 66 Verificación del uso del token de autenticación en solicitudes HTTPS.....	159

RESUMEN

El presente trabajo tuvo como objetivo diseñar una metodología de seguridad informática basada en OWASP para identificar, evaluar y mitigar vulnerabilidades en el sistema ViTECED, plataforma destinada al monitoreo de información biomédica. La metodología se estructuró en cinco fases: recolección de información, identificación de vulnerabilidades, remediación, reevaluación y elaboración del informe. Se emplearon OWASP Testing Guide para la aplicación web, OWASP MASVS y MASTG para la aplicación móvil. Durante la evaluación se analizaron la arquitectura, los activos críticos, la superficie de ataque y los componentes web, móvil, APIs, servicios y bases de datos. Se consolidaron 33 pruebas principales de seguridad, conformadas por diferentes subpruebas o verificaciones técnicas, de las cuales 14 presentaron uno o más hallazgos, equivalentes al 42,42 %, y 19 no presentaron hallazgos, correspondientes al 57,58 %. De las 14 vulnerabilidades identificadas, 11 fueron corregidas completamente, 1 fue parcialmente mitigada y 2 permanecieron pendientes. La reevaluación determinó una tasa de mitigación del 78,6 %, una tasa de persistencia del 21,4 %, una reducción absoluta de 11 vulnerabilidades y una reducción relativa del 78,6 %. La metodología propuesta permitió realizar una evaluación de seguridad al sistema ViTECED permitiendo identificar y mitigar vulnerabilidades.

Palabras claves: Seguridad informática, OWASP, vulnerabilidades.

ABSTRACT

The objective of this work was to design a computer security methodology based on OWASP to identify, evaluate and mitigate vulnerabilities in the ViTECED system, a platform for monitoring biomedical information. The methodology was structured in five phases: information collection, identification of vulnerabilities, remediation, re-evaluation and preparation of the report. OWASP Testing Guide was used for the web application, OWASP MASVS and MASTG for the mobile application. During the evaluation, the architecture, critical assets, attack surface and web and mobile components, APIs, services and databases were analyzed. 33 main safety tests were

consolidated, made up of different subtests or technical verifications, of which 14 presented one or more findings, equivalent to 42.42%, and 19 did not present findings, corresponding to 57.58%. Of the 14 vulnerabilities identified, 11 were completely fixed, 1 was partially mitigated, and 2 remained pending. The reassessment determined a mitigation rate of 78,6%, a persistence rate of 21,4%, an absolute reduction of 11 vulnerabilities, and a relative reduction of 78,6%. The proposed methodology allowed a security evaluation to be carried out on the ViTECED system, allowing vulnerabilities to be identified and mitigated.

Keywords: Information security, OWASP, vulnerabilities.

INTRODUCCIÓN

Con el crecimiento acelerado de la tecnología la seguridad de la información cumple un papel importante, ya que gran parte de las actividades diarias dependen del uso de herramientas digitales[1]. Aunque ha traído numerosos beneficios, también ha abierto numerosas puertas donde la información puede ser expuesta, alterada o utilizada sin autorización [2]. Como consecuencia, los problemas relacionados con la pérdida de datos y el acceso indebido han aumentado de forma considerable[3], [4]. Según informes de seguridad de aplicaciones de Fortinet y el reporte anual de Digital.ai confirman que los incidentes más frecuentes se dan en aplicaciones de juegos, servicios financieros y plataformas de salud[5] [6].

En el ámbito de la salud, esta situación resulta especialmente delicada. Porque con el incremento del uso Internet de las Cosas Médicas (IoMT), llega el desafío para proteger los datos sensibles que viajan por el medio[7]. Estos datos biomédicos, al contener detalles personales de las personas, se convierten en el principal objetivo, y su divulgación puede comprometer seriamente la privacidad y el manejo adecuado de la atención médica de miles de pacientes en todo el mundo [8].

En el año 2025, según informes recientes, el 92% de las organizaciones sanitarias a nivel mundial sufrieron al menos un incidente de seguridad. Derivando la exposición de más de 276 millones de registros médicos de pacientes [9]. Además, el sector sanitario enfrentó un promedio de 2465 intentos de ataques semanales lo que representa un aumento interanual del 17% en comparación con años anteriores [10]. Estos eventos no solo implican el robo de información sensible, sino que provocaron retrasos críticos en cirugías y alteraciones directas en los tratamientos médicos, demostrando que la falta de medidas de protección sólida en sistemas conectados sigue siendo un punto crítico de vulnerabilidades en la atención de salud moderna [11].

El estudio [12], indica que los principales ataques que pone en riesgo estos sistemas son; el robo de datos de otros usuarios mediante la interceptación de pequeños programas maliciosos en las páginas (Malware), la introducción de códigos maliciosos (Inyección SQL), los intentos repetidos de accesos aprovechando errores de diseños (Fuerza bruta), los fallos de seguridad en la conexión entre el celular y los servidores (Man-in-the-Middle), él envió de mensajes falsos para engañar a los trabajadores (Phishing), el

bloqueo de equipos mediante software de extorsión (Ransomware) y uso indebido de datos personales de pacientes, afectando directamente la atención sanitaria y la confianza de la población [13], [14], [15].

Además, informes sobre ataques ransomware en empresas del sector salud resaltan que los incidentes en aplicaciones web y sistemas médicos suelen provocar retrasos en operaciones, cancelación de citas y aumento de costos, lo que obliga a las instituciones a reforzar la protección de sus plataformas digitales y a revisar con más rigor quién entra y qué puede hacer dentro de ellas [16] [17].

Investigaciones sobre aplicaciones móviles en salud señalan que muchas de estas aplicaciones no cifran bien la información, comparten datos con terceros sin control y carecen de controles claros para garantizar que solo el paciente y el personal autorizado puedan acceder a la información médica [18]. El experto Cristian Pillajo, especialista en ciberseguridad y hacking ético, mencionan que las aplicaciones web y móvil biomédicas deben diseñarse con revisiones de seguridad desde el inicio, pruebas frecuentes y normas claras para proteger datos sensibles, porque los incidentes en este tipo de aplicaciones no solo comprometen la privacidad, sino que pueden afectar directamente la atención médica y el bienestar de los pacientes [19].

En el proyecto de investigación multidisciplinar “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas”, liderado por el grupo de investigación Tecnología, Ciencia y Educación (TECED) de la Facultad de Sistemas y Telecomunicaciones, se cuenta con un sistema biomédico para monitorear los signos vitales de los pacientes, llamado ViTECED.

El sistema se encuentra en una fase de pruebas; sin embargo, durante su desarrollo se priorizó en el diseño del sistema y selección de protocolos de comunicación para la interconexión con los dispositivos IoT, sin considerar adecuadamente los aspectos de seguridad, a pesar de que se trata de un sistema que maneja datos de pacientes. Por ello, en este trabajo se planteó como objetivo: diseñar una metodología de seguridad informática basada en OWASP para evaluar las vulnerabilidades del sistema que integra una aplicación móvil y una web.

El presente documento está dividido en dos capítulos. En el primer capítulo se explica el motivo por el cual se inició la propuesta; además, se definen los objetivos que se desean

cumplir, se justifica la importancia del proyecto y se establece el alcance de la investigación. En el segundo capítulo se describen la base teórica, el contexto y los conceptos principales que respaldan el desarrollo del proyecto. También se presentan los resultados obtenidos después de aplicar la metodología de seguridad, así como la comparación del estudio del sistema antes y después de realizar las mejoras, demostrando así la efectividad del trabajo realizado.

CAPÍTULO I

1. FUNDAMENTACIÓN

1.1. Antecedentes

Hoy en día, la tecnología forma parte de la vida cotidiana, ya que cada vez más actividades se realizan mediante dispositivos conectados a internet [20]. No obstante, esa conectividad también ha incrementado la exposición a amenazas cibernéticas, debido a que personas malintencionadas la aprovechan para realizar ataques dirigidos a la información y a los dispositivos de los usuarios [21].

En este contexto, los sistemas que monitorean la salud mediante dispositivos conectados enfrentan problemas de seguridad [22], convirtiéndose en un blanco fácil para los ciberdelincuentes. Esto se debe a que dichos sistemas recopilan información sensible, como datos médicos y personales, los cuales pueden ser modificados o eliminados por un atacante [23] [24]. En consecuencia, esta situación no solo afecta la privacidad de las personas, sino también puede alterar los resultados médicos que revisan los profesionales de la salud para la toma de decisiones [25].

El presente proyecto es un trabajo investigativo que surgió del proyecto de investigación multidisciplinar “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctrica” (CUP: 91870000.0000.389571), liderado por el grupo de investigación Tecnología, Ciencia y Educación (TECED) de la Facultad de Sistemas y Telecomunicaciones [26]. Actualmente cuentan con una plataforma informática para el monitoreo de los parámetros biomédicos como frecuencia cardíaca, saturación de oxígeno, temperatura corporal y el electrocardiograma. Por ello, se pretende realizar pruebas de seguridad y análisis estadísticos de vulnerabilidades, con la finalidad de cumplir con los 3 pilares de seguridad de la implementación: confidencialidad, integridad y disponibilidad del sistema.

La plataforma informática ViTECED tiene dos versiones [27] [28], y mediante los resultados obtenidos (Ver Anexo 1), actualmente se encuentra en estado de preproducción. Sin embargo, no ha pasado por un proceso de pruebas de seguridad, lo que incrementa su exposición a riesgos cibernéticos. Esta situación representa un riesgo significativo especialmente cuando se trata de información sensible como son los signos

vitales y datos de los pacientes. Otros de los problemas se encuentran en las fases de desarrollo (Ver Anexo 2), donde se priorizó la conectividad y la disponibilidad en tiempo real, pero sin integrar métodos estandarizados de seguridad en su ciclo de vida [29]. Esta ausencia de controles genera vulnerabilidades graves provocando que la plataforma se vea expuesta a amenazas que podrían comprometer tanto la continuidad operativa como la protección de los datos [30].

Si la plataforma se envía a producción sin hacer las correctas pruebas de seguridad y validaciones adecuadas, queda expuesta a diferentes tipos de ataques [31]. Entre ellos se encuentran las inyecciones SQL, Cross-Site Scripting (XSS) y otras vulnerabilidades comunes [32] [33]. Por lo general, esto sucede cuando no se ha corroborado de manera satisfactoria que todos los campos de entrada de la plataforma tengan las correctas validaciones [34]. Además, la falta de pruebas de penetración incrementa el riesgo de explotación por parte de atacantes [35], [36]. Esto puede derivar en el acceso no autorizado a la información, alteración de datos sensibles o interrupciones en el funcionamiento normal del sistema, afectando directamente la seguridad y confidencialidad de la plataforma [35], [36].

A nivel internacional, se realizó un estudio donde se evaluó la seguridad de una aplicación móvil vinculada a pagos mediante billetera digital en Indonesia, utilizando las guías OWASP MASTG y MASVS para verificar la seguridad de la aplicación [37]. Se combinaron análisis estáticos, análisis de dependencia y pruebas dinámicas, además de un enfoque de recopilación de información cualitativa. Como resultados se evaluaron 84 criterios de seguridad, de los cuales 6 fallaron. Posteriormente se determinó que el nivel de riesgo de la aplicación es medio [37]. Sin embargo, el estudio presentó limitaciones ya que no se aplicaron las medidas de mitigación propuestas, solo se realizaron pruebas en Android y no en iOS, tampoco se realizaron ataques avanzados, algunos criterios quedaron sin verificar, lo que redujo el alcance del análisis de seguridad y restringió la profundidad de los resultados obtenidos.

En Ecuador, se desarrolló una investigación cuyo objetivo fue analizar la seguridad de un servidor web del laboratorio de ciberseguridad de la Universidad Politécnica Estatal del Carchi, mediante pruebas técnicas bajo la guía OWASP. Se identificaron vulnerabilidades como inyección SQL, Cross Site Scripting, fuerza bruta, errores de configuración y fallos

en métodos HTTP, además, se efectuó una comparación entre servidores Apache y Microsoft IIS, así como una evaluación de riesgo que permitió determinar un incremento de seguridad del 60% tras la aplicación de proceso hardening [38]. No obstante, el estudio se limitó a la fase de diagnóstico y evaluación, ya que no se aplicaron todas las correcciones de seguridad propuestas, tampoco se verificó la efectividad de las mejoras aplicadas, ni se desarrolló un proceso continuo de monitoreo que permitiera garantizar la reducción sostenida de los riesgos identificados [38].

En Santa Elena, se elaboró un proyecto en el cual se realizó una evaluación de seguridad en las aplicaciones web de una Cooperativa de Ahorro y Crédito, aplicando la metodología Top 10 de OWASP 2021; para ello se realizaron pruebas pasivas y activas, lo que permitió identificar que las aplicaciones eran vulnerables a 5 de los 10 riesgos del Top Ten, destacando problemas como configuración de seguridad incorrecta, fallas de integridad de datos y software, diseño inseguro, manejo inadecuado de errores y exposición de información sensible en la URL [39]. Con base en los resultados se elaboró un informe donde se valoró la severidad de cada vulnerabilidad y se presentaron recomendaciones de mejora, pero debido al periodo corto de pruebas no se garantizó la identificación de todas las posibles vulnerabilidades, además, no se evaluó la aplicación móvil, quedando pendiente un análisis más amplio y de mayor alcance [39].

Los antecedentes analizados evidencian que la aplicación de metodologías basadas en OWASP constituye una estrategia efectiva para identificar vulnerabilidades de seguridad en aplicaciones web y móviles. Sin embargo, los estudios revisados presentan limitaciones relacionadas con el alcance de las pruebas, la cobertura de los componentes evaluados o la ausencia de procesos de reevaluación posteriores a la implementación de medidas de mitigación. En este contexto, la presente investigación propone una metodología de seguridad informática adaptada al sistema ViTECED, que integra OWASP Testing Guide v4 para la evaluación de la aplicación web, OWASP MASVS y OWASP MASTG para la aplicación móvil, el modelo OWASP SAMM para evaluar el nivel de madurez de las prácticas de seguridad. A diferencia de los trabajos analizados, la metodología considera de forma conjunta la aplicación web y los componentes asociados del sistema, estableciendo un proceso estructurado de reconocimiento, identificación de vulnerabilidades, remediación, reevaluación y elaboración del informe.

1.2. Descripción del Proyecto

El presente proyecto consiste en el diseño de una metodología de seguridad informática para evaluar las vulnerabilidades del sistema de monitoreo biomédico ViTECED, desarrollado por el grupo de investigación Tecnología, Ciencia y Educación (TECED) de la Universidad Estatal Península de Santa Elena. La metodología está orientada a establecer un proceso sistemático para identificar, evaluar, mitigar y reevaluar vulnerabilidades presentes en la aplicación web, la aplicación móvil y los componentes asociados del sistema, antes de su despliegue en un entorno de producción.

La metodología incluye el uso de herramientas de análisis de seguridad y guías reconocidas como OWASP Testing Guide v4 y OWASP MASVS v2.1.0 junto con el OWASP MASTG v1.7.0, mediante las cuales se evaluaron las debilidades del sistema, registrando la cantidad, el tipo y el nivel de riesgo de las vulnerabilidades detectadas. La metodología propuesta basada en OWASP está dividida en 5 fases relacionadas entre sí.

Fase 1. Recolección de información:

- Recopilar documentación técnica disponible del sistema con el propósito de orientar las pruebas de las fases siguientes.
- Identificar la arquitectura del sistema ViTECED y los componentes que lo conforma.
- Determinar las tecnologías utilizadas en el desarrollo e implementación del sistema.
- Medir el nivel de madurez del proceso de desarrollo seguro de la aplicación mediante el modelo OWASP SAMM (Software Assurance Maturity Model), con el fin de establecer una línea base del estado de seguridad del sistema.

Fase 2. Identificación de vulnerabilidades:

- Ejecutar pruebas de seguridad en la aplicación web con base a las recomendaciones de OWASP, para identificar las vulnerabilidades que representen riesgos significativos para la seguridad del sistema. Para ello, se evaluarán los siguientes aspectos:
 - Recolección de información
 - Pruebas de gestión y configuración e implementación
 - Pruebas de gestión de identidad

- Prueba de autenticación
- Pruebas de autorización
- Pruebas de gestión de sesiones
- Pruebas de validación de ingreso
- Manejo de errores
- Criptografía
- Prueba de lógica del negocio
- Prueba del punto de vista del cliente
- Ejecutar pruebas de seguridad en la aplicación móvil con base en OWASP MASVS y el OWASP MASTG, utilizados de manera conjunta para identificar las vulnerabilidades específicas del componente móvil del sistema. Para ello, se evaluarán los siguientes aspectos:
 - Pruebas de almacenamientos de datos
 - Pruebas de autenticación
 - Pruebas de comunicación de red
 - Pruebas de interacción con la plataforma
 - Pruebas de calidad del código
 - Pruebas de resistencia a la ingeniería inversa
 - Pruebas de privacidad
- Realizar análisis de seguridad de la aplicación mediante pruebas dinámicas (DAST) durante su ejecución y análisis estático (SAST) del código fuente para identificar vulnerabilidades, errores de programación y malas prácticas de desarrollo.
- Clasificar cada vulnerabilidad identificada según su nivel de riesgo (Crítico, Alto, Medio o Bajo), considerando su impacto potencial sobre la confidencialidad, integridad y disponibilidad de los datos biomédicos gestionados por el sistema.
- Relacionar cada vulnerabilidad identificada con su categoría de riesgo correspondiente según el OWASP Top 10 y el OWASP Mobile Top 10, con el fin de evidenciar su nivel de criticidad y prevalencia en el contexto de la industria.

Fase 3. Remediación:

- Presentar al equipo de desarrollo todas las vulnerabilidades encontradas durante el proceso de evaluación.

- Implementar las medidas de seguridad o mitigación con el fin de reducir el riesgo asociado a las vulnerabilidades identificadas.
- Registrar el estado de cada vulnerabilidad durante el proceso de remediación (pendiente, en proceso, corregida), con el fin de llevar un control formal del avance de las correcciones.

Fase 4. Reevaluación:

- Repetir las pruebas de seguridad y el análisis de las evidencias técnicas obtenidas después de aplicar las medidas de mitigación.
- Determinar, a partir de los resultados de la reevaluación, el estado de cada vulnerabilidad, estableciendo si fue corregida, parcialmente mitigada o permanece presente.
- Realizar un análisis estadístico descriptivo comparativo pretest–posttest del estado de las vulnerabilidades antes y después de la aplicación de las medidas de seguridad, utilizando frecuencias, porcentajes, tasa de mitigación, tasa de persistencia y reducción absoluta y relativa.

Fase 5. Informe

- En esta fase se debe realizar un informe técnico de los hallazgos encontrados y cuáles fueron las soluciones o medidas de mitigación.
- Presentar una comparativa del estado de seguridad del sistema antes y después de la aplicación de la metodología, destacando las mejoras obtenidas, las vulnerabilidades mitigadas y aquellas que permanezcan pendientes de corrección.

Para el desarrollo de la metodología informática del presente proyecto se utilizó la guía OWASP Testing Guide v4 y OWASP MASVS junto con el OWASP MASTG, herramientas de análisis de seguridad como SAST y DAST, y las plantillas de 90 preguntas del modelo SAMM que permite evaluar el nivel de madurez de las prácticas de seguridad del sistema ViTECED.

Este proyecto se encamina a la línea de investigación de Tecnología y Sistemas de la Información (TSI), basándose en la sub línea de Ingeniería y gestión de Tecnología y Sistemas de la Información (TSI)[40]. Debido a la necesidad de reforzar la seguridad y confiabilidad del sistema, mediante un análisis enfocado en la detección y evaluación de

vulnerabilidades, con el fin de aplicar estrategias de mitigación que aporten a la mejora de la gestión tecnológica.

1.3. Objetivos del Proyecto

Objetivo General

Diseñar una metodología de seguridad informática para la identificación, evaluación y mitigación de vulnerabilidades en el sistema informático ViTECED.

Objetivo Especifico

- ✓ Identificar la arquitectura y los componentes del sistema informático ViTECED, identificando los activos críticos y su superficie de ataque.
- ✓ Diseñar una metodología de seguridad informática basada en la guía OWASP, adaptada al contexto del sistema ViTECED.
- ✓ Aplicar la metodología propuesta para identificar y clasificar las vulnerabilidades de seguridad presentes en la plataforma web, móvil y componentes asociados.
- ✓ Implementar medidas de mitigación orientada a reducir las vulnerabilidades detectadas, de acuerdo con su nivel de riesgo.
- ✓ Evaluar la efectividad de la metodología mediante un análisis estadístico comparativo del antes y después de la aplicación de las medidas de seguridad.

1.4. Justificación del Proyecto

La seguridad informática se establece como un pilar fundamental en el desarrollo de aplicaciones digitales, especialmente en el ámbito de la salud, donde se manejan datos clínicos altamente sensibles [41]. En este sentido, la integración de medidas de protección permitió que la información médica fuera gestionada de manera eficiente, confiable y segura, fortaleciendo así la organización y el correcto funcionamiento de los servicios de salud [42]. Bajo este enfoque, resultó indispensable la aplicación de la metodología OWASP como un marco de referencia para garantizar la seguridad en el desarrollo de software [43].

Esta metodología, junto con las herramientas de análisis SAST y DAST, facilitó la identificación de puntos débiles de la aplicación web y móvil antes de su implementación en un entorno real [43]. De este modo, permite evaluar el nivel de exposición de los datos

de los pacientes y se establecieron medidas que aseguraron su integridad, confidencialidad y disponibilidad.

La protección de los datos no solo representó una responsabilidad técnica, sino también ética y legal, ya que cualquier filtración o manipulación indebida pudo afectar directamente la privacidad y la salud de las personas [44] [45]. Por ello, la adopción de la metodología de OWASP y el uso de herramientas SAST y DAST representaron un avance significativo para el proyecto de “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas”, generando múltiples beneficios que contribuyeron al fortalecimiento de la seguridad del sistema de monitoreo biomédico ViTECED, principalmente en la protección de datos de los usuarios gestionados en la plataforma informática. Al aplicar las medidas de seguridad ayuda a fortalecer el sistema y evita la fuga de información en especial cuando se trata de un sistema con información sensible y relacionado con la salud.

De esta manera, disponer de un sistema con mayores niveles de seguridad fortaleció la confianza de los usuarios y pacientes al proporcionar un entorno más confiable para el uso de la plataforma. Asimismo, contribuyó a reforzar la relación entre la tecnología y la sociedad, al generar mayor credibilidad, fomentar un uso responsable de la plataforma y favorecer su aceptación como una herramienta para el monitoreo biomédico.

El presente proyecto está alineado al Plan Nacional de Desarrollo Ecuador no se Detiene 2025-2029, poniendo atención en el eje Institucional, el cual detalla lo siguiente:

Eje: Institucional [46].

Objetivo 8: Fortalecer la institucionalidad pública de forma eficiente, transparente y participativa [46].

Política 8.3: Impulsar la transformación digital del Estado, la adopción del modelo de Estado Abierto, la protección de la información, con un entorno digital seguro y confiable en todos los niveles de gobierno, así como la integridad pública y la lucha contra la corrupción, que promueva la gestión pública eficiente, inclusiva, transparente y participativa [46].

1.5. Alcance del Proyecto

La investigación tiene un alcance descriptivo y un diseño de investigación no experimental de tipo cuasiexperimental, basado en un estudio de caso [47], [48], [49]. Es descriptivo [50] [51], porque se enfoca en explicar y detallar cómo se encuentra actualmente el Sistema de Monitoreo Biomédico ViTECED, identificando las vulnerabilidades y describiendo el nivel de seguridad que presenta. No se busca modificar el sistema en esta etapa inicial, sino comprender su estado real y las debilidades que existen.

Se considera cuasiexperimental [52], [53] porque se realiza una comparación entre dos momentos: antes de aplicar mejoras de seguridad y después de aplicarlas. En este proceso se interviene el sistema aplicando medidas de seguridad basadas en buenas prácticas, luego se revisan nuevamente los resultados para observar que cambios se produjeron. Sin embargo, no se trabaja con grupos separados ni condiciones controladas, sino con el mismo sistema en distintos momentos.

Por otro lado, se basa en un estudio de caso [54] porque, el análisis se centra únicamente en el sistema ViTECED, que es un caso específico dentro de un entorno académico. Esto permite realizar un estudio más detallado del sistema, considerando sus diferentes partes como la aplicación web, la aplicación móvil, la base de datos y los dispositivos conectados.

Además, es importante aclarar que el estudio se centra en el análisis del estado actual del Sistema de Monitoreo Biomédico ViTECED y no incluye la implementación del sistema en un entorno clínico real, ni el desarrollo de nuevas funcionalidades de monitoreo biomédico; el objetivo principal es asegurar la confidencialidad, integridad y disponibilidad de los datos durante la fase de pruebas y validaciones internas.

El desarrollo de la presente investigación se estructura en 5 fases principales, las cuales permiten abordar de manera sistemática la identificación, análisis y mitigación de vulnerabilidades en el Sistema de Monitoreo Biomédico ViTECED, considerando tanto la aplicación web, la aplicación móvil y la infraestructura asociada.

En la fase 1: se realiza el reconocimiento del sistema ViTECED con el propósito de comprender su funcionamiento general, su estructura y los elementos que los conforma.

Para ello, se identifican los componentes principales del sistema, las tecnologías utilizadas durante su desarrollo y la documentación técnica disponible, información que servirá como base para orientar las actividades de las fases posteriores.

Además, se evalúa el nivel de madurez de las prácticas de seguridad del proceso de desarrollo mediante el modelo OWASP SAMM (Software Assurance Maturity Model) (Ver Anexo 3). Este modelo fue seleccionado debido a que proporciona un enfoque orientado específicamente a la evaluación de la seguridad durante el ciclo de desarrollo del software, permitiendo establecer una línea base sobre el estado de seguridad del sistema. La evaluación considera cinco áreas fundamentales: gobernanza, diseño, verificación, implementación y operación [55].

Aunque existen otros marcos de referencia para evaluar aspectos relacionados con la seguridad, como el NIST Cybersecurity Framework (CSF) [56], el presente trabajo únicamente se emplea el modelo OWASP SAMM por su mayor adecuación a los objetivos de esta investigación y al contexto del sistema ViTECED.

En la fase 2: se procede a la identificación detallada de vulnerabilidades presentes en el sistema ViTECED. Para ello se aplican las guías OWASP Testing Guide v4 para aplicaciones web, mientras que para la aplicación móvil se emplea de manera conjunta OWASP MASVS (Estándar de Verificación de Seguridad en Aplicaciones Móviles) y MASTG (Guía de Prueba de Seguridad en Aplicaciones Móviles), con el fin de evaluar los diferentes aspectos relacionados con la seguridad de cada componente del sistema.

Como parte del proceso, se realizan análisis del código fuente y del comportamiento de la aplicación durante su ejecución, con el propósito de identificar errores de programación, configuraciones inadecuadas y otras debilidades que puedan afectar el funcionamiento seguro del sistema. Para ello, se emplean herramientas de análisis estáticos (SAST), que permiten detectar problemas en el código sin necesidad de ejecutar la aplicación, y las herramientas de análisis dinámico (DAST), utilizadas para evaluar el comportamiento del sistema mientras se encuentra en funcionamiento y detectar posibles vulnerabilidades durante su operación.

Una vez identificadas las vulnerabilidades, cada una es clasificada de acuerdo con su nivel de riesgo (Crítico, Alto, Medio o Bajo), considerando el posible impacto que podría generar sobre la confidencialidad, integridad y disponibilidad de la información.

Finalmente, los hallazgos se relacionan con las categorías establecidas por el OWASP Top 10 y el OWASP Mobile Top 10, permitiendo identificar los riesgos más relevantes presentes en la aplicación.

En la fase 3: se presenta al equipo de desarrollo el reporte de vulnerabilidades identificado durante la evaluación de seguridad, detallando la descripción de cada hallazgo, su nivel de riesgo y los controles de seguridad propuestas para su corrección. Posteriormente, se realizan las correcciones correspondientes mediante modificaciones en el código, ajustes de codificación u otras acciones necesarias para reducir los riesgos identificados. Durante todo este proceso se registra el estado de cada vulnerabilidad, permitiendo llevar un seguimiento del avance de las actividades de remediación hasta su corrección.

En la fase 4 se comprueba la efectividad de las acciones implementadas sobre las vulnerabilidades detectadas durante la evaluación inicial. Para ello, se realiza una nueva evaluación al sistema y se analizan los resultados obtenidos después de las aplicaciones de las medidas de mitigación. Los resultados se comparan el estado inicial de las vulnerabilidades, lo que permite determinar los cambios producidos y establecer su condición posterior, considerando las acciones implementadas y el nivel de riesgo. Posteriormente, se emplea un análisis estadístico descriptivo comparativo pretest-posttest para cuantificar los cambios observados mediante frecuencias, porcentajes, tasa de mitigación, tasa de persistencia y reducción absoluta y relativa.

Y por último en la fase 5 se realiza la documentación detallada de todos los hallazgos encontrados y cuáles son las medidas de remediación y las acciones efectuadas para corregir esos fallos de seguridad encontrados.

Finalmente, se presenta una comparación del nivel de seguridad del sistema antes y después de la aplicación de la metodología, destacando las vulnerabilidades corregidas, aquellas que permanecen pendientes y las mejoras alcanzadas en el sistema.

La investigación se limita a la evaluación del sistema ViTECED en un entorno de preproducción, utilizando la versión disponible durante el desarrollo del estudio. No contempla pruebas en un entorno clínico real, evaluaciones sobre otras plataformas distintas a las implementadas en el sistema ni el desarrollo de nuevas funcionalidades. Asimismo, los resultados corresponden a la versión evaluada y no garantizan la ausencia de vulnerabilidades en futuras actualizaciones del sistema.

1.6. Metodología del Proyecto

1.6.1. Metodología de la investigación

Contexto de la investigación

La evaluación de seguridad se realizó en el sistema de monitoreo biomédico ViTECED, desarrollado por el grupo de investigación TECED de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena (UPSE). Este sistema está diseñado para monitorear parámetros biomédicos, como la frecuencia cardíaca, la temperatura corporal, la saturación de oxígeno y el electrocardiograma (ECG), mediante una infraestructura compuesta por una aplicación web, una aplicación móvil, dispositivos IoT, un servidor, base de datos y diversas interfaces de programación de aplicaciones (API). Actualmente, la plataforma es utilizada por docentes, estudiantes e investigadores que trabajan con la plataforma dentro de la universidad.

Hoy en día, el sistema está en fase de pruebas y funciones de forma interna, permitiendo la conexión en tiempo real con los dispositivos. Sin embargo, todavía no tiene controles de seguridad que protejan adecuadamente la información biomédica. Por esta razón, este entorno es el más adecuado para aplicar la metodología basada OWASP, y herramientas SAST y DAST, ya que permite detectar vulnerabilidades, analizar riesgos y tomar acciones antes de que la plataforma sea usada en un entorno real de salud.

Tipo y método de investigación

Esta investigación tiene un enfoque mixto, de tipo aplicada [57][58]. El enfoque cuantitativo se basó en el análisis estadístico de los datos obtenidos, utilizando métodos descriptivos para interpretar la información numérica relacionada con las vulnerabilidades del sistema ViTECED [59]. En este sentido, se realizó un análisis estadístico descriptivo comparativo pretest–posttest, considerando como pretest el estado inicial de las vulnerabilidades identificadas y como posttest el estado registrado después de la aplicación de las medidas de mitigación. Para la comparación se utilizaron frecuencias, porcentajes, tasa de mitigación, tasa de persistencia y reducción absoluta y relativa, permitiendo cuantificar los cambios observados en las vulnerabilidades identificadas.

Por otro lado, el enfoque cualitativo se fundamentó en la recolección de información mediante entrevistas, observación directa y estudio de caso real [60]. Las entrevistas permitieron obtener información detallada sobre las experiencias y percepciones relacionadas con el sistema. La observación directa facilitó el análisis del comportamiento del sistema en su entorno real de funcionamiento. Finalmente, el estudio de caso permitió realizar un análisis profundo del sistema ViTECED dentro de su contexto específico, lo que ayudó a comprender de manera integral su funcionamiento y sus debilidades en materia de seguridad.

El enfoque se considera aplicado [57], porque el estudio se realiza sobre un sistema real con el propósito de reforzar su seguridad. De manera complementaria, se interpretarán los resultados para identificar los riesgos más relevantes, lo que aporta un componente cualitativo secundario que ayuda a contextualizar los hallazgos cuantitativos.

Para alcanzar el objetivo del proyecto, se emplearán el siguiente método [61]:

- **Método aplicado:** consiste en la ejecución práctica de medidas de seguridad orientadas a fortalecer la protección del sistema ViTECED. Aplicar parches de seguridad y la implementación de medidas de mitigación con el propósito de reducir las vulnerabilidades identificadas durante la evaluación.

Población y muestra

Población

La población del proyecto está conformada por todos los componentes que integran el sistema ViTECED y que son objeto de evaluación durante las pruebas de seguridad. Estos incluyen la plataforma web y móvil, APIs, servicios que soportan el funcionamiento del sistema y las bases de datos. Estos componentes contribuyen el conjunto total de actores y recursos tecnológico sobre el cual se desea evaluar la seguridad y aplicar las medidas de mitigación.

Elementos	Descripción	Cantidad
Integrantes del grupo de investigación TECED	Miembros que conforman el grupo de investigación del sistema ViTECED	23
Front-end / Aplicaciones	Aplicaciones cliente que permiten la interacción de los usuarios con el sistema (web y móvil).	2

Elementos	Descripción	Cantidad
Back-end / APIs	Interfaces encargadas de la comunicación e intercambio de información entre los diferentes componentes del sistema.	3
Servicios	Servicios que ejecutan procesos especializados para el funcionamiento de la plataforma.	2
Base de datos	Sistemas de almacenamiento utilizados para la gestión de la información biomédica.	2

Tabla 1 Componentes del Sistema ViTECED

Muestra

Se aplicó un muestreo no probabilístico por conveniencia [62] [63], seleccionando los elementos más críticos y accesible del sistema, aquellos que tengan mayor relevancia para la evaluación de seguridad. Esto permitió concentrarse en los componentes que, de presentar vulnerabilidades podrían afectar de manera relevante la confidencialidad, integridad y disponibilidad de los datos biomédicos.

La muestra incluye los siguientes componentes:

- Módulos principales de la aplicación web y móvil considerados.

Módulos	Componentes
1: General	Autenticación
	Perfil
	Reportes
2: Rol paciente	Médico
	Telemedicina
	Biomédico
3: Rol médico o especialista	Paciente
	Asistente
4: Rol administrador	Administrador
5: Rol asistente del médico	Enfermería

Tabla 2 Módulos del Sistema.

- APIs evaluadas

API	Cantidad
General	1
Compatible con HL7 Fhir	1
Para consultas Mongo DB	1

Tabla 3 APIs principales del Sistema.

- Servicios evaluados

Servicios	Cantidad
Servicio generación de alertas con lógica difusa	1
Servicio análisis morfológico con ECG	1

Tabla 4 Servicios del Sistema.

- Base de datos evaluados

Base de dato	Cantidad
MongoDB	1
MariaDB	1

Tabla 5 Bases de datos del Sistema

- Usuarios considerados para la realización pruebas

Integrante	Cantidad
Integrantes del grupo de investigación TECED	4 personas

Tabla 6 Usuarios considerados para la prueba

1.6.2. Beneficiarios del Proyecto

Los principales beneficiarios de este proyecto son el grupo de investigación Tecnología, Ciencia y Educación (TECED) de la Facultad de Sistemas y Telecomunicaciones, específicamente en el proyecto de investigación multidisciplinar “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas ya que, a través de la evaluación de vulnerabilidades y la aplicación de medidas de seguridad, se contribuye al fortalecimiento de la plataforma informática de monitoreo biomédico ViTECED.

1.6.3. Variables

En el presente proyecto se establece como variable principal el nivel de seguridad del sistema ViTECED, el cual se evalúa a partir de la identificación y mitigación de las vulnerabilidades presentes en el sistema. Esta variable permite analizar el estado de protección de la información biomédica gestionada por el sistema, considerando los principios de confidencialidad, integridad y disponibilidad de la información.

La evaluación de esta variable se desarrolló mediante tres fases, las cuales permitieron determinar el estado inicial de seguridad, aplicar las medidas de mitigación y verificar los cambios obtenidos posteriormente.

Fase de medición	Recurso utilizado	Variable observada	Datos obtenidos
Evaluación inicial	Evaluación de vulnerabilidades del sistema siguiendo la metodología basada OWASP	Estado inicial de seguridad del sistema	Número y tipo de vulnerabilidades detectadas
Intervención	Aplicación de medidas de mitigación basadas en las recomendaciones de OWASP	Medidas correctivas implementadas	Registros de medidas de mitigación aplicadas
Evaluación final	Verificación del estado de las vulnerabilidades después de la aplicación de las medidas de mitigación y análisis estadístico descriptivo comparativo pretest–posttest.	Estado de las vulnerabilidades después de la aplicación de las medidas de seguridad.	Número de vulnerabilidades corregidas, pendientes y parcialmente corregidas; frecuencias, porcentajes, tasa de mitigación, tasa de persistencia y reducción absoluta y relativa.

Tabla 7 Análisis de la variable

Método de análisis

Para el análisis se consideraron las frecuencias absolutas y porcentuales de las vulnerabilidades identificadas, así como la tasa de mitigación, tasa de persistencia,

reducción absoluta y reducción relativa. Adicionalmente, los resultados se compararon según el nivel de riesgo y las categorías OWASP asociadas a los hallazgos.

La tasa de mitigación se calculó mediante la siguiente ecuación:

$$TM = \frac{V_c}{V_i} \times 100$$

Donde:

- **TM:** Tasa de mitigación expresada en porcentaje.
- **Vc:** Número de vulnerabilidades corregidas después de aplicar las medidas de mitigación.
- **Vi:** Número total de vulnerabilidades identificadas en la evaluación inicial.

La tasa de persistencia se calculó mediante:

$$TP = \frac{V_p}{V_i} \times 100$$

Donde:

- **TP:** Tasa de persistencia expresada en porcentaje.
- **Vp:** Número de vulnerabilidades que permanecen pendientes después de la aplicación de las medidas de mitigación.
- **Vi:** Número total de vulnerabilidades identificadas en la evaluación inicial.

La reducción absoluta de vulnerabilidades se determinó mediante:

$$RA = V_i - V_f$$

Donde:

- **RA:** Reducción absoluta.
- **Vi:** Número de vulnerabilidades presentes en la evaluación inicial.
- **Vf:** Número de vulnerabilidades presentes después de la aplicación de las medidas de mitigación.

Finalmente, la reducción relativa se calculó mediante:

$$RR = \frac{V_i - V_f}{V_i} \times 100$$

Donde:

- **RR:** Reducción relativa expresada en porcentaje.
- **Vi:** Número de vulnerabilidades presentes en la evaluación inicial.
- **Vf:** Número de vulnerabilidades presentes después de la aplicación de las medidas de mitigación.

Análisis de recolección de datos

Para la recolección de datos se utilizó un muestreo por conveniencia, mediante la aplicación de entrevistas y observaciones técnicas del sistema, con el objetivo de obtener información completa sobre el funcionamiento y seguridad del sistema ViTECED.

Se realizó una entrevista dirigida a los colaboradores del proyecto “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas”, principalmente a los encargados del desarrollo del sistema, lo que permitió obtener información directa sobre cómo funciona la plataforma, que medidas de protección se han aplicado y cuáles son las principales dificultades que posee la plataforma relacionada con la seguridad.

Para esto, se entrevistó a la Ing. Angela Parra y al Ing. Gustavo Cisnero, colaborador del proyecto de investigación mencionado anteriormente, realizó la entrevista con preguntas abiertas, como se muestra en el Anexo 1. Donde se determinó que el sistema cuenta con ciertos elementos básicos de seguridad. En cuanto a la autenticación, se utilizan tokens y existe una gestión de accesos basada en roles de usuarios. Además, se mencionó que existe una política de contraseñas seguras y que los datos biomédicos son protegidos mediante el uso de protocolos como MQTT junto con cifrado TLS/SSL, y en su almacenamiento se implementa cifrado de datos sensibles dentro de la base de datos, así como mecanismos de auditoría mediante el registro de logs en una base de datos MongoDB.

No obstante, se evidencian varias debilidades que afectan la seguridad del sistema. Entre ellas, destaca la ausencia de autenticación multifactor lo que aumenta el riesgo a accesos no autorizados. De igual manera, la falta de mecanismos específicos para prevenir fugas de información representa una vulnerabilidad crítica en el manejo de datos sensibles.

Asimismo, solo se ha considerado parcialmente el análisis de vulnerabilidades del OWASP Top 10, limitándose a la inyección SQL. Además, se reconoce que muchas de las herramientas y librerías utilizadas están desactualizadas, lo que aumenta el riesgo de explotación de fallas.

Por otra parte, no se han realizados pruebas de penetración basados en un estándar o guía específico y de análisis de vulnerabilidades de forma periódica, la falta de aplicación de pruebas con herramientas técnicas SAST y DAST sobre el código o cuando la aplicación esté en funcionamiento, limitando la detección oportuna de posibles errores en el sistema.

En cuanto a la continuidad del servicio, aunque existen medidas básicas como el bloqueo por IP ante tráfico malicioso, no se dispone de un mecanismo de respaldo confiable que garantice la recuperación del sistema ante fallos o ataques. Además, si bien se guardan registros de las acciones de los usuarios, no se menciona un proceso estructurado para revisar estos registros y detectar comportamientos extraños. Incluso se identificó que una de las principales limitaciones para la implementación de controles de seguridad más avanzados es la falta de personal con experiencia especializada en el área. Sin embargo, se indicó que no existen funciones de seguridad pendientes por implementar desde la fase de diseño, lo que podría reflejar una falta de planificación o actualización en los requerimientos de seguridad del sistema frente a estándares actuales.

También se aplicó una observación técnica al sistema ViTECED. Desde el punto de vista técnico, como se muestra en el Anexo 2, se observó que el sistema requiere una clave especial (token) para que el usuario pueda acceder a ciertas partes, lo que condiciona el ingreso solo a quienes hayan iniciado sesión correctamente. Se comprobó, además, que el sistema diferencia entre distintos tipos de usuarios (médico, administrador, usuario, asistente) y que, según el tipo de usuario, se bloquea o se permite el acceso a ciertas pantallas o funciones, de manera que cada persona solo puede usar lo que le corresponde.

También se observó que, cuando los datos biomédicos se envían entre el dispositivo del usuario y el servidor, se usa un método de protección que impide que la información pueda leerse con facilidad si alguien intenta capturarla en el camino. Por otro lado, se notó que el sistema utiliza herramientas y componentes que no han sido actualizados recientemente, muchos de ellos en versiones antiguas que podrían contener problemas ya

conocidos. Además, no se encontró evidencia de pruebas periódicas de seguridad ni de revisiones automáticas del código para detectar fallos.

En general, mediante la entrevista y la observación técnica se deduce que el sistema cuenta con elementos de seguridad básicos en su funcionamiento, pero que en su diseño y configuración aún no se incorporan prácticas completas de seguridad y mantenimiento, es decir, que requiere mejoras significativas para garantizar una protección más adecuada de la información.

1.7. Metodología de Desarrollo

Para el desarrollo del presente proyecto se diseñó una metodología basada en OWASP para la evaluación de aplicaciones web y móviles. Esta metodología tiene como finalidad evaluar la seguridad del sistema de monitoreo biomédico ViTECED <https://zenodo.org/records/21956366>, identificar vulnerabilidades presentes y verificar la efectividad de las medidas de mitigación aplicadas en un entorno de preproducción. La metodología se fundamenta en los lineamientos establecidos por OWASP debido a que proporciona un marco de referencia ampliamente reconocido para la evaluación de la seguridad de aplicaciones [64]. Asimismo, permite integrar técnicas de análisis estáticos (SAST), análisis dinámicos (DAST) y el modelo de madurez OWASP SAMM, favoreciendo un proceso de evaluación estructurado y acorde con los objetivos de la investigación.

El presente proyecto se enfocó en las siguientes fases:

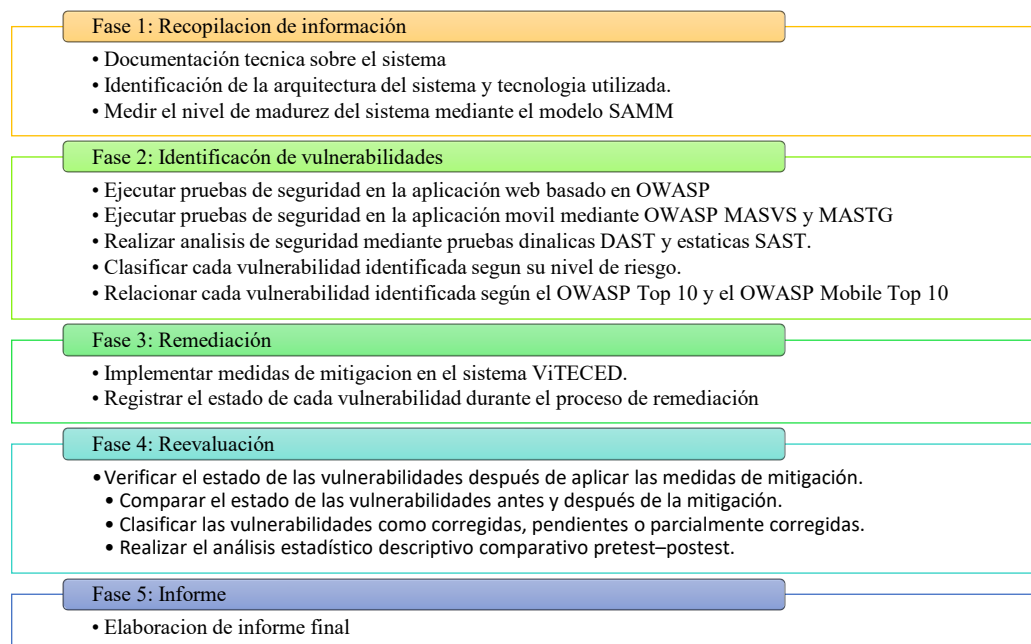


Figura 1 Metodología de pruebas de penetración con OWASP.

Fase 1. Recolección de información: En esta fase se recopiló toda la información necesaria sobre el sistema ViTECED antes de iniciar con las pruebas de seguridad. Para ello, se recopiló la documentación técnica y los manuales disponibles que permiten comprender la arquitectura del sistema, sus componentes y la lógica de su funcionamiento.

Asimismo, se obtuvo el código fuente de los diferentes componentes del sistema incluyendo las APIs, las aplicaciones frontend y backend, los servicios, las bases de datos y los dominios utilizados. Por otro lado, se evaluó el nivel de madurez de las prácticas de seguridad mediante el modelo SAMM. Esta evaluación permitió analizar el estado de las prácticas de seguridad implementadas en el proceso de desarrollo del sistema, considerando los dominios de gobernanza, diseño, implementación, verificación y operación.

Fase 2. Identificación de vulnerabilidades: Esta fase comprendió la ejecución sistemática de pruebas de seguridad sobre el sistema ViTECED, tomando como base el OWASP Testing Guide v4 para la aplicación web y el OWASP MASVS junto con el MASTG para la aplicación móvil. El objetivo es identificar y analizar posibles fallas o

puntos débiles que puedan afectar la seguridad del sistema, para ello, se utilizaron herramientas de análisis estático (SAST) y dinámico (DAST).

Posteriormente, se clasificaron las vulnerabilidades identificadas de acuerdo con su nivel de riesgo. Después se relacionó con las categorías correspondientes del OWASP Top 10 y del OWASP Mobile Top 10, con el fin de determinar su impacto y alinearlos con los principales riesgos de seguridad reconocidos para aplicaciones web y móviles. El resultado de esta fase es un listado preliminar de vulnerabilidades que sirve para priorizar las acciones de mitigación.

Fase 3. Remediación: Esta fase estuvo orientada en la realización de las correcciones de las vulnerabilidades identificadas durante la fase anterior. Primer lugar, se presentó al equipo de desarrollo un reporte con todos los hallazgos detectados en el sistema ViTECED, incluyendo su descripción, nivel de riesgo y las recomendaciones de mitigación correspondiente.

Fase 4. Reevaluación: Esta fase consiste en volver a aplicar la prueba donde se encontró la vulnerabilidad para confirmar que ya está corregida y posterior a eso se realiza el análisis estadístico-descriptivos con el fin de observar los resultados del antes y después de aplicar las medidas de seguridad.

Fase 5. Informe: Durante esta fase se lleva a cabo la elaboración del informe, donde se documentan todos los resultados obtenidos a lo largo del proceso de evaluación de seguridad del sistema de monitoreo biomédico ViTECED, incluyendo las vulnerabilidades identificadas, las medidas de mitigación implementadas y los resultados de la reevaluación. Asimismo, se refleja el análisis comparativo del estado de seguridad del sistema ViTECED antes y después de la aplicación de la metodología, destacando las vulnerabilidades mitigadas, aquellas que permanecieron pendientes de corrección y las mejoras alcanzadas en el nivel de seguridad del sistema.

CAPÍTULO II

2. PROPUESTA

2.1. Marco Contextual

En la actualidad, el uso de herramientas digitales para el monitoreo de la salud ha adquirido gran importancia, ya que permite obtener y analizar información relevante sobre el estado de los pacientes. Estas tecnologías facilitan el seguimiento continuo de distintos indicadores de salud y apoyan la toma de decisiones por parte de los profesionales. Sin embargo, el manejo de este tipo de información implica ciertos riesgos, debido a que se trata de datos personales que deben ser protegidos para evitar accesos no autorizados o un uso inadecuado.

El grupo de investigación Tecnología, ciencia y Educación de la Facultad de Sistemas y Telecomunicaciones de la Universidad Estatal Península de Santa Elena se encuentran en la fase final de proyecto “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas” (CUP:91870000.0000.389571), donde su objetivo principal es crear un dispositivo que adquiera señales bioeléctricas y este envíe a una plataforma que permita el monitoreo remoto de los signos vitales de los pacientes, para ello se considera una aplicación web y móvil donde hay varios tipos de rol entre ellos paciente, médico y administrador por esta razón es necesario que el sistema este protegido y evitar fugas de información o vulnerabilidades.

De esta manera, se busca contribuir al fortalecimiento del sistema, promoviendo un uso más seguro y confiable de la tecnología en el ámbito de la salud.

2.2. Marco Conceptual

2.2.1. Seguridad Informática

La seguridad informática es el conjunto de acciones que se realizan para proteger los sistemas y la información de posibles riesgos. Su objetivo es evitar que personas no autorizadas puedan entrar a causar daño. También busca prevenir la pérdida o el mal uso de los datos. De esta manera, se cuida que la información se mantenga segura en todo momento. Esto es importante para que las organizaciones funcionen de forma correcta y confiable [65].



Figura 2 Triada CID: Confidencialidad, Integridad y Disponibilidad

Fuente: Adaptado de [66].

Confidencialidad: Se refiere a que la información solo debe ser vista por personas que tienen permiso. Esto ayuda a evitar que otros accedan a datos importantes sin autorización. También protege la privacidad de la información personal o sensible. Mantener la confidencialidad es clave para generar confianza en los sistemas. Así se evita que la información sea mal utilizada por terceros [67].

Integridad: significa que la información debe mantenerse correcta y sin cambios indebidos. Los datos no deben ser alterados por personas que no tienen permiso. Esto permite que la información sea confiable y útil en todo momento. Además, ayuda a que los sistemas funcionen de manera adecuada. Mantener la integridad evita errores y problemas en los procesos [67].

Disponibilidad: Se refiere a que la información y los sistemas estén accesibles cuando se necesiten. Es importante que no existan interrupciones que impidan su uso. Esto asegura que las actividades puedan continuar sin problemas. También ayuda a brindar un buen servicio a los usuarios. Mantener la disponibilidad evita retrasos y pérdidas en la organización [67].

Para evaluar el nivel de riesgo de una vulnerabilidad se utilizó la fórmula:

$$\text{Riesgo} = \text{Probabilidad} \times \text{Impacto}$$

2.2.2. Pentesting o pruebas de penetración

Son evaluaciones en las que se simulan intentos de acceso no autorizados a un sistema, con el fin de identificar fallas de seguridad y corregirlas antes de que puedan ser

aprovechadas por personas malintencionadas [68]. Estas pruebas pueden realizarse de diferentes formas, como pruebas de caja blanca, negra y gris, dependiendo del nivel de información disponible [69].

Caja negra: Es un tipo de pruebas donde la persona que evalúa no tiene información previa del sistema, por lo que actúa como alguien externo que intenta ingresar sin conocer cómo funciona por dentro [70]. Esto permite ver cómo respondería el sistema ante intentos reales de acceso no permitido, ayudando a identificar que tan bien está protegido desde el exterior [71], de esta forma se puede detectar fallas que podrían ser aprovechadas por personas ajenas.

Caja blanca: En este caso, la persona que evalúa conoce todo sobre el sistema y tiene acceso a su funcionamiento, estructura y datos importantes, lo que le permite revisar cada parte con muchos detalles [67]. Así puede encontrar errores o debilidades que no son visibles fácilmente dentro de un sistema, por ende, este tipo de prueba ayuda a mejorar el sistema desde adentro [68], [72].

Caja gris: En este tipo de prueba combina las dos pruebas anteriores [73], la persona que evalúa tiene solo una parte de la información del sistema, por lo que actúa como un usuario que tiene cierto acceso, pero no conoce todo sobre el sistema. Esto permite analizar tanto el comportamiento desde fuera como desde dentro y ayuda a detectar problemas que pueden ocurrir en situaciones intermedias [74].

2.2.3. Herramientas para análisis de seguridad

SAMM (Software Assurance Maturity Model)

El modelo SAMM fue creado para medir el nivel de madurez de la seguridad en el desarrollo de software [75]. Su principal función es ayudar a las organizaciones a identificar que tan seguros son sus procesos y qué acciones debe implementar para fortalecer la seguridad de sus aplicaciones [76]. Además, permite establecer controles y medir el nivel de madurez alcanzado y avanzar de manera planificada hacia procesos más confiables [77].

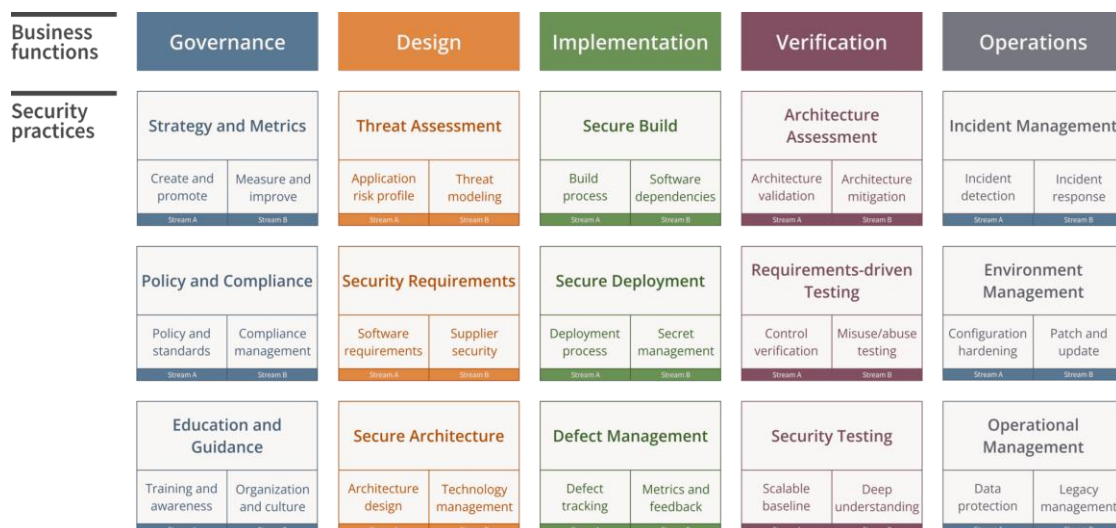


Figura 3 Componentes funcionales y prácticas de seguridad del modelo SAMM.

Fuente: Adaptado de [78].

OWASP (Open Web Application Security Project)

OWASP es una comunidad internacional que ofrece información y recursos para apoyar la creación de aplicaciones más seguras [79]. Su función principal es brindar los mecanismos necesarios para identificar, evaluar y mitigar vulnerabilidades que pueda tener una aplicación web o móviles para reducir fallos que puedan comprometer el funcionamiento completo del sistema [80].

SAST (Static Application Security Testing)

Las pruebas de seguridad estáticas de aplicaciones consisten en revisar el código de una aplicación o sistema sin necesidad de ejecutarlo, lo que permite analizar con detalle cómo está estructurado [81]. Su principal función es identificar y corregir los errores en la programación que podrían afectar la seguridad del software, antes del que el sistema o aplicación sea puesta en funcionamiento real [82].

DAST (Dynamic Application Security Testing)

Las pruebas dinámicas de seguridad de aplicaciones consisten en evaluar el comportamiento de una aplicación mientras está en funcionamiento [83]. No se necesita acceder al código fuente, ya que se evalúa desde fuera, simulando como un atacante o usuario interactuaría con ella [84]. Este tipo de pruebas permite detectar problemas que solo aparecen durante el uso, como errores en la comunicación con la base de datos o fallas en la gestión de sesiones [85].

2.2.4. Herramientas para prueba de seguridad

Kali linux: Es un sistema operativo basado en Linux que ya viene preparado con muchas herramientas profesionales y prácticas en un solo lugar para revisar la seguridad de páginas web y sistemas [86]. Se utiliza principalmente para hacer pruebas de seguridad ofensivas, defensivas y encontrar fallas de forma ética antes de que alguien más las aproveche con malas intenciones [87].

Duck Duck Go: Es un motor de búsqueda de internet que ayuda a encontrar información relevante sobre un dominio, empresa o sistema minimizando el rastreo de la actividad de búsqueda [88]. Es especialmente útil cuando se desea realizar investigaciones de manera anónima, evitando que otra persona o entidad puedan rastrear fácilmente consultas realizadas en lianas, garantizando así una navegación más segura y privada [89].

Bing: Es un motor de búsqueda en internet similar a otros que se usa para recopilar información pública sobre un sitio, su historia, servicios o datos que pueden dar pista sobre su funcionamiento [90]. Es útil como fuente adicional por que a veces muestra detalles distintos a otros buscadores, lo que permite complementar la investigación con mayor contexto y precisión informativa en línea actualizada [91].

Google Hacking: Es una técnica que consiste en usar operadores de búsqueda avanzada de Google como “site”, con el fin de localizar contenidos específicos como formularios, archivos o páginas que no deberían estar tan visibles públicamente [92]. Permite identificar qué información de una organización puede encontrarse fácilmente en internet, incluyendo documentos o configuraciones que se suponía estarían ocultas, lo cual puede evidenciar posibles fallas de exposición [93].

Curl: Es una herramienta de línea de comandos que permite enviar peticiones HTTP, HTTPS o FTP a un sitio web. Su función es comprobar si una página funciona correctamente, si redirige bien a otra URL o si devuelve mensajes extraños o código de error cuando se le envían ciertos datos [94]. Es muy útil para entender cómo se comunican las aplicaciones sin necesidad de usar el navegador [95].

Whatweb: Es una herramienta rápida y muy útil para identificar qué tipo de tecnología usa una página web, tanto como el lenguaje de programación o sistema gestor de contenido está detrás de ella, permitiendo conocer cómo está construida internamente, lo

que da buenas pistas sobre posibles problemas de seguridad ya conocidos en versiones específicas, facilitando el análisis técnico detallado del sitio [96], [97].

Wget: Es una herramienta que permite descargar y guardar el contenido de un sitio web para revisión posterior mayor detalle y calma [98], [99]. Es útil cuando se desea obtener archivos disponibles o comprender cómo está organizada una página web, ya que permite descargar recursos como imágenes, documentos y código fuente, lo que facilita el análisis técnico del contenido descargado [99].

Burp Suite Community: es una de las herramientas más usadas en pruebas de seguridad de aplicaciones web que permite ver y modificar las comunicaciones entre el navegador y una página web, como si se estuviera escuchando la conversación entre ambos [100], [101], [102]. Útil para detectar y manipular datos, ver información que no debería mostrarse o aprovechar errores de seguridad [102]

Wfuzz: es una herramienta de pentesting que realiza fuzzing y ataques de fuerza bruta en aplicaciones web mediante diccionarios predefinidos, utilizando la palabra reservada FUZZ como marcador para insertar entradas en solicitudes HTTP [103]. Permite descubrir directorios ocultos, archivos, subdominios y parámetros HTTP en servidores web que no son visibles públicamente, ayuda a identificar posibles vulnerabilidades como fallos de configuraciones o inyección [104].

Nmap: Es una herramienta que sirve para escanear y mapear una dirección IP, permitiendo descubrir qué equipo y servicios están activos en una red, así como los puertos que se encuentran abiertos [105]. Ayuda a entender qué máquinas se pueden comunicar entre sí y qué servicios están visibles desde el exterior, además identificar versiones de servicios, sistemas operativos y posibles vulnerabilidades [106].

Netcraft: Es un sitio web que recopila información sobre servidores y dominio como quién los aloja y qué tipo de tecnologías utilizan [107]. Permite investigar la infraestructura detrás de un dominio antes de realizar pruebas más profundas, siendo importante para tener una primera visión general de cómo está organizado un sitio web y su nivel de exposición en internet [108].

Hackertarget: Sitio web que ofrece herramientas en línea para revisar información sobre una red o dominio, como puertos abiertos, direcciones IP, registros DNS y otros datos

relacionados con la infraestructura [109]. Permite obtener una visión rápida del estado y configuración de una red sin necesidad de instalar múltiples programas, facilitando el reconocimiento inicial del sistema [110].

ViewDNS: Es un sitio web gratuito y accesible que permite realizar consultas técnicas sobre un dominio en internet, proporcionando información detallada como registro DNS y direcciones IP asociadas al mismo. Es muy útil porque ayuda a diagnosticar problemas de configuración, verificar si un dominio está correctamente delegado o detectar posibles vectores de ataques antes de que sean explotados [111].

SonarQube: Es una herramienta de código abierto orientada al análisis estático del código fuente, que permite inspeccionar, evaluar y mejorar tanto la calidad como la seguridad de las aplicaciones en desarrollo [112]. Su principal propósito es detectar errores, malas prácticas y puntos críticos del sistema de forma temprana, antes de que la aplicación entre en producción [113].

Owasp ZAP: Es un escáner web diseñado especialmente para evaluar la seguridad de aplicaciones web [114]. Permitiendo realizar análisis tanto de forma automática y manual mediante el escaneo activo y pasivo de la aplicación, actúa como un proxy intermedio que intercepta y permite modificar el tráfico entre el navegador y el servidor en busca de errores comunes de seguridad, como agujeros por donde un atacante podrían inyectar datos maliciosos robar información [115], [116].

Wappalyzer: Es una extensión de navegador y herramienta de análisis web que permite identificar la tecnología que utiliza un sitio web, de manera no invasiva y en tiempo real [117]. Además, indica qué programa y versiones están en uso, lo que permite detectar posibles problemas, ya que muchos fallos dependen de versiones específicas de estas tecnologías [117].

Gobuster: Es una herramienta orientada a la enumeración y descubrimiento de recursos web [118]. Su funcionalidad se basa en la técnica de fuerza bruta por diccionario, mediante la cual prueba de forma sistemática y automatizada una gran cantidad de nombres de directorios, archivos y rutas sobre una página o aplicaciones web, analizando las respuestas del servidor para determinar cuáles de estos recursos existen y cuáles no, permitiendo descubrir zonas ocultas donde podría haber información sensible [119].

Nikto: Es una herramienta diseñada para realizar escaneos exhaustivos sobre servidores web, con el objetivo de identificar configuraciones inseguras, componentes desactualizados y vulnerabilidades conocidas [120]. Su funcionamiento consiste en enviar una serie de solicitudes al servidor objetivo y analizar las respuestas obtenidas, permitiendo así detectar posibles fallos de seguridad que puedan ser explotados en el sistema [121].

Dirsearch: Es una herramienta de línea de comando especialmente en la enumeración y descubrimiento de rutas, directorios y archivos ocultos dentro de aplicaciones y sitios web [122]. Su uso permite encontrar partes del sistema que podrían ser vulnerables, facilitando la identificación de recursos no expuestos de manera pública o accesible desde la interfaz principal de la aplicación [123].

2.2.5. Prueba de Análisis Estadístico

Análisis estadístico descriptivo comparativo pretest–posttest: Es un procedimiento que permite organizar, resumir y comparar cuantitativamente los datos obtenidos en dos momentos de evaluación, sin realizar inferencias estadísticas sobre una población. La estadística descriptiva permite sintetizar los datos mediante frecuencias, porcentajes y otras medidas que facilitan la identificación e interpretación de los cambios observados [124]. En la presente investigación, el pretest corresponde al estado inicial de las vulnerabilidades identificadas, mientras que el posttest corresponde al estado registrado después de la aplicación de las medidas de mitigación. La comparación se realiza mediante frecuencias, porcentajes, tasa de mitigación, tasa de persistencia y reducción absoluta y relativa, permitiendo cuantificar los cambios observados en el estado de las vulnerabilidades [125].

2.3. Marco Teórico

2.3.1. Ataques informáticos dirigidos a los sistemas de salud.

Los ataques en el entorno digital se han convertido en uno de los problemas de seguridad que más han crecido desde el año 2005 [126]. En la actualidad, mediante el mapa de ciberamenazas Kaspersky, Ecuador se posiciona en el puesto número 60 entre los países más atacados del mundo, donde se registra 12.603 eventos generales detectados, 4.670 amenazas detectadas, 459 actividades de malware, 3.595 amenazas web, 37.777

detecciones por sistemas de intrusión, 34 vulnerabilidades y 6 ransomware [127]. Debido a estos indicadores Ecuador ha llegado a ubicarse incluso entre lo más atacados de América Latina [128].

En el ámbito de la salud, los sistemas médicos son uno de los principales objetivos de los ataques informáticos porque manejan datos sensibles (historias clínicas, diagnósticos, datos personales) [129]. Entre los ataques más comunes están el ransomware: que bloquea los equipos para exigir dinero, phishing: correos falsos que engañan al personal de salud para el robo de datos o instalar malware, ataques a dispositivos médicos, fuga de información: accesos no autorizados a base de datos de pacientes, ataques DDoS: saturación de los servicios hospitalarios [130] [131] [132] [133].

En el año 2017 ocurrió un ataque ransomware WannaCry donde se aprovechó un fallo de sistemas Microsoft Windows conocida como EternalBlue, que afecta al protocolo SMB y permitía acceder a múltiples sistemas sin necesidad de una contraseña [134]. A nivel mundial, este ataque afectó a más de 150 países y más de 600 centros de salud, debido a que muchos de ellos usaban sistemas desactualizados de Microsoft Windows [135]. Como consecuencia, se produjo la paralización de servicios médicos, la inaccesibilidad a información crítica y generando riesgos directos para la vida de los pacientes, además de significativas pérdidas económicas e institucionales [136] [137].

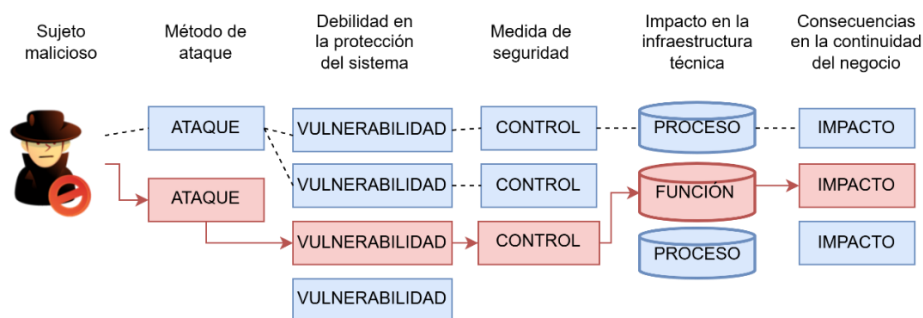


Figura 4 Riesgo de seguridad en Sistemas.

Fuente: Adaptado de [138].

El proceso de riesgos de la seguridad en un sistema se muestra en la Figura 4. Los atacantes tienen la capacidad de realizar diferentes tipos de ataques con el objetivo de identificar una debilidad en un sistema. Si logran encontrar una, se considera que el sistema presenta fallas en sus controles de protección, lo que brinda a un atacante la

oportunidad de acceder al sistema y analizar sus funciones. Por ende, si un sistema no tiene controles de seguridad adecuados, corre el riesgo de que un atacante obtenga información sobre su infraestructura y robe datos sensibles. Además, esto puede ocasionar la interrupción del servicio, lo que produciría un impacto grave en el negocio, generando pérdidas económicas, sanciones legales y daño a la reputación.

2.3.2. Principales vulnerabilidades identificadas en los sistemas de salud.

Los sistemas de salud presentan varias vulnerabilidades importantes, una de las principales es el uso de programas y equipos antiguos que no reciben actualizaciones, lo que facilita el acceso no autorizado a la información [139]. También se ha observado que muchos hospitales manejan grandes cantidades de datos personales, lo que los convierte en un objetivo atractivo para los delincuentes que buscan robar información sensible [140]. Además, la falta de capacitación del personal influye directamente, ya que los errores humanos pueden provocar filtraciones o accesos indebidos [141]. Otros aspectos relevantes es la complejidad de los sistemas, ya que incluyen múltiples aplicaciones y dispositivos conectados, lo que genera más puntos vulnerables [142].

Por otra parte, diferentes investigaciones señalan que la rápida digitalización en el sector ha incrementado los riesgos, debido a que no siempre se implementan medidas de protección adecuadas [143]. Asimismo, la presencia de muchos dispositivos conectados, como equipos médicos, aumenta las posibilidades de fallas en la seguridad [144]. También se ha identificado que la falta de mantenimiento y control en los sistemas permite que existan múltiples debilidades que pueden ser aprovechadas [145]. El crecimiento de ataques como el robo de datos o el bloqueo de sistemas demuestra que las instituciones de salud aún presentan deficiencias en la protección de la información [146], [147].

2.3.3. Vulnerabilidades más comunes detectadas en aplicaciones web y móvil.

Las aplicaciones web y móviles es un componente principal en el diario vivir, ya que muchas personas las utilizan para sus actividades diarias [148]. Las aplicaciones web que usan múltiples tecnologías como frontend, backend, base de datos, APIs, sistemas integrados son las principales de surgir más vulnerabilidades de seguridad [149]. A medida que se realice un cambio en la arquitectura del sistema, existe la posibilidad de

que se visualicen nuevas vulnerabilidades que puedan comprometer el sistema si no son analizadas con tiempo [150].

Las aplicaciones móviles son muy importantes en la actualidad debido a su múltiple función como comunicarnos con amigos, realizar compras en línea, transferencias entre otras actividades [151]. Estudios científicos y técnicos señalan que muchas de estas aplicaciones presentan vulnerabilidades que pueden ser explotadas por atacantes para robar datos personales, acceder a cuentas financieras o incluso afectar el correcto funcionamiento del dispositivos [152].

La organización OWASP la cual se dedica a estudiar la seguridad de las aplicaciones, ha elaborado una lista donde describe las vulnerabilidades más comunes en aplicaciones, tales como [153], [154]:

Top 10 de OWASP Aplicación web		OWASP Mobile Top 10 Aplicación móvil	
A1	Perdidas de control de acceso	M1	Uso indebido de credenciales
A2	Fallos de criptografías	M2	Seguridad inadecuada de la cadena de suministro
A3	Inyecciones	M3	Autenticación/autorización insegura
A4	Diseño inseguro	M4	Validación insuficiente de entrada/salida
A5	Configuración de seguridad incorrecta	M5	Comunicación insegura
A6	Componentes vulnerables y desactualizados	M6	Controles de privacidad inadecuados
A7	Fallos de identificación y autenticación	M7	Protecciones binarias insuficientes
A8	Fallos en el software y en la integridad de los datos	M8	Configuración de seguridad incorrecta
A9	Fallos en el registro de monitoreo	M9	Almacenamiento de datos inseguro
A10	Falsificación de solicitud del lado del servidor (SSRF)	M10	Criptografía insuficiente

Tabla 8 Lista de vulnerabilidades más comunes en aplicaciones web y móvil según OWASP

2.3.4. Aplicación de OWASP en los sistemas informáticos

OWASP es una guía internacional que ayuda a proteger los sistemas informáticos contra los ataques más comunes, como el robo de información, los accesos no autorizados o los errores de configuración[155]. Seguir las recomendaciones de OWASP durante el desarrollo y mantenimiento de un sistema permite identificar y corregir problemas de seguridad antes de que los aprovechen los delincuentes informáticos[155]. Esto es especialmente importante en sistemas que manejan información sensible, como los del sector salud, donde una falla de seguridad podría afectar la privacidad de los pacientes o interrumpir la atención médica[156]. Usar OWASP trae varios beneficios, como reducir riesgos, cumplir con norma de protección de datos, mejorar la calidad del software desde su fase inicial y sobre todo genera confianza a los usuarios[157].

Requerimientos

2.3.5. Requerimientos Funcionales

El objetivo de los requerimientos funcionales es describir de manera clara y ordenada las actividades principales que se desarrollarán durante la evaluación del sistema ViTECED, con el fin de identificar sus posibles debilidades y establecer una base adecuada para el análisis de seguridad.

Actividad de recopilación de información

Código	Descripción
RF-01	Obtener una caracterización de la arquitectura del sistema ViTECED que permita definir el alcance de la evaluación y reconocer la relación entre sus principales componentes.
RF-02	Recopilar información sobre las tecnologías, herramientas y recursos utilizados en el desarrollo e implementación del sistema para orientar el proceso de evaluación.
RF-03	Establecer una línea base del estado de las prácticas de seguridad del sistema mediante la aplicación del modelo OWASP SAMM.
RF-04	Consolidar la documentación técnica disponible del sistema para facilitar la planificación y ejecución de las actividades de evaluación.

Tabla 9 Requerimientos funcionales – Actividad de recopilación de información

Actividad de identificación de vulnerabilidades

Código	Descripción
RF-05	Obtener evidencias sobre el estado de seguridad de la aplicación web y de la aplicación móvil mediante la ejecución de las evaluaciones definidas en la metodología.
RF-06	Analizar el comportamiento del sistema y la información disponible de sus componentes para identificar condiciones que puedan favorecer la presencia de vulnerabilidades.
RF-07	Registrar de forma organizada los resultados obtenidos durante la evaluación, garantizando la trazabilidad de los hallazgos identificados.
RF-08	Valorar el nivel de riesgo de las vulnerabilidades identificadas considerando el impacto que representan para la información y los servicios del sistema.
RF-09	Asociar las vulnerabilidades identificadas con las categorías de referencia propuestas por OWASP para facilitar su análisis e interpretación.

Tabla 10 Requerimientos funcionales - Actividad de identificación de vulnerabilidades

Actividad de remediación

Código	Descripción
RF-10	Elaborar un reporte técnico que reúna los hallazgos obtenidos durante la evaluación y sirva como base para la aplicación de acciones correctivas.
RF-11	Coordinar la aplicación de medidas orientadas a reducir o eliminar las vulnerabilidades identificadas durante la evaluación del sistema.
RF-12	Mantener un seguimiento del estado de cada vulnerabilidad hasta concluir el proceso de remediación.

Tabla 11 Requerimientos funcionales - Actividad de remediación

Actividad de reevaluación

Código	Descripción
RF-13	Obtener una nueva valoración del estado de seguridad del sistema después de aplicar las acciones de remediación.
RF-14	Confirmar la efectividad de las medidas implementadas mediante la verificación del estado final de las vulnerabilidades previamente identificadas.
RF-15	Determinar la existencia de nuevas vulnerabilidades que puedan haberse originado como consecuencia de las modificaciones realizadas al sistema.
RF-16	Comparar el estado de las vulnerabilidades antes y después del proceso de remediación mediante un análisis estadístico descriptivo comparativo pretest–posttest, utilizando frecuencias, porcentajes, tasa de mitigación, tasa de corrección parcial, tasa de persistencia, reducción absoluta y reducción relativa para valorar los cambios observados.

Tabla 12 Requerimiento funcionales - Actividad de reevaluación

Actividad de elaboración del informe

Código	Descripción
RF-17	Elaborar un informe final con los resultados obtenidos durante la evaluación del sistema ViTECED.
RF-18	Incluir recomendaciones para mejorar la seguridad en las partes del sistema donde se detectaron problemas.

Tabla 13 Requerimientos funcionales - Actividad de elaboración del informe

2.3.6. Requerimientos no funcionales

Las necesidades no funcionales son importantes para garantizar que el proceso de evaluación se realice de forma ordenada, segura y útil sin afectar el funcionamiento normal del sistema ViTECED. A continuación, se describen estas necesidades.

Seguridad y cuidado de la información

Código	Descripción
RNF-01	El proceso de evaluación no debe afectar de forma importante el funcionamiento normal del sistema ViTECED.

Código	Descripción
RNF-02	La información obtenida durante la evaluación debe ser tratada con reserva y utilizada únicamente con fines académicos y de análisis.
RNF-03	La evaluación debe realizarse sin poner en riesgo los datos sensibles manejados por el sistema.

Tabla 14 Requerimientos no funcionales - Seguridad y cuidado de la información

Claridad y presentación

Código	Descripción
RNF-05	El informe final debe redactarse con un lenguaje claro, sencillo y formal.
RNF-06	La información presentada debe estar organizada de forma que sea fácil de leer y entender.
RNF-07	El documento debe incluir tablas o listas cuando sea necesario para facilitar la comprensión de los resultados.

Tabla 15 Requerimientos no funcionales - Claridad y presentación

2.4. Componentes de la propuesta

El desarrollo del proyecto inicia con la identificación de la arquitectura y los componentes del sistema informático ViTECED, lo que permite reconocer los activos críticos y delimitar la superficie de ataque. A partir de este análisis se obtiene una visión general del entorno evaluado, la cual sirve como base para la aplicación de la metodología de seguridad y la posterior identificación de vulnerabilidades.

2.5. Arquitectura del sistema

Como se observa en la Figura 5, el flujo de datos del sistema ViTECED inicia desde el dispositivo IoT, el cual se encarga de capturar los parámetros biomédicos de los pacientes y los publica al Server Broker MQTT mediante el protocolo MQTT sobre TLS garantizando que la información enviada llegue de forma cifrada y segura. El bróker actúa como intermediario central que distribuye la información recibida a los servicios internos suscritos a sus tópicos en tiempo real. El servicio MQTT se ejecuta dentro de un contenedor Docker y funciona mediante un script en Python.

Este servicio se encuentra suscrito a los tópicos del bróker, por lo que recibe cada dato publicado por los dispositivos IoT. Una vez que obtiene la información, el servicio realiza

De esta manera, la arquitectura permite un flujo de información continuo y seguro, desde la captura de datos en el dispositivo IoT hasta su almacenamiento estructurado o no estructurado, así como su posterior visualización y consulta por parte de los usuarios a través del cliente web y móvil.

2.5.1. Componentes del sistema ViTECED

En la Tabla 16, se muestra los componentes que forman parte de la arquitectura del sistema.

Componentes	Descripción
Front-end	Aplicación móvil
	Aplicación web
Back-end	API General – RESTful PHP
	API compatible con HL7 Fhir
	API para consultas MongoDB
Servicios	Servicio de generación de alertas con lógica difusa. (MQTT)
	Servicio análisis morfológico del ECG
Dispositivos IoT	Monitor de signos vitales tales como: frecuencia cardíaca, saturación de oxígeno, temperatura corporal y ECG.
Base de datos	MongoDB
	MaríaDB
Servidor	Servidor MQTT Mosquito

Tabla 16 Componente del sistema ViTECED

2.5.2. Puntos críticos considerados para el análisis del sistema ViTECED

Como se observa en la Tabla 17, los componentes críticos del sistema ViTECED son aquellos que gestionan información sensible, controlan el acceso de los usuarios y mantienen la comunicación entre los diferentes elementos de la plataforma. Estos componentes requieren especial atención, ya que concentran el ingreso, procesamiento, almacenamiento y visualización de la información biomédica.

Componente / Activo	Activo crítico identificado	Superficie de ataque
Aplicación móvil Android	Credenciales de usuarios, datos biomédicos en tiempo real, sesiones activas	Robo de credenciales, malware en dispositivos móviles, interceptación de tráfico MQTT/WebSocket, acceso no autorizado por sesiones inseguras.
Aplicación web	Información médica de pacientes, historiales clínicos, reportes y recetas.	Ataques XSS, SQL Injection, secuestro de sesiones, explotación de formularios web, acceso indebido por errores de permisos.
API RESTful PHP	Servicios de autenticación, consultas médicas, operaciones CRUD del sistema.	Manipulación de endpoints, ataques por inyección, abuso de métodos HTTP, robo de tokens de autenticación.
API HL7 FHIR	Histórias clínicas electrónicas e interoperabilidad médica.	Exposición de datos clínicos sensibles, acceso indebido a recursos FHIR, manipulación de datos JSON clínicos.
API MongoDB	Registros históricos biomédicos y señales ECG.	Consultas no autorizadas, exposición de datos no estructurados, modificación o eliminación de registros clínicos.
Servicio MQTT	Flujo de datos biomédicos provenientes de dispositivos IoT.	Inyección de mensajes falsos, publicación no autorizada, manipulación de tópicos MQTT.
Servicio ECG	Procesamiento y análisis de señales electrocardiográficas.	Alteración de señales biomédicas, falsos positivos/negativos en alertas, explotación de procesos automatizados.

Componente / Activo	Activo crítico identificado	Superficie de ataque
Base de datos MySQL/MariaDB	Información clínica, usuarios, médicos, pacientes y configuraciones del sistema.	Robo de bases de datos, escalamiento de privilegios, inyección SQL, pérdida de información crítica.

Tabla 17 Punto críticos considerados para el análisis del sistema ViTECED

2.5.3. Análisis del nivel de madurez del sistema ViTECED mediante SAMM

Con la aplicación del modelo SAMM en el sistema ViTECED, se determinó que el nivel de madurez de la seguridad del sistema en las distintas funciones comerciales es heterogéneo, evidenciando áreas con mayor grado de desarrollo y otras que requieren mejoras significativas. Como se muestra en la Tabla 18, el promedio general de madurez obtenido fue de 0.91, lo que indica que el sistema se encuentra en un nivel de madurez, caracterizado por la existencia de procesos básicos de seguridad. Esto significa que, aunque se reconoce la importancia de la seguridad, aun no existen procedimientos formales ni se aplican de manera constante.

Funciones comerciales	Puntaje
Gobernanza	0,29
Diseño	1,79
Implementación	1,50
Verificación	0,50
Operación	0,46
General	0,91

Tabla 18 Resultados general de la evaluación del modelo SAMM en el sistema ViTECED

A continuación, se presenta el análisis de los resultados obtenidos mediante la evaluación del modelo SAMM (Software Assurance Maturity Model) aplicado al sistema ViTECED. Estas evaluaciones permitieron determinar el nivel de madurez en las diferentes funciones comerciales y prácticas de seguridad, organizadas en los dominios de Gobernanzas, Diseño, Implementación, Verificación y Operación.

En la Tabla 19 y la Figura 6, se detallan los puntajes obtenidos por cada practica de seguridad, así como su nivel de madurez asociado.

Funciones comerciales	Prácticas de seguridad	Puntaje	Madurez		
			1	2	3
Gobernanza	Estrategia y métricas	0,00	0,00	0,00	0,00
Gobernanza	Política y cumplimiento	0,50	0,25	0,00	0,25
Gobernanza	Educación y Guía	0,38	0,00	0,13	0,25
Diseño	Evaluación de amenazas	1,38	0,63	0,25	0,50
Diseño	Requisitos de seguridad	1,00	0,38	0,38	0,25
Diseño	Arquitectura segura	3,00	1,00	1,00	1,00
Implementación	Construcción segura	1,75	1,00	0,50	0,25
Implementación	Implementación segura	1,88	0,75	0,75	0,38
Implementación	Gestión de defectos	0,88	0,38	0,25	0,25
Verificación	Evaluación de Arquitectura	1,13	0,50	0,50	0,13
Verificación	Pruebas de requisitos	0,38	0,25	0,00	0,13
Verificación	Pruebas de seguridad	0,00	0,00	0,00	0,00
Operación	Gestión de Incidentes	0,00	0,00	0,00	0,00
Operación	Gestión Ambiental	0,25	0,13	0,13	0,00
Operación	Gestión Operativa	1,13	0,63	0,50	0,00

Tabla 19 Resultados de evaluación de prácticas de seguridad y niveles de madurez en el sistema ViTECED mediante SAMM

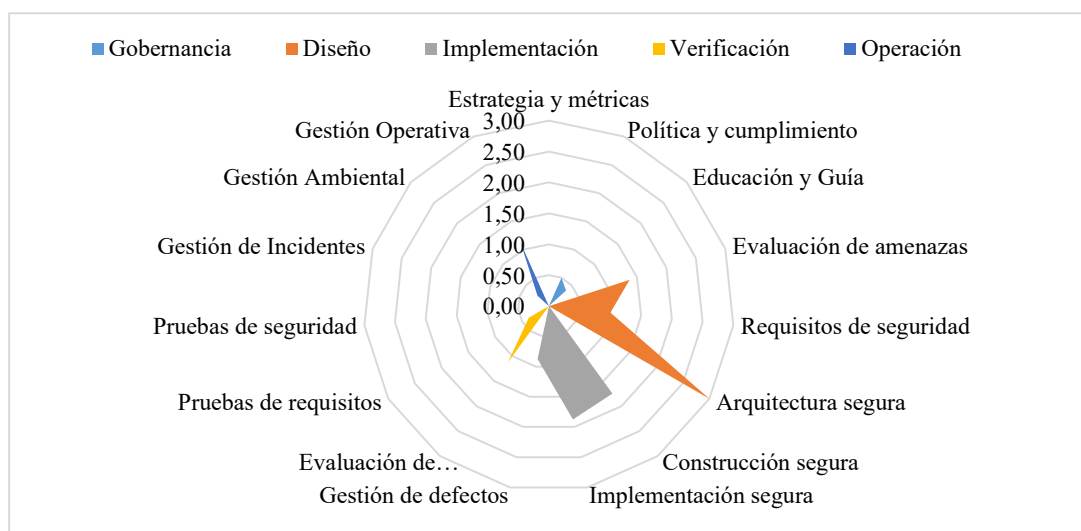


Figura 6 Análisis gráfico de las funciones comerciales y prácticas de seguridad evaluadas en el sistema ViTECED mediante el modelo SAMM.

Fuente: Elaboración propia

Gobernanza

En el dominio de Gobernanza se obtuvo una puntuación de madurez promedio de 0,29, lo que muestra que el sistema se encuentra en una etapa inicial en la gestión estratégica y organizacional de la seguridad de sus aplicaciones.

Este dominio se evaluó mediante tres prácticas de seguridad:

- **Estrategia y métrica**

En esta práctica se obtuvo un nivel de madurez de 0.00, lo que muestran que el sistema se encuentra en un nivel bajo de madurez en seguridad. Ninguna de las actividades evaluadas se cumple, lo que evidencia que la organización aún no ha desarrollado una visión clara respecto al manejo del riesgo en su aplicación. No existe un plan estratégico de seguridad, tampoco se cuenta con procesos formales para medir la eficacia de los controles implementados. La ausencia de métricas y KPI impide evaluar si las acciones actuales realmente contribuyen a mejorar la seguridad del sistema.

- **Política de cumplimiento**

En esta categoría, se obtuvo un promedio de 0.50, lo que demuestra que la evaluación refleja que la mayoría de los controles no están implementados, especialmente aquellos relacionados con políticas internas de seguridad. La organización no cuenta con políticas estandarizadas ni con documentación clara para que los equipos de desarrollo las sigan. Tampoco existe una revisión periódica del cumplimiento de las políticas internas.

Sin embargo, se observan dos avances importantes:

- La organización tiene algo de visibilidad sobre las obligaciones de cumplimiento externo, aunque solo para una parte de sus aplicaciones.
- Se realizan informes periódicos relacionados con el cumplimiento externo, lo que indica un esfuerzo parcial por atender requisitos regulatorios.

A pesar de estos puntos positivos, la falta de políticas internas claras y estandarizadas deja al sistema vulnerable, pues los equipos no tienen lineamientos definidos para trabajar de manera segura.

- **Educación y Guía**

En esta práctica, los resultados muestran un nivel de madurez bajo de 0.38, pero con algunos elementos iniciales. La mayoría de las actividades no se realizan, como la capacitación obligatoria en SDLC seguro o la designación de responsables de seguridad en cada equipo. Tampoco existe un sistema formal para gestionar la formación del personal.

No obstante, se identifican algunos avances:

- Parte de la capacitación que sí se realiza está personalizada según roles, lo cual es positivo.
- Existe un portal centralizado que permite compartir información entre algunas áreas de la organización.

Estos elementos muestran que existe una intención de mejorar la cultura de seguridad, pero todavía no es suficiente para sostener un programa maduro. Para avanzar, es necesario formalizar los procesos de capacitación, establecer responsables y promover espacios de aprendizaje continuo.

Diseño

En este dominio de diseño, se obtuvo un nivel de madurez promedio de 1.79, lo que indica un desempeño intermedio en la integración de prácticas de seguridad durante la fase de diseño del software.

Dentro de este dominio, se evaluaron tres practicas claves:

- **Evaluación de Amenazas**

La evaluación realizada demuestra que VITECED presenta un nivel de madurez moderado de 1.38 en la identificación y gestión de amenazas. La mayoría de las aplicaciones se clasifican según su nivel de riesgo utilizando un conjunto predefinido de preguntas, lo cual es un punto fuerte y refleja un enfoque organizado para entender la exposición al riesgo. Además, la organización actualiza los perfiles de riesgo cuando existe un cambio relevante en la aplicación, lo que muestra una preocupación por mantener la información actualizada.

Sin embargo, algunas actividades esenciales solo se aplican a una parte de las aplicaciones. Por ejemplo, el uso de perfiles de riesgo cuantificados y el modelado de

amenazas no se realizan de manera completa en toda la organización. Aunque sí se utiliza una metodología estándar para el modelado de amenazas, su aplicación es parcial. A pesar de estas limitaciones, el hecho de que la metodología sea revisada periódicamente muestra que existe una intención clara de mejorar.

- **Requisitos de Seguridad**

En la práctica de requisitos de seguridad, VITECED muestra un nivel de madurez intermedio de 1.00, con varios avances importantes. Los equipos definen requisitos de seguridad durante el desarrollo para al menos la mitad de sus aplicaciones, lo cual indica que la seguridad sí está siendo considerada desde las primeras fases del ciclo de vida del software. También existe un esfuerzo por priorizar estos requisitos y por organizarlos adecuadamente.

Además, la organización utiliza un marco estándar para obtener requisitos de seguridad, aunque solo se aplica en algunos proyectos. Esto contribuye a mantener un enfoque más estructurado, pero aún no se ha estandarizado del todo. En cuanto al trabajo con proveedores la evaluación muestra que el cumplimiento de responsabilidades de seguridad es irregular. Las revisiones y alineaciones con controles estándar ocurren solo a veces, lo que revela una falta de consistencia en la supervisión externa.

- **Arquitectura Segura**

Esta es la práctica se demuestra un mayor nivel de madurez de 3.00. Los resultados indican que los equipos aplican principios de seguridad en el diseño para la mayoría o todas las aplicaciones. También utilizan servicios de seguridad compartidos y basan su diseño en arquitecturas de referencia, lo que fortalece la consistencia y la protección del sistema.

En el entorno tecnológico, VITECED evalúa la calidad de seguridad de las herramientas utilizadas y mantiene una lista de tecnologías recomendadas. Además, impulsa su adopción dentro de la organización, lo cual fortalece la estandarización y reduce riesgos asociados al uso de tecnologías no evaluadas. Este conjunto de prácticas demuestra que la organización tiene un enfoque muy sólido en la arquitectura segura y que utiliza mecanismos confiables para diseñar y estructurar sus sistemas. Comparado con las otras áreas evaluadas, aquí se observa un nivel de madurez mucho más alto.

Implementación

Mediante el análisis del dominio de implementación en ViTECED, se evidencia un nivel de madurez promedio de 1.50, lo que indica que el sistema se encuentra en un nivel de madurez parcial. Esto significa que, aunque se observan algunos avances, aún existen debilidades importantes que afectan la consistencia y seguridad de las aplicaciones implementadas.

Este dominio se evaluó mediante tres prácticas de seguridad:

- **Compilación Segura**

Con base en los resultados obtenidos, en esta práctica registró un promedio de nivel de madurez de 1.75 en los procesos relacionados con la compilación de software. El aspecto más destacado es que el proceso de compilación está descrito formalmente y se aplica en la mayoría de las aplicaciones, lo que demuestra que existe una estructura clara y documentada para construir el software. Además, el proceso está en gran parte automatizado, lo que reduce errores humanos y aumenta la confiabilidad del sistema.

Sin embargo, las comprobaciones de seguridad automatizadas solo se aplican en algunas aplicaciones, lo que evidencia una brecha importante: no existe una cobertura completa de análisis o validaciones automáticas de seguridad en todos los proyectos. En cuanto al manejo de dependencias, la organización conoce bien las tecnologías que utiliza, pero no cuenta con un proceso formal para gestionar los riesgos asociados a dependencias de terceros. Este es un punto débil, ya que muchas vulnerabilidades actuales provienen justamente de librerías externas. Aun así, el sistema es capaz de impedir la compilación cuando detecta algunas vulnerabilidades, lo que representa una medida de control positiva.

- **Implementación Segura**

En esta práctica se alcanzó un nivel de madurez de 1.88, reflejando prácticas sólidas relacionadas con la automatización y protección durante la etapa de despliegue. Los resultados muestran que ViTECED utiliza procesos de implementación repetibles y parcialmente automatizados, lo que ayuda a garantizar consistencia entre versiones. Además, en la mitad de sus aplicaciones, estos procesos incluyen comprobaciones de seguridad, lo cual eleva la confiabilidad del despliegue.

La validación de la integridad de los artefactos es más limitada, ya que solo se realiza en algunas aplicaciones. Este punto debería fortalecerse, pues la integridad es fundamental para evitar manipulaciones o versiones corruptas del software. Una de las fortalezas más importantes de VITECED es su gestión de secretos. La organización aplica el principio de mínimo privilegio, inyecta los secretos durante la implementación y mantiene un ciclo de vida controlado para ellos. Esto demuestra un enfoque maduro en la protección de credenciales, tokens y llaves, que son elementos críticos para la seguridad del sistema.

- **Gestión de Defectos**

En esta práctica, se alcanzó un nivel de madurez de 0.88, lo que refleja una gestión parcial de los defectos de seguridad dentro de la organización. Aunque existe un seguimiento básico de vulnerabilidades, este únicamente se aplica en cierta parte de la aplicación, lo que limita una visión completa del estado de seguridad del sistema. Además, la organización no cuenta con una cobertura completa de niveles de servicios (SLA) para la corrección de defectos ni de un control centralizado que permita gestionar adecuadamente el ciclo de vida de las vulnerabilidades identificadas.

A pesar de estas limitaciones, se evidencian avances importantes en el uso de métrica básicas para apoyar la toma de acciones rápidas frente a problemas de seguridad. Así mismo, se aplican algunas métricas estandarizadas orientadas a mejorar el programa de seguridad, lo que demuestra un interés por adoptar un enfoque más analítico en la gestión de vulnerabilidades. Sin embargo, estas prácticas aún no se implementan de manera uniforme en todas las áreas y aplicaciones del sistema. De igual forma, aunque la eficacia de las métricas es evaluada periódicamente, esta actividad solo se realiza en ciertos casos, evidenciando que la organización todavía necesita fortalecer la consistencia y cobertura de sus procesos de gestión de defectos.

Verificación

El análisis del dominio de verificación en ViTECED muestra un nivel de madurez promedio de 0.50 lo que indica que el sistema se encuentra en un nivel de madurez parcial bajo. Esto significa que, aunque existen algunos avances en ciertas áreas, todavía hay importantes debilidades que limitan la eficacia de la seguridad durante la fase de verificación de las aplicaciones.

Este dominio se evaluó mediante tres prácticas de seguridad:

- **Evaluación de la Arquitectura**

La revisión realizada en esta prueba permitió determinar un nivel de madurez de 1.13, en relación con la evaluación de la arquitectura del sistema. Los resultados evidencian que la organización realiza revisiones a la arquitectura del sistema para identificar objetivos de seguridad en al menos la mitad de su aplicación. Además de efectuar revisiones periódicas sobre algunos de los mecanismos de seguridad existentes.

Esto demuestra que existe un esfuerzo por analizar la estructura del sistema y fortalecer ciertos componentes relacionados con la seguridad, aunque estas actividades aún no se aplican de manera integral en toda la organización. No obstante, también se identificaron debilidades importantes en el proceso de evaluación.

La revisión de la eficacia de los controles de seguridad solo se realiza en cierta parte de la aplicación. Esto limita la capacidad del sistema para detectar fallas o controles que ya no funcionan adecuadamente. Además, aunque se revisa parcialmente la arquitectura para mitigar amenazas, la organización no evalúa periódicamente estas amenazas ni actualiza sus arquitecturas de referencia. Esto significa que los cambios en el entorno, nuevas vulnerabilidades o buenas prácticas recientes no se reflejan automáticamente en la arquitectura del sistema.

- **Pruebas de Requisitos**

En esta práctica, el nivel de madurez es de 0.38, ya que solo se cumplen parcialmente los requisitos mínimos que se esperarían en un ciclo de pruebas más completo. El sistema realiza pruebas para verificar que los controles de seguridad estándar funcionen, pero esto solo se aplica a la mitad de las aplicaciones. Esto es positivo, pero insuficiente para garantizar seguridad integral. Por otro lado, no se realizan pruebas sistemáticas mediante scripts ni pruebas automatizadas para detectar regresiones de seguridad. Esto representa una debilidad clara porque impide identificar fallos introducidos en nuevas versiones del software.

Tampoco se utilizan técnicas avanzadas como fuzzing, creación de casos de abuso o pruebas exhaustivas de denegación de servicio. Estas técnicas son esenciales para identificar comportamientos inesperados o vulnerabilidades que no se detectan con

pruebas tradicionales. Aunque ocasionalmente se realizan pruebas de estrés o de denegación de servicio, estas no se aplican de manera completa ni constante.

- **Pruebas de Seguridad**

En esta práctica se muestra el nivel de madurez bajo de 0.00, por lo que se deduce que el sistema no utiliza herramientas automatizadas de pruebas de seguridad, no adapta herramientas a sus tecnologías y no integra pruebas de seguridad en los procesos de compilación o implementación. Esto es significativo porque la automatización es una de las formas más efectivas para detectar vulnerabilidades antes de que las aplicaciones lleguen a producción.

Tampoco se realizan revisiones manuales de componentes críticos, ni se ejecutan pruebas de penetración de manera periódica. Estas dos actividades son fundamentales para descubrir fallas que solo pueden detectarse con análisis manuales o ataques simulados. Finalmente, la organización no utiliza los resultados de pruebas de seguridad para mejorar su ciclo de vida de desarrollo. Esto significa que no existe un proceso de retroalimentación, lo cual impide corregir errores de forma estructural. Este conjunto de resultados evidencia que esta área es una prioridad inmediata para mejorar, ya que un sistema sin pruebas de seguridad formales está expuesto a fallas y vulnerabilidades no detectadas.

Operación

El análisis del dominio de Operación, se obtuvo un nivel de madurez bajo de 0.46. Esto significa que, aunque existen algunas acciones aisladas para proteger y gestionar los recursos, las prácticas de seguridad operativa no se aplican de manera sistemática, lo que aumenta los riesgos sobre la continuidad y protección de las aplicaciones.

Este dominio se evaluó mediante tres prácticas de seguridad:

- **Gestión de incidentes**

Los resultados obtenidos en esta práctica reflejan un promedio de madurez de 0.00, lo que evidencia que actualmente el sistema no cuenta con procesos establecidos para el análisis de registros, la detección documentada de incidentes, ni revisa periódicamente la eficacia de sus procedimientos de seguridad. Tampoco existe un equipo dedicado a atender incidentes, ni se aplican procesos repetibles para la respuesta. Esto indica que la

organización no está preparada para detectar ni reaccionar ante incidentes de seguridad, lo que representa un riesgo importante para la protección de la información y la continuidad del sistema. La implementación de procesos formales y un equipo responsable serían un paso crítico para mejorar esta área.

- **Gestión del medio ambiente**

Durante la evaluación de esta práctica se determinó un nivel de madurez de 0.25. Algunos componentes tecnológicos están reforzados, pero no existen bases de refuerzo formales, ni supervisión sistemática del cumplimiento de estas configuraciones. Además, la identificación y parcheo de componentes vulnerables se aplican de manera parcial, y no se evalúa regularmente el estado de los parches ni se mantiene un proceso completo de actualización de los sistemas. Aunque existen esfuerzos aislados por mejorar la seguridad del entorno, no hay un enfoque organizado ni continuo. Esto deja al sistema expuesto a vulnerabilidades conocidas y sistemas no actualizados.

- **Gestión operativa**

La gestión operativa muestra un nivel de madurez medio de 1.13, lo que indica que la organización protege la información de la mayoría de sus aplicaciones y realiza la eliminación de recursos y sistemas obsoletos de manera adecuada. Sin embargo, no mantiene un catálogo formal de datos, ni revisa periódicamente las políticas de protección de la información. Tampoco se evalúa de manera constante el ciclo de vida de los activos tecnológicos. Esto evidencia que, aunque existen buenas prácticas en la protección de datos y eliminación de recursos, faltan procesos estructurados y consistentes para la gestión completa de los activos y la información

2.6. Metodología basada en OWASP

En esta sección se describen las pruebas de seguridad realizadas sobre la aplicación web, siguiendo la Guía de Pruebas OWASP v4.0, y sobre la aplicación móvil del sistema ViTECED, basadas en la Mobile Security Verificación Standard (MSVS) y el Mobile Aplicación Security Testing Guide (MASGT).

2.6.1. Pruebas en la aplicación web

Recopilación de Información

En esta etapa inicial se realiza la obtención de información relacionada con las aplicaciones web, debido a que conocer sus características y funcionamiento constituye a un aspecto fundamental para el desarrollo de las evaluaciones de seguridad. Asimismo, con el propósito de organizar e identificar de manera adecuada cada una de las pruebas establecidas en la metodología, se emplea una nomenclatura específica para su clasificación.

Identificador de prueba:	Conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información
Objetivo:	Identificar información sensible relativa al diseño y configuración de la plataforma que se encuentre expuesta públicamente a través de motores de búsqueda, ya sea de forma directa en el sitio web de la organización o de forma indirecta mediante sitios de terceros, cachés indexados u otras fuentes de acceso público.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org)
Herramientas Utilizadas:	Bing, Google Hacking, Shodan.io

Tabla 20 Prueba conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información

Identificador de prueba:	Huellas digitales en el servidor web
Objetivo:	Identificar Encontrar la versión y el tipo de servidor web en ejecución para determinar vulnerabilidades conocidas y la manera de explotarlas para usar durante la prueba.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend de API (https://apiviteced1.teced.org).
Herramientas Utilizadas:	OWASP ZAP v2.17.0, curl, Whatweb

Tabla 21 Prueba huellas digitales en el servidor web

Identificador de prueba:	Revisión de los metas archivos del servidor web en busca de fugas de información
Objetivo:	Identificar fugas de información relativas a rutas o directorios internos de la aplicación web expuestas a través de metaarchivos del servidor. Construir el listado de directorios y recursos que deben ser excluidos del rastreo por parte de arañas, robots o crawlers web.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org)
Herramientas Utilizadas:	Curl

Tabla 22 Prueba revisión de los metas archivos del servidor web en busca de fugas de información

Identificador de prueba:	Enumere las aplicaciones en el servidor web
Objetivo:	Identificar las aplicaciones que se ejecutan en el servidor web, las cuales en muchos casos pueden presentar vulnerabilidades y patrones de ataques conocidos que podrían ser explotados para acceder a información sensibles de la aplicación.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend de API (https://apiviteced1.teced.org).
Herramientas Utilizadas:	wfuzz, nmap, netcraft, hackertarget, ViewDNS.info, nslookup

Tabla 23 Prueba enumere las aplicaciones en el servidor web

Identificador de prueba:	Identificar puntos de entrada de la aplicación web
Objetivo:	Revisar las solicitudes y respuestas que indican los métodos GET y POS, para identificar todos los puntos de entrada disponibles en la

Identificador de prueba:	Identificar puntos de entrada de la aplicación web
	aplicación web, lugares donde un usuario o atacante puede interactuar con el sistema e ingresar datos.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend de API (https://apiviteced1.teced.org).
Herramientas Utilizadas:	OWASP ZAP v2.17.0

Tabla 24 Prueba identificar puntos de entrada de la aplicación web

Categoría de prueba: Prueba de administración de identidad

Identificador de prueba:	Prueba de definición de Roles
Objetivo:	Validar los diferentes roles definidos dentro de la aplicación y verificar que exista una separación adecuada entre las funciones del negocio y las funciones del sistema, asegurando que cada rol acceda únicamente a la información y funcionalidad que le corresponde según las políticas de acceso de la organización.
Entornos Evaluados:	Frontend (https://viteced.teced.org)
Herramientas Utilizadas:	Documentación manual del sistema, Navegador Web

Tabla 25 Prueba de definición de Roles

Identificador de prueba:	Prueba de enumeración y adivinanzas de cuentas de usuario
Objetivo:	Verificar si la aplicación web revela la existencia de nombres de usuarios registrados en el sistema, a través de diferencias observables en las respuestas del mecanismo de autenticación.
Modalidad:	Caja negra, Caja gris

Identificador de prueba:	Prueba de enumeración y adivinanzas de cuentas de usuario
Entornos Evaluados:	Frontend (https://viteced.teced.org), backend (https://apiviteced.teced.org)
Herramientas Utilizadas:	Burp Suite v2026.7.2

Tabla 26 Prueba de enumeración y adivinanzas de cuentas de usuario

Categoría de prueba: Pruebas de Autenticación

Identificador de prueba:	Prueba de transporte de credenciales en canal encriptado
Objetivo:	Verificar que las credenciales de autenticación y demás datos sensibles ingresados por el usuario sean transmitidos a través de canales cifrados mediante el protocolo HTTPS/TLS, garantizando que la información no viaje en texto plano y no pueda ser interceptada por un atacante mediante técnicas de captura de tráfico de red.
Modalidad:	Caja negra, Caja gris
Entornos Evaluados:	Backend API REST (https://apiviteced.teced.org)
Herramientas Utilizadas:	Burp Suite v2026.7.2, Entrevista técnica

Tabla 27 Prueba de transporte de credenciales en canal encriptado

Identificador de prueba:	Pruebas para buscar la debilidad de memoria caché del navegador
Objetivo:	Verificar que una aplicación web esté configurada correctamente para evitar que el navegador almacene información sensible del usuario en su caché o en su historial de navegación.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org)

Identificador de prueba:	Pruebas para buscar la debilidad de memoria caché del navegador
Herramientas Utilizadas:	Desarrollador del navegador

Tabla 28 Pruebas para buscar la debilidad de memoria caché del navegador

Identificador de prueba:	Prueba de credenciales por defecto
Objetivo:	Verificar si la aplicación permite acceso no autorizado mediante el uso de credenciales predeterminadas, predecibles o dejadas por defecto en cualquiera de sus puntos de autenticación.
Modalidad:	Caja negra, Caja gris
Entornos Evaluados:	Formulario de Inicio de Sesión (POST /login), Módulo de Registro de Usuarios
Herramientas Utilizadas:	Burp Suite v2026.7.2 y DevTools del navegador.

Tabla 29 Prueba de credenciales por defecto

Categoría de prueba: Pruebas de Autorización

Identificador de prueba:	Pruebas para determinar el escalamiento de privilegios
Objetivo:	Verificar que los mecanismos de control de acceso implementados en la aplicación impidan que un usuario modifique sus privilegios o acceda a funcionalidades y recursos que no le corresponden, garantizando que las validaciones de autorización sean realizadas correctamente por el servidor.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend API REST (https://apiviteced.teced.org)
Herramientas Utilizadas:	Burp Suite V2026.7.2, Logto.io

Tabla 30 Pruebas para determinar el escalamiento de privilegios

Categoría de prueba: Pruebas de Administración e sesión

Identificador de prueba:	Pruebas de los atributos de las cookies
Objetivo:	Verificar que los atributos de seguridad de las cookies estén correctamente configurados para proteger la integridad y confidencialidad de las sesiones de usuario en una aplicación web.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org)
Herramientas Utilizadas:	OWASP ZAP

Tabla 31 Pruebas de los atributos de las cookies

Identificador de prueba:	Prueba para falsificación de petición de sitio cruzado (CSRF)
Objetivo:	Determinar si una acción que modifica información o ejecuta una operación relevante puede ser provocada mediante una solicitud construida desde un origen externo mientras la víctima mantiene una sesión autenticada.
Modalidad:	Caja negra, Caja gris
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend API (https://apiviteced.teced.org)
Herramientas Utilizadas:	Burp Suite

*Tabla 32 Prueba para falsificación de petición de sitio cruzado (CSRF)***Categoría de prueba: Pruebas de manejo de errores**

Identificador de prueba:	Pruebas de errores de código
Objetivo:	Identificar la información técnica o sensible expuesta mediante los mensajes de error generados por la aplicación web, el servidor y la base de datos.

Identificador de prueba:	Pruebas de errores de código
Modalidad:	Caja Negra
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend API REST (https://apiviteced1.teced.org)
Herramientas Utilizadas:	Burp Suite v2026.7.2

Tabla 33 Pruebas de errores de código

Categoría de prueba: Pruebas para criptografía débil

Identificador de prueba:	Pruebas para el envío de información sensible por canales sin encriptar
Objetivo:	Verificar que la aplicación web transmita la información sensible mediante canales de comunicación seguros y cifrados, evitando que credenciales, tokens, identificadores de sesión, cookies u otros datos sensibles sean enviados en texto claro o mediante protocolos y mecanismos de cifrado inseguros.
Entornos Evaluados:	Frontend (https://viteced.teced.org) y Backend API REST (https://apiviteced1.teced.org) y API para consulta a MongoDB (https://apimongo1.teced.org)
Herramientas Utilizadas:	Burp Suite v2026.7.2, Curl, Wireshark Network Analyzer

Tabla 34 Pruebas para el envío de información sensible por canales sin encriptar

Las pruebas de la aplicación móvil se encuentran en el informe final <https://zenodo.org/records/21956366>.

RESULTADOS

- Recolección de información**

Identificador de prueba: Conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información

Ejecución de la prueba:

Puntos evaluados	Consultas	Hallazgos encontrados
Diagramas de red y configuraciones	site:viteced.teced.org filetype:pdf OR filetype:xml OR filetype:conf OR filetype:cfg	Ninguna
	site:viteced.teced.org "network diagram" OR "configuracion" OR "topologia"	Ninguna
Mensajes archivados y correos electrónicos	site:viteced.teced.org "@teced.org" OR "admin@" OR "administrator" OR "correo"	Ninguna
Procedimientos de inicio de sesión y formatos de nombre de usuario	site:viteced.teced.org inurl:login OR inurl:signin OR inurl:auth	Ninguna
	site:viteced.teced.org "nombre de usuario" OR "usuario" OR "cédula"	Ninguna
Nombres de usuario y contraseñas	site:viteced.teced.org intext:"password" OR intext:"contraseña" OR intext:"passwd"	Ninguna
	site:viteced.teced.org intext:"username" OR intext:"user" filetype:env OR filetype:cfg	Ninguna
Contenido de mensajes de error	site:viteced.teced.org intext:"error" OR intext:"exception" OR intext:"stack trace"	Ninguna
	site:viteced.teced.org intext:"Warning:" OR intext:"Fatal error" OR intext:"undefined"	Ninguna
Versiones de desarrollo, prueba y UAT	site:viteced.teced.org dev OR test OR uat OR staging OR preprod OR qa	Ninguna
Puntos de apoyo o bases de apoyo (<i>Footholds</i>)	site:viteced.teced.org intitle:"index of" OR intitle:"directory listing"	Ninguna
	site:viteced.teced.org inurl:admin OR inurl:panel OR inurl:console	Ninguna
Archivos que contienen nombres de usuario	site:viteced.teced.org filetype:txt OR filetype:csv OR filetype:xls intext:"usuario"	Ninguna
	site:viteced.teced.org filetype:log intext:"user" OR intext:"login"	Ninguna
Directorios sensibles	site:viteced.teced.org intitle: index of /	Ninguna
	site:viteced.teced.org inurl:backup OR inurl:config OR inurl:db OR inurl:sql	Ninguna
Detección de servidores web	site:viteced.teced.org intext:"Apache" OR intext:"nginx" OR intext:"PHP"	Ninguna
	site:viteced.teced.org intext:"powered by" OR intext:"server" filetype:html	Ninguna
Archivos vulnerables	site:viteced.teced.org backup	Ninguna
	site:viteced.teced.org filetype:log	Ninguna
	site:viteced.teced.org .bak	Ninguna

Puntos evaluados	Consultas	Hallazgos encontrados
Servidores vulnerables	site:viteced.teced.org intext:"PHP/7" OR intext:"PHP/8" OR intext:"Apache/2"	Ninguna
	site:viteced.teced.org intext:"OpenSSL" OR intext:"mod_ssl"	Ninguna
Mensaje de error	site:viteced.teced.org intext:"SQL syntax" OR intext:"mysql_fetch" OR intext:"ORA-"	Ninguna
	site:viteced.teced.org intext:"stack trace" OR intext:"Traceback" OR intext:"Exception"	Ninguna
	site:viteced.teced.org intext:"undefined index" OR intext:"undefined variable"	Ninguna
	site:viteced.teced.org intext:"Internal Server Error" OR intext:"500 error"	Ninguna
	site:viteced.teced.org intext:"syntax error" OR intext:"uncaught exception"	Ninguna
Archivos que contienen información sensible	site:viteced.teced.org filetype:pdf OR filetype:doc OR filetype:xls intext:"confidencial"	Ninguna
	site:viteced.teced.org filetype:log OR filetype:txt intext:"internal" OR intext:"privado"	Ninguna
Archivos que contienen contraseñas	site:viteced.teced.org "DB_PASSWORD"	Ninguna
	site:viteced.teced.org "password"	Ninguna
	site:viteced.teced.org filetype:env	Ninguna

Tabla 35 Resultados de la prueba conducir motores de búsqueda para el descubrimiento y reconocimiento de fugas de información

Diagramas de red y configuraciones

Mensajes archivados y correos electrónicos

Procedimientos de inicio de sesión y formatos de nombre de usuario

Nombres de usuario y contraseñas

Contenido de mensajes de error

Versiones de desarrollo, prueba y UAT

Puntos de apoyo o bases de apoyo (Footholds)

Archivos que contienen nombres de usuario

Mensaje de error

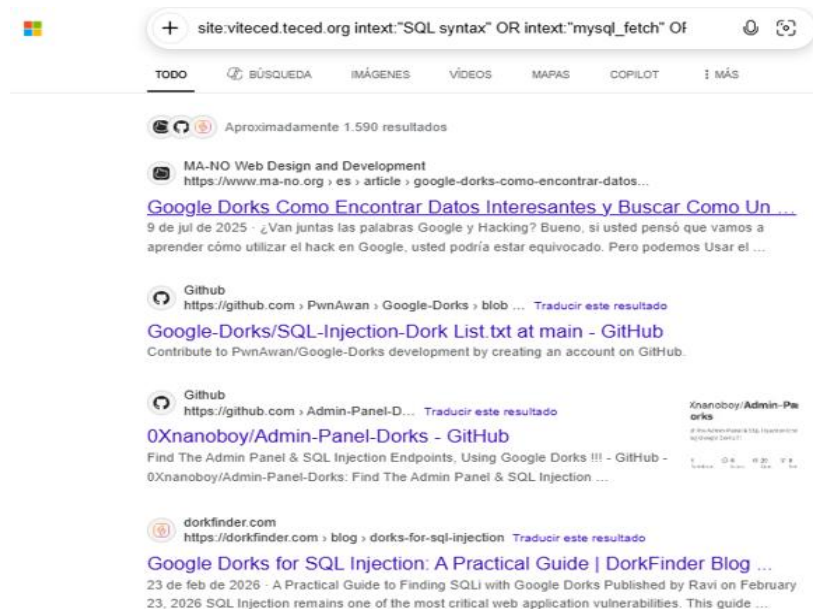


Figura 7 identificaron mensajes de error visibles en motores de búsqueda.

Fuente: Elaboración propia

Mediante la ejecución de la prueba, no se identificaron resultados relevantes que evidenciaran la divulgación de información del sistema, tal como se muestra en la Figura 7 y en la Tabla 35. Por lo tanto, no se hallaron resultados asociados.

Identificador de prueba: Huellas digitales en el servidor web

Ejecución de la prueba:

Comando: curl -I https://viteced.teced.org

Mediante la ejecución del comando curl -I https://viteced.teced.org como se muestra en la Figura 8, se identificó el encabezado Server: nginx/1.29.4, revelando tanto el software del servidor web como su versión específica. dominio apiviteced1.teced.org, se identificó adicionalmente el encabezado Server: Apache/2.4.66 (Debian) en las respuestas HTTP capturadas con la herramienta OWASP ZAP, evidenciando que la infraestructura utiliza diferentes servidores para el frontend y el backend. Asimismo, las herramientas WhatWeb y Netcraft confirmaron la presencia del servidor nginx 1.29.4 en el dominio

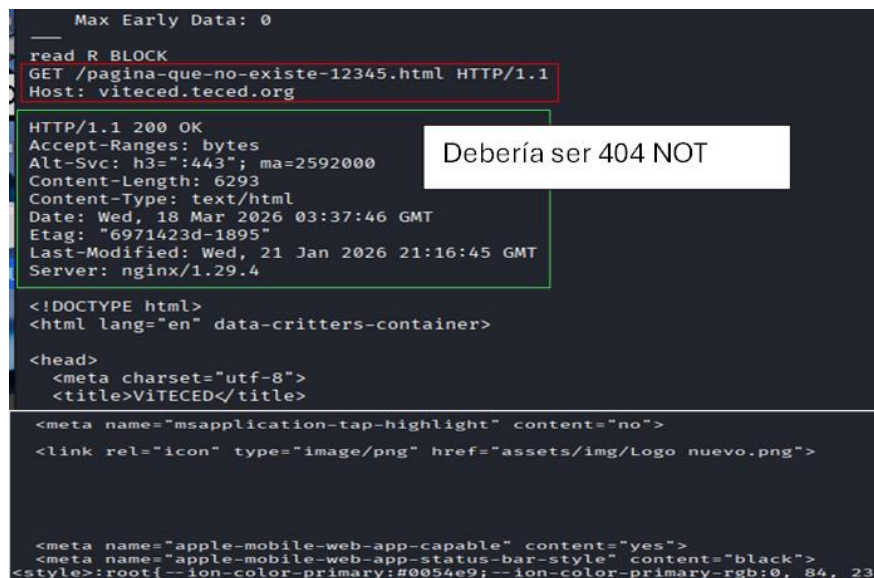
viteced.teced.org. Estas herramientas también proporcionaron información complementaria, como la dirección IP pública (72.61.11.127) y la ubicación geográfica del servidor (Estados Unidos).

Solicitud con versión HTTP no soportada

Al enviar una solicitud utilizando una versión HTTP no soportada (HTTP/3.0), el servidor respondió con el código 505 HTTP Version Not Supported, indicando que únicamente acepta versiones compatibles del protocolo y rechaza aquellas no reconocidas.

Solicitud de recurso inexistente

Al solicitar un recurso inexistente, el servidor respondió con el código 200 OK en lugar de 404 Not Found, devolviendo la página principal de la aplicación desarrollada con Angular/Ionic. Asimismo, la respuesta continuó incluyendo el encabezado Server: nginx/1.29.4, confirmando nuevamente la exposición de información sobre la tecnología empleada.



```
Max Early Data: 0
read R BLOCK
GET /pagina-que-no-existe-12345.html HTTP/1.1
Host: viteced.teced.org

HTTP/1.1 200 OK
Accept-Ranges: bytes
Alt-Svc: h3=":443"; ma=2592000
Content-Length: 6293
Content-Type: text/html
Date: Wed, 18 Mar 2026 03:37:46 GMT
Etag: "6971423d-1895"
Last-Modified: Wed, 21 Jan 2026 21:16:45 GMT
Server: nginx/1.29.4

<!DOCTYPE html>
<html lang="en" data-critters-container>

<head>
  <meta charset="utf-8">
  <title>ViTECED</title>

  <meta name="msapplication-tap-highlight" content="no">
  <link rel="icon" type="image/png" href="assets/img/Logo nuevo.png">

  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-status-bar-style" content="black">
<style>:root{--ion-color-primary:#0054e9;--ion-color-primary-rgb:0, 84, 23
```

Figura 8 Solicitud con métodos inexistentes

Fuente: Elaboracion propia

Hallazgo identificado

HALLAZGO VULN-001: Exposición del server en el encabezados HTTP.

Descripción: La aplicación web expone información sobre la infraestructura tecnológica mediante los encabezados HTTP de las respuestas. Durante la evaluación se identificó el tipo y la versión del servidor web Server: nginx/1.29.4, así como información relacionada con la tecnología Server: Apache/2.4.66 (Debian) utilizados en el backend. Esta información permite realizar un fingerprinting de la infraestructura y conocer componentes específicos empleados por la aplicación.

Impacto: La exposición del tipo y versión del servidor, junto con la tecnología y el framework del backend, proporciona información que puede ser utilizada por un atacante durante la fase de reconocimiento. Estos datos facilitan la identificación de vulnerabilidades conocidas y posibles exploits asociados con versiones específicas de los componentes expuestos, especialmente cuando estos presentan vulnerabilidades conocidas o no se encuentran actualizados.

Identificador de prueba: Revisión de los metas archivos del servidor web en busca de fugas de información

Ejecución de la prueba:

Acceso mediante el navegador web

Los resultados obtenidos, indican que el dominio evaluado no expone un archivo robots.txt accesible públicamente. Tanto la verificación manual mediante el navegador como la comprobación utilizando wget devolvieron la página principal de la aplicación, comportamiento característico de aplicaciones de página única (SPA) desarrolladas con Angular/Ionic, donde las solicitudes a rutas inexistentes son redirigidas al documento principal.

URL: <https://viteced.teced.org/robots.txt> desde el navegador

Método profesional

Desde la perspectiva de seguridad, la ausencia de un archivo robots.txt, tal como se muestra en la Figura 9, implica que no se divulgan rutas o directorios que puedan facilitar las actividades de reconocimiento mediante motores de búsqueda. Sin embargo, este resultado no permite afirmar si el sitio utiliza otros mecanismos para controlar la indexación de contenido, como encabezados HTTP o metaetiquetas HTML.

Comando: wget https://viteced.teced.org/robots.txt

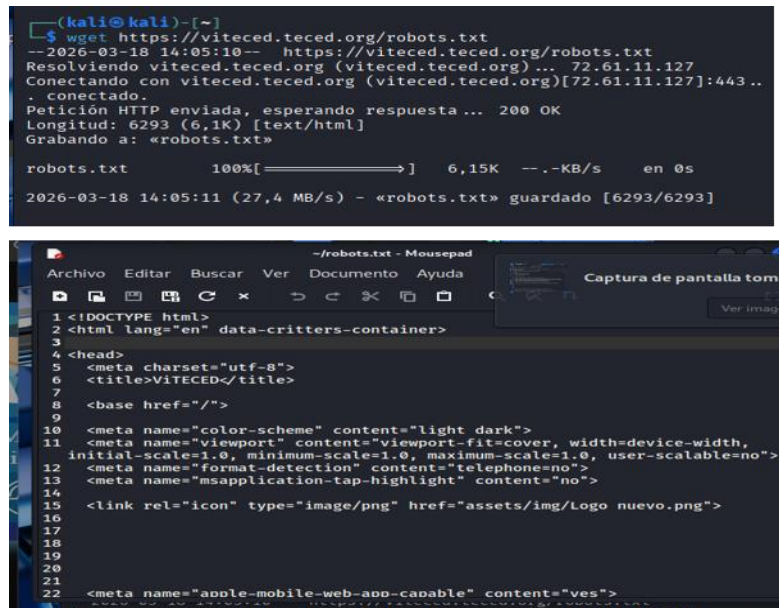


Figura 9 Verificación del archivo robots.txt mediante wget

Fuente: Elaboración propia

Identificador de prueba: Enumere las aplicaciones en el servidor web

Ejecución de la prueba:

Comando: nmap -Pn -sT -sV -p- IP

Descripción

- -Pn: No hace ping previo (asume que el host está activo)
- -sT: Escaneo TCP completo
- -sV: Detecta versión de servicios
- -p-: Escanea todos los puertos (1–65535)

Comando: nmap -PN -sT -sV -p0-9000 IP

Descripción

- -PN: No hace ping previo (asume que el host está activo)
- -sT: realiza un escaneo mediante conexión TCP, estableciendo una conexión completa con cada puerto para determinar su estado.

- -sV: activa la detección de versiones de servicios, permitiendo identificar el tipo de servicio que se ejecuta en los puertos abiertos y la versión del software utilizado.
- -p0-9000: indica que se deben analizar todos los puertos comprendidos entre el 0 y el 9000.

El escaneo identificó múltiples puertos TCP abiertos con diferentes servicios expuestos públicamente.

Entre los principales servicios detectados se encuentran:

PORT	STATE	SERVICE	VERSION
22/tcp	open	ssh	Openssh 9,6p1 ubuntu 3ubuntu13.15 (ubuntu linux; protocol 2.0)
80/tcp	open	tcpwrapped	
80/tcp	open	http	nginx 1.29.6
81/tcp	open	http	nginx 1.29.4 (ubuntu)
443/tcp	open	ssl/http	Golang net/http server
587/tcp	open	smtp-proxy	ISP SMTP block
1883/tcp	open	mqtt	
2377/tcp	open	ssl/grpc	
3000/tcp	open	http	Samsung Allshare httpd
3055/tcp	open	http	Gunicorn
3319/tcp	open	mysql	MariaDB 11.8.5
3366/tcp	open	mysql	MariaDB 11.8.6
3736/tcp	open	mysql	MariaDB 11.8.5
7946/tcp	open	unknown	
8080/tcp	open	tcpwrapped	
8080/tcp	open	http	Werkzeug httpd 3.1.4 (Python 3.12.3)

Tabla 36 Tablas de resultados de puertos abiertos

Los resultados de la Tabla 36, obtenidos permitieron conocer la cantidad de servicios accesibles desde la red externa. Debido a que la identificación realizada mediante Nmap puede presentar falsos positivos en algunos casos, se efectuó una validación manual posterior de los servicios detectados.

Verificación de cada puerto encontrado

Puerto 22 (SSH): Se encuentra abierto y ejecutando el servicio OpenSSH 9.6p1 sobre el sistema operativo Ubuntu. Este servicio permite el acceso remoto seguro al servidor

mediante el protocolo SSH, siendo comúnmente utilizado para la administración del sistema.

Intento de acceder SSH

A partir de los resultados obtenidos al intentar conectarse al puerto 22, se observa que el servicio solicita autenticación mediante contraseña, lo que indica que el acceso está restringido y requiere un usuario válido y credenciales correctas para poder ingresar.

Puerto 80 (HTTP): En el puerto 80 se evidencia una aplicación web denominada STEMLab TECED ("Science, Technology, Engineering and Mathematics Laboratory"), accesible directamente mediante la IP sin necesidad de resolución de nombre de dominio. Esta aplicación cuenta con módulos de Inicio, Convocatorias, Noticias, Proyectos y un panel de Administración.

Se determinó que dicha aplicación opera exclusivamente sobre el protocolo HTTP sin cifrado TLS/SSL, evidenciado por la advertencia "No seguro" emitida por el navegador. Esta condición implica que cualquier credencial o dato transmitido a través del panel de Administración viaja en texto plano, exponiendo la aplicación a ataques de interceptación (Man-in-the-Middle).

Puerto 443 (HTTPS): El puerto 443 se encuentra abierto y sobre este se ejecuta una aplicación desarrollada en el lenguaje Go (Golang). Al intentar acceder al servicio mediante un navegador web, se obtuvo un código de respuesta HTTP 404 (Not Found), lo que indica que la URL solicitada no existe en el servidor o no se encuentra disponible públicamente. Asimismo, se identificó que el servicio utiliza el servidor HTTP nativo de Golang (net/http), encargado de gestionar las comunicaciones cifradas con la aplicación web, garantizando la confidencialidad e integridad de los datos transmitidos.

Puerto 587 (SMTP): El puerto 587 fue identificado con el servicio "smtp-proxy ISP SMTP block", asociado al protocolo SMTP utilizado para el envío de correos electrónicos, esta identificación sugiere que se trata de un mecanismo de filtrado o proxy implementado por el proveedor de servicios de Internet (ISP), cuya función es restringir o controlar la comunicación SMTP para prevenir usos no autorizados o actividades maliciosas relacionadas con el envío de correos electrónicos.

Puerto 8080 y 81 (HTTP): En el puerto 8080 y 81 se detecta la presencia de una aplicación web, posiblemente un dashboard adicional de MQTT, que utiliza tecnologías como ANFIS IPS (sistema de detección y previsión de intrusos) y MQTT Security Dashboard (panel de monitoreo de tráfico MQTT). El protocolo MQTT se utiliza comúnmente en IoT, sensores y sistemas conectados. Sin embargo, el acceso a la página no permite consultar logs, tráfico MQTT ni dispositivos conectados, lo que limita la supervisión directa.

Puerto 7946: El puerto 7946 está abierto y corresponde a un servicio desconocido ya que aparece como “unknown”, aunque si está aceptando conexiones a nivel de red TCP, pero no expone de manera clara un servicio reconocible. Por lo general este puerto es muy usado por Docker Swarm (red de clúster) para comunicación entre nodos del clúster, intercambio de comunicación del estado de los nodos, y descubrimiento de servicios mediante TCP y UDP. Sin embargo, en este caso no se puede confirmar su uso específico.

Puerto 3736, 3366, 3319: Estos puertos están asociados a servicios de base de datos MySQL/MariaDB en distintas versiones el puerto 3319, 3736 con la versión 11.8.5 y puerto 3366 con la versión 11.8.6. La presencia de estos puertos en estado abierto indica que existen servicios de bases de datos accesibles a nivel de red, lo que permite el establecimiento de conexiones remotas.

En caso de no contar con mecanismos adecuados de autenticación, control de acceso y cifrado de la comunicación, estos servicios podrían representar un vector de ataque, facilitando intentos de acceso no autorizado mediante técnicas como fuerza bruta, así como la posible exfiltración de información o la explotación de vulnerabilidades propias del servicio de base de datos.

Verificación si el servicio de base de datos exige autenticación

Comando: `mysql -h 72.61.11.127 -P 3736 -u root -p`

Este comando se utiliza para intentar establecer una conexión remota hacia un servidor MySQL mediante la dirección IP y el puerto especificado. Además, permite verificar si el servicio de base de datos se encuentra accesible desde la red y si acepta autenticación con el usuario root. Este tipo de prueba ayuda a identificar configuraciones inseguras o exposiciones innecesarias del servicio de base de datos.

Puerto 3055: El puerto 3055/tcp es identificado como un servicio Gunicorn. El servicio Gunicorn es un servidor HTTP para aplicaciones web desarrolladas en Python. Normalmente este servicio se usa en APIs REST, backends de aplicación web, servicios de microservicios, lo que indica la presencia de un posible componente backend expuesto directamente.

Al verificar el puerto 3055 en el navegador se confirma que el servicio si es una aplicación web desarrollada en Django, sin embargo, la solicitud es rechazada debido a la configuración de seguridad `ALLOWED_HOSTS`, la cual restringe los dominios o direcciones IP permitidas para interactuar con el servicio.

Puerto 3000: Se identificó el puerto 3000 con el fingerprint Samsung Allshare httpd, resultado que constituye un falso positivo frecuente en dicho puerto.

La verificación manual mediante navegador confirmó la presencia de un panel de administración de contenedores que muestra un formulario de autenticación (correo y contraseña), accesible públicamente sin restricción de IP. Lo que representa un riesgo de seguridad porque cualquier persona que conozca la IP y el puerto podría intentar acceder, realizar ataques de fuerza bruta, explotar vulnerabilidades de la interfaz administrativa y, en caso de autenticación exitosa, modificar contenedores, ejecutar código arbitrario o extraer información confidencial.

Puerto 1883: El puerto 1883 está asociado al servicio MQTT, que generalmente se usa para la comunicación de dispositivos IoT y contenedores, y al estar expuesto públicamente puede permitir que un atacante intercepte, publique o suscriba mensajes sin autorización.

La verificación del puerto 1883, asociado al protocolo MQTT, mediante herramientas de escaneo y conexión directa, evidencia que el servicio se encuentra activo y accesible a nivel de red. Sin embargo, las pruebas realizadas con clientes MQTT y scripts de Nmap muestran que el bróker rechaza las conexiones por falta de autorización, lo que indica la implementación de mecanismos de autenticación y control de acceso. Este comportamiento sugiere una configuración segura del servicio, al impedir la interacción no autorizada con el sistema de mensajería.

Puerto 2377: El puerto 2377 se encuentra abierto, aunque no se identifica de manera explícita el servicio que lo utiliza. Cabe recalcar que este puerto es comúnmente empleado en entornos de contenedores Docker en modo Swarm.

El servicio asociado utiliza el protocolo gRPC sobre SSL/TLS, lo que indica que la comunicación entre servicios se realiza de forma cifrada. Sin embargo, la exposición de este puerto hacia redes externas representa un riesgo potencial de seguridad, ya que podría permitir la identificación o interacción no autorizada con el clúster en caso de configuraciones inadecuadas o controles de acceso insuficientes.

Resultados obtenidos

Punto evaluado	Descripción
Puertos no estándar	Puertos abiertos: 3000: Panel de administración accesible públicamente 3319, 3366, 3736: Servicios MariaDB expuestos a Internet 1883: Broker MQTT expuesto públicamente 2377: Puerto Docker Swarm/gRPC expuesto 7946: Servicio desconocido expuesto 8080, 81: Interfaces web adicionales activas 22: SSH accesible con autenticación requerida 587: SMTP filtrado por ISP 3055: Backend Django con ALLOWED_HOSTS restrictivo
Servicios de IP inversa	10 subdominios activos identificados, pero esto no representa una vulnerabilidad.
Transferencia de zona DNS (AXFR)	Transferencia denegada en todos los nameservers,

Identificador de prueba: Identificar puntos de entrada de la aplicación web

Ejecución de la prueba:

En la Tabla 36, se observan los 10 endpoints identificados mediante solicitudes HTTP GET, utilizados principalmente para la carga de recursos del frontend y la consulta de información del sistema, tales como la página principal (/), /inicio, /autenticacion, así como endpoints destinados a la consulta de catálogos y datos.

ID	MÉTODO	URL	ESTADO	ETIQUETA
1	GET	https://viteced.teced.org/	200 OK	Script
42	GET	https://apiviteced1.teced.org/listapais	200 OK	JSON
292	GET	https://apiviteced1.teced.org/listarmedico	200 OK	JSON
454	GET	https://viteced.teced.org/inicio	200 OK	Script
582	GET	https://apiviteced1.teced.org/listaparametros	200 OK	JSON
590	GET	https://broker.teced.org/mqtt	101 Switching Protocols	
689	GET	https://viteced.teced.org/autenticacion	200 OK	Script
717	GET	https://apiviteced1.teced.org/listacentro	200 OK	
718	GET	https://apiviteced1.teced.org/listaespe	200 OK	JSON
731	GET	https://apiviteced1.teced.org/listadoima?id_usuario=327	200 OK	JSON

Tabla 37 Registro de endpoints identificados mediante solicitudes GET

ID	MÉTODO	URL	ESTADO	ETIQUETA
45	POST	https://apiviteced1.teced.org/listaprovincias	200 OK	JSON
49	POST	https://apiviteced1.teced.org/listacantones	201 OK	
51	POST	https://apiviteced1.teced.org/vericorreo	202 OK	JSON
56	POST	https://apiviteced1.teced.org/crear_usuario	201 Created	JSON
58	POST	https://apiviteced1.teced.org/edidispotivo	200 OK	JSON
60	POST	https://apiviteced1.teced.org/login	200 OK	JSON
67	POST	https://apiviteced1.teced.org/listaralertas	200 OK	
82	POST	https://apiviteced1.teced.org/listar_datosfisicos	200 OK	JSON
84	POST	https://apiviteced1.teced.org/listarHistorialDatosFisicos	200 OK	JSON
112	POST	https://apiviteced1.teced.org/Buscarxcedula_ide	200 OK	JSON
116	POST	https://apiviteced1.teced.org/nuevochatmedi	200 OK	JSON
121	POST	https://apiviteced1.teced.org/nuevochatpaci	200 OK	JSON
129	POST	https://apiviteced1.teced.org/agregarima	200 OK	JSON
151	POST	https://apiviteced1.teced.org/editar_usuario	200 OK	JSON
291	POST	https://apiviteced1.teced.org/listamedi	200 OK	JSON
305	POST	https://apiviteced1.teced.org/Guardarmedico	200 OK	JSON
334	POST	https://apiviteced1.teced.org/listarchat	200 OK	JSON

ID	MÉT ODO	URL	ESTADO	ETIQU ETA
348	POST	https://apiviteced1.teced.org/enviarmensaje	200 OK	JSON
374	POST	https://apiviteced1.teced.org/listar_horar_tel e	200 OK	JSON
386	POST	https://apiviteced1.teced.org/listaralertasfech a	200 OK	JSON
469	POST	https://apiviteced1.teced.org/listaranteper	200 OK	JSON
488	POST	https://apiviteced1.teced.org/LISTARECET AS	200 OK	JSON
583	POST	https://apiviteced1.teced.org/obtenerTopicfo rUser	400 Bad Request	JSON
586	POST	https://apiviteced1.teced.org/obtenerParamet rosEstadistica	200 OK	JSON
626	POST	https://apiviteced1.teced.org/obtenerdatosagr upados	200 OK	JSON
707	POST	https://apiviteced1.teced.org/verificarCedula	200 OK	JSON
709	POST	https://apiviteced1.teced.org/VerificarMedic os	200 OK	JSON
715	POST	https://apiviteced1.teced.org/crear_usuariom edico	201 Created	JSON
720	POST	https://apiviteced1.teced.org/crear_medico	200 OK	JSON
722	POST	https://apiviteced1.teced.org/crear_horarios	200 OK	JSON
758	POST	https://apiviteced1.teced.org/obtenerDatosPa cienteDeshabilitados	200 OK	JSON
759	POST	https://apiviteced1.teced.org/listapacientes	200 OK	JSON
760	POST	https://apiviteced1.teced.org/medicoxi	200 OK	JSON
781	POST	https://apiviteced1.teced.org/listarasistente	200 OK	JSON
909	POST	https://apiviteced1.teced.org/asignarAsistent e	200 OK	JSON
965	POST	https://apiviteced1.teced.org/buscararticulos porusuarios	200 OK	JSON
1036	POST	https://apiviteced1.teced.org/guardararticulo s	200 OK	JSON
1056	POST	https://apiviteced1.teced.org/editararticulos	200 OK	JSON
1069	POST	https://apiviteced1.teced.org/pacientemed_as is	200 OK	JSON
939	POST	https://apiviteced1.teced.org/obtenerAsistent esDisponibles	200 OK	JSON

Tabla 38 Registro de endpoints identificados mediante solicitudes POST

En la tabla 38 se evidencia los 40 endpoints identificados mediante solicitudes HTTP POST, utilizados para el envío de información al servidor. Estos endpoints soportan funcionalidades relacionadas con la autenticación, el registro y actualización de usuarios, la mensajería, la gestión de información médica y otras operaciones que requieren el procesamiento de datos por parte de la API.

Análisis detallado de parámetros y encabezados

Parámetros en peticiones GET

Mediante el análisis de las solicitudes HTTP GET, se identificó que la aplicación utiliza parámetros visibles en la URL para realizar consultas de información. Como se muestra en la Tabla 39, estos parámetros son empleados principalmente para recuperar datos asociados a usuarios e imágenes almacenadas en el sistema.

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
731	GET	https://apiviteced1.teced.org/estadoima?id_usuario={id}	id_usuario	JSON	Si

Tabla 39 Parámetros identificados en la petición GET

Nota Técnica de Seguridad: La presencia del parámetro id_usuario en la URL requiere que el backend valide estrictamente la autorización del token JWT para evitar vulnerabilidades de Referencia Directa Insegura a Objetos (IDOR / BOLA), garantizando que un usuario no pueda consultar datos de otros cambiando dicho ID.

Parámetros en peticiones POST

En el análisis de la Tabla 40, se examinó el cuerpo (*body*) de las 40 solicitudes HTTP POST interceptadas, confirmando que todas ellas reciben y transmiten datos estructurados en formatos JSON y Form-Data. Entre la información procesada se identificaron credenciales de autenticación, registros de pacientes, coordenadas geográficas y datos relacionados con los servicios de telemedicina.

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
45	POST	https://apiviteced1.teced.org/listaprovincias	"id_pais":1	JSON	No

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
49	POST	https://apiviteced1.teced.org/listacantones	"id_provincia":2		No
51	POST	https://apiviteced1.teced.org/vericorreo	"correo_electronico": "jorges@gmail.com"	JSON	No
56	POST	https://apiviteced1.teced.org/crear_usuario	"nombre": "JORGE", "apellido": "SALTOS", "cedula": "0151692860", "Genero": "MASCULINO", "id_provincia": 2, "id_pais_origen": 4, "id_canton": 9, "correo_electronico": "jorges@gmail.com", "fecha_nacimiento": "1998-05-06", "contrasena": "0151692860", "telefono": "0999999999", "id_rol": "1", "estado": "A", "username": "0151692860"	JSON	No
58	POST	https://apiviteced1.teced.org/edidispotivo	"codigo": "BIO/CV8N1", "clave": "CV8N1", "id_usuario": 326	JSON	No
60	POST	https://apiviteced1.teced.org/login	name="username" 0151692860, name="contrasena" 0151692860	JSON	No
67	POST	https://apiviteced1.teced.org/listaralertas	"id_usuario": 326		Si
82	POST	https://apiviteced1.teced.org/listar_datosfisicos	"id_usuario": 326	JSON	Si

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
84	POST	https://apiviteced1.teced.org/listarHistorialDatosFisicos	"id_usuario":326	JSON	Si
112	POST	https://apiviteced1.teced.org/Buscarxcedula_ide	"id_usuario":326	JSON	Si
116	POST	https://apiviteced1.teced.org/nuevochatmedi	"id_usuario":326	JSON	Si
121	POST	https://apiviteced1.teced.org/nuevochatpaci	"id_usuario":326	JSON	Si
129	POST	https://apiviteced1.teced.org/agregarima	"id_usuario":326,"foto": base64 JPEG	JSON	Si
151	POST	https://apiviteced1.teced.org/editar_usuario	"id_usuario":326,"correo_electronico": "jordy@gmail.com", "telefono": "0999999999"	JSON	Si
291	POST	https://apiviteced1.teced.org/listmedi	"id_usuario":326	JSON	Si
305	POST	https://apiviteced1.teced.org/Guardarmedico	"id_usuario":326,"id_medico":40,"acepta_correo":1	JSON	Si
334	POST	https://apiviteced1.teced.org/listarchat	"id_usuario":326,"id_medico":40	JSON	Si
348	POST	https://apiviteced1.teced.org/enviarmensaje	"id_usuario":326,"id_medico":40,"mensaje": "buenos dias", "id_rol":1	JSON	Si
374	POST	https://apiviteced1.teced.org/listar_horar_tele	"id_medico":40	JSON	Si
386	POST	https://apiviteced1.teced.org/listaralertasfecha	"id_usuario":326, "fecha_ini": "", "fecha_fin": ""	JSON	Si
469	POST	https://apiviteced1.teced.org/listaranteper	"id_paciente":326	JSON	Si
488	POST	https://apiviteced1.teced.org/LISTARECETAS	"id_paciente":326,"anio":2026	JSON	Si

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
626	POST	https://apiviteced1.teced.org/obtenerdatosagrupados	"id_usuario":326,"id_parametro":2,"fechaInicio":"2025-01-01","fechaFin":"2025-12-31"	JSON	Si
707	POST	https://apiviteced1.teced.org/verificarCedula	"cedula":"0603339664"	JSON	Si
709	POST	https://apiviteced1.teced.org/VerificarMedicos	"cedula":"0603339664"	JSON	Si
715	POST	https://apiviteced1.teced.org/crear_usuariomedico	"nombre":"NO DISPONIBLE","apellido":"RODRIGUEZ CESAR","cedula":"0603339664","fecha_nacimiento":"1988-05-10","correo_electronico":"cesar@gmail.com","telefono":"0999999999","id_pais":1,"id_provincia":2,"id_canton":11,"username":"0603339664","contrasena":"0603339664","id_rol":"3","estado":"A","id_pais_origen":1,"Genero":"MASCULINO"	JSON	No
720	POST	https://apiviteced1.teced.org/crear_medico	"id_usuario":"327","id_centro":2,"especialidades":[1,2,4,9], "estado":"A","codigo_medico":"sdncuvbybvy"	JSON	No
722	POST	https://apiviteced1.teced.org/crear_horarios	"id_usuario":"327","hora_inicio":"08:00"	JSON	No

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
			0", "hora_fin": "17:00", "lunes": false, "martes": false, "miercoles": true, "jueves": true, "viernes": true, "sabado": true, "domingo": false		
758	POST	https://apiviteced1.teced.org/obtenerDatosPacienteDeshabilitados	"id_usuario": 327	JSON	Si
759	POST	https://apiviteced1.teced.org/listapacientes	"id_usuario": 327	JSON	Si
760	POST	https://apiviteced1.teced.org/medicoxi	"id_usuario": 327	JSON	Si
781	POST	https://apiviteced1.teced.org/listarasistente	"id_usuario": 327	JSON	Si
909	POST	https://apiviteced1.teced.org/asignarAsistente	"id_usuario": 327, "id_asistente": 22	JSON	Si
965	POST	https://apiviteced1.teced.org/buscararticulosporusuarios	"id_usuario": 327	JSON	Si
1036	POST	https://apiviteced1.teced.org/guardararticulos	"id_usuario": 327, "titulo": "Metodología OWASP Un Enfoque para la Prevención y Resolución de Vulnerabilidades", "autor": "NO DISPONIBLE", "enlace": "https://dialnet.unirioja.es/servlet/articulo?codigo=10118346", "descripcion": "mi articulo"	JSON	Si
1056	POST	https://apiviteced1.teced.org/editararticulos	"id": 10, "titulo": "Metodología OWASP Un	JSON	Si

ID	Método	Endpoint / URL	Parámetros Identificados	Tipo de Payload	Requiere Auth
			Enfoque para la Prevención y Resolución de Vulnerabilidad", "autor": "NO DISPONIBLE", "descripcion": "mi articulo", "enlace": "https://dialnet.unirioja.es/servlet/articulo?codigo=10118346", "estado": ""		
1069	POST	https://apiviteced1.teced.org/pacientemed_asis	"id_usuario": 327	JSON	Si
939	POST	https://apiviteced1.teced.org/obtenerAsistentesDisponibles	"id_usuario": 327	JSON	Si

Tabla 40 Parámetros identificados en las peticiones POST

Campos Ocultos: No se identificaron parámetros ocultos en el código HTML de las peticiones POST.

Evidencia

URL: https://apiviteced1.teced.org/listadoima?id_usuario=326

URL: <https://apiviteced1.teced.org/listarmedico>

URL: <https://apiviteced1.teced.org/listaprovincias>

URL: <https://apiviteced1.teced.org/vericorreo>

URL: https://apiviteced1.teced.org/crear_usuario

URL: <https://apiviteced1.teced.org/edidispositivo>

URL: <https://apiviteced1.teced.org/listaralertas>

Análisis de Encabezados en solicitudes HTTP

- **Encabezados Adicionales / Propios de la Aplicación:** Se identificó la presencia del encabezado Authorization: Bearer <JWT> en 35 endpoints del backend.

Este encabezado fue programado explícitamente en el frontend para enviar un token *JSON Web Token (JWT)* que transporta los datos de autenticación del usuario (id_usuario, nombre, id_rol).

- **Encabezados Estándar del Navegador:** Los encabezados Host, User-Agent, Referer u Origin corresponden a valores estándar generados automáticamente por el navegador.
- **Encabezados Personalizados:** No se identificaron encabezados personalizados (como X-Debug).

Análisis de encabezados de respuesta del servidor

Mediante la inspección de los encabezados de respuesta HTTP enviados por el servidor, se identificó información relacionada con la infraestructura tecnológica y las configuraciones de seguridad implementadas por el sistema. Como se presenta en la Tabla 40, se registraron los principales encabezados de respuesta observados durante la evaluación.

Encabezado de Respuesta	Ubicación / Componente	Valor Identificado / Aplicación	Implicación de Seguridad
Server	Frontend	nginx/1.29.4	Revela la versión exacta del servidor web frontend.
Server	Backend API	Apache/2.4.66 (Debian)	Revela el servidor web y SO del backend en las 47 endpoints de la APIs.
X-Powered-By	Backend API	PHP/8.3.30	Expone la versión de PHP en rutas de provincias/imágenes.
X-Powered-By	Backend API	Fat-Free Framework	Expone el framework PHP utilizado en 37 endpoints.
Access-Control-Allow-Origin	Backend API	* (Permisivo)	CORS Abierto: Permite solicitudes desde cualquier dominio externo.
X-Frame-Options	Backend API	SAMEORIGIN	Protección contra ataques de Clickjacking.
X-Xss-Protection	Backend API	1; mode=block	Filtro de protección XSS activado.

Encabezado de Respuesta	Ubicación / Componente	Valor Identificado / Aplicación	Implicación de Seguridad
X-Content-Type-Options	Backend API	nosniff	Previene la suposición de tipo MIME por parte del navegador.

Tabla 41 Encabezados adicionales identificados - OWASP ZAP

Códigos de Estado HTTP:

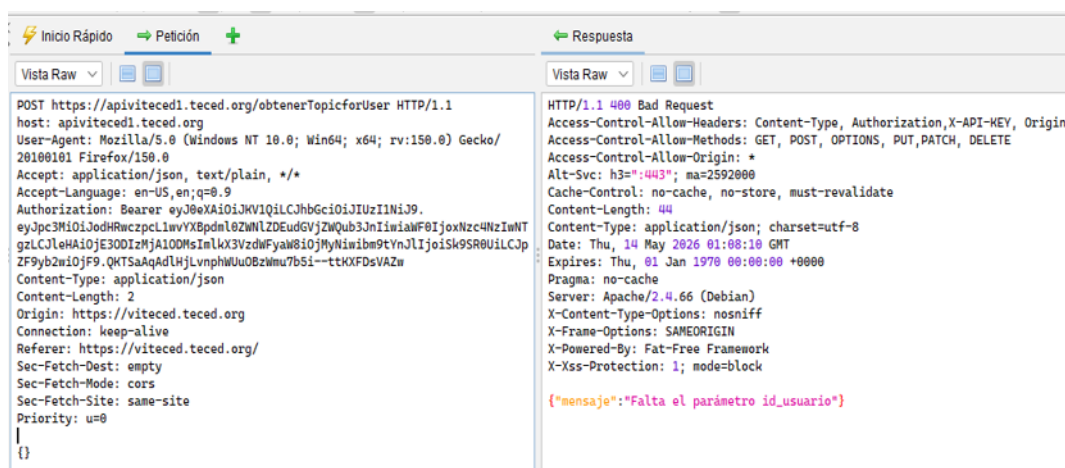


Figura 10 Encabezados de respuesta HTTP identificados durante la evaluación - OWASP ZAP.

Fuente: Elaboración propia

Como se muestra en la Figura 10, se identificó una respuesta HTTP 400 Bad Request correspondiente al endpoint /obtenerTopicforUser, generada al enviar una solicitud sin los parámetros requeridos. La respuesta permitió analizar los encabezados HTTP devueltos por el servidor y la estructura del mensaje de error asociado.

Nuevas Cookies: Durante el análisis no se identificó la generación de cookies de sesión adicionales (Set-Cookie).

Identificador de prueba: Marco referencial para el uso de huellas digitales en aplicaciones web

Ejecución de la prueba:

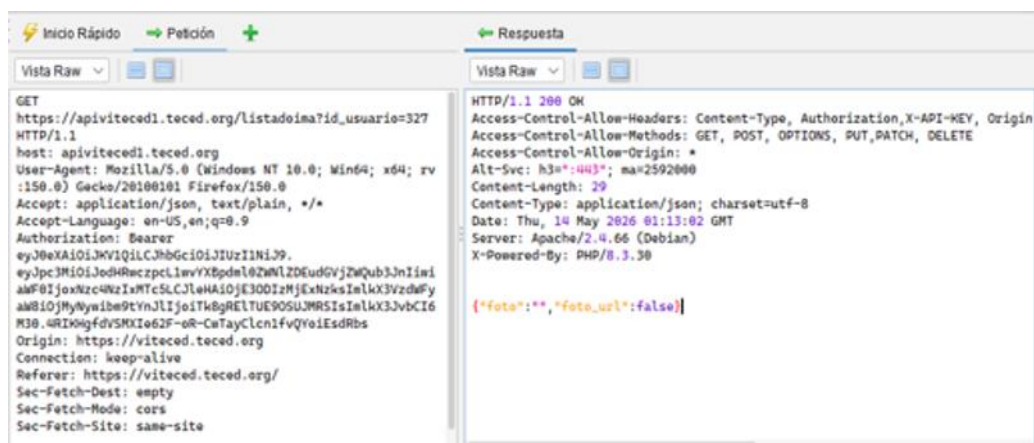


Figura 11 Identificación de framework mediante el encabezado X-Powered-By

Fuente: Elaboración propia

Sin embargo, al evaluar el subdominio del backend (<https://apiviteced1.teced.org>), como se muestra en la Figura 11, reveló la exposición de tres indicadores tecnológicos clave a través del encabezado X-Powered-By: el framework Fat-Free Framework, el lenguaje de programación PHP y su versión exacta 8.3.30. Esta información, si bien no representa por sí sola una vulnerabilidad explotable, proporciona a un atacante un perfil tecnológico preciso del backend, facilitando la búsqueda dirigida de vulnerabilidades o exploits específicos para dichas versiones.

Análisis de Cookies

Mediante la ejecución del comando `curl -I https://viteced.teced.org | grep -i "set-cookie"`, y la revisión de los encabezados HTTP en el navegador durante una sesión autenticada, no se identificó la presencia de cookies de sesión tradicionales, tales como PHPSESSID, laravel_session o similares. Este resultado indica que la aplicación no utiliza un mecanismo de gestión de sesiones basado en cookies HTTP. En su lugar, el proceso de autenticación emplea un token de acceso (JWT) que es devuelto en el cuerpo de la respuesta tras un inicio de sesión exitoso y posteriormente utilizado para autenticar las solicitudes realizadas por el cliente.

Comando: `curl -I https://viteced.teced.org | grep -i "set-cookie"`

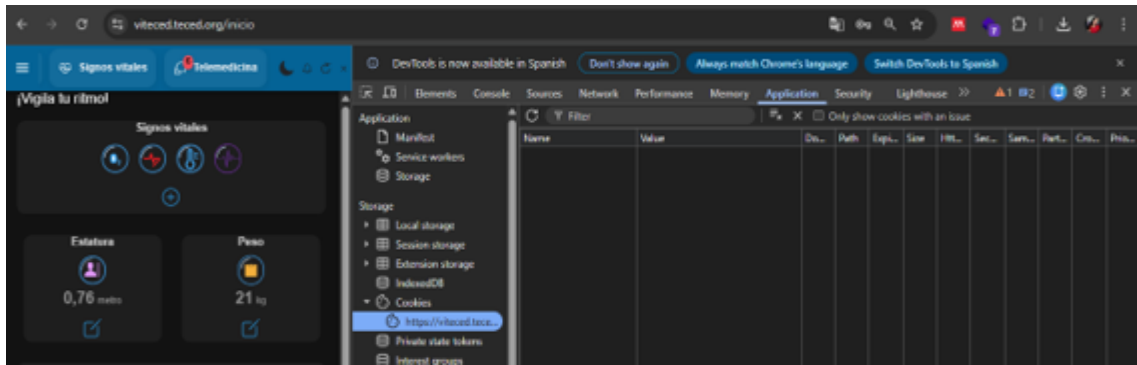


Figura 11 Inspección cookies en el código HTML.

Fuente: Elaboración propia

Análisis: El sistema no utiliza cookies tradicionales de sesión, sino autenticación basada en JSON Web Tokens (JWT) mediante el encabezado Authorization: Bearer.

Inspección del código fuente HTML

La inspección del código fuente HTML del dominio <https://viteced.teced.org>, realizada mediante las herramientas para desarrolladores del navegador. Durante el análisis no se evidenció la presencia de etiquetas `<meta name="generator">` ni de comentarios HTML que divulgaran información sobre el framework utilizado por la aplicación. Sin embargo, se identificaron etiquetas propias de Ionic Framework y archivos JavaScript generados durante el proceso de compilación de Angular, entre ellos:

- `<script src="runtime.a96c21713a2a85f3.js" type="module"></script>`
- `<script src="polyfills.d55c87604a563394.js" type="module"></script>`
- `<script src="main.c87070f591f0b90c.js" type="module"></script>`

Los resultados proporcionados por Wappalyzer fueron consistentes con la información obtenida mediante la inspección del código fuente HTML, confirmando que el frontend de la aplicación fue desarrollado utilizando Angular 18.2.13 en conjunto con Ionic Framework.

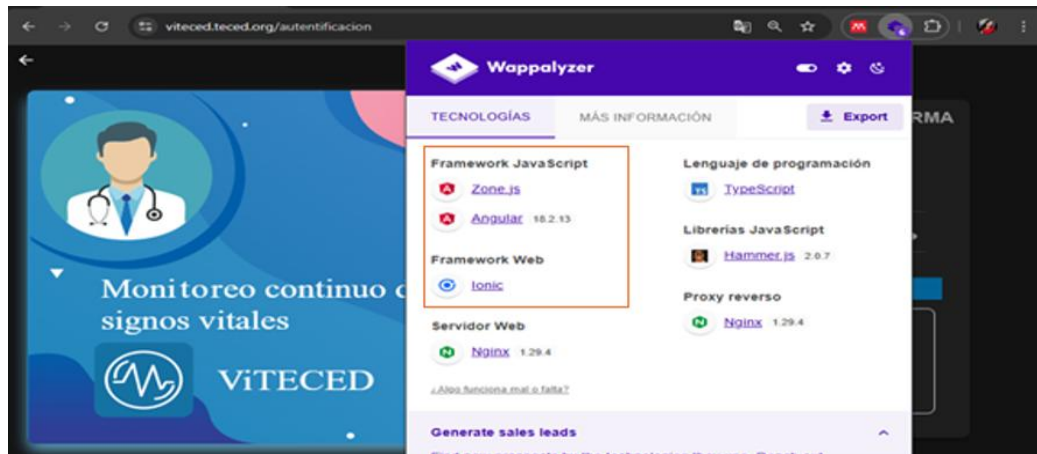


Figura 13 Resultado del análisis de Wappalyzer sobre el dominio principal del sistema

Fuente: Elaboración propia

Enumeración de Carpetas y Archivos Específicos

Como se muestra en la Figura 15, se realizó la enumeración de directorios y archivos mediante la herramienta Gobuster sobre el dominio <https://viteced.teced.org>, evaluando un total de 4 613 rutas. Los resultados no evidenciaron la presencia de directorios característicos de sistemas de gestión de contenidos (CMS) como WordPress, Joomla o Drupal.

Durante la enumeración se identificó la siguiente ruta activa:

- /assets/ (Código de estado HTTP 301 Moved Permanently).

La presencia de /assets/ es propia de la estructura estática de aplicaciones de página única (SPA) bajo el framework Angular.

Comando: gobuster dir -u <https://viteced.teced.org> -w /usr/share/wordlists/dirb/common.txt --exclude-length 6293

```
(kali㉿kali)-[~]
$ gobuster dir -u https://viteced.teced.org \
-w /usr/share/wordlists/dirb/common.txt \
--exclude-length 6293

Gobuster v3.8.2
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:             https://viteced.teced.org
[+] Method:          GET
[+] Threads:         10
[+] Wordlist:         /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] Exclude Length:  6293
[+] User Agent:      gobuster/3.8.2
[+] Timeout:         10s

Starting gobuster in directory enumeration mode

assets                (Status: 301) [Size: 169] [→ http://viteced.teced.org/a
ssets/]
Progress: 4613 / 4613 (100.00%)

Finished
```

Figura 14 Resultado de la enumeración de directorios con Gobuster sobre el dominio del sistema

Fuente: Elaboración propia

Hallazgo identificado

HALLAZGO VULN-01: Exposición de framework en el encabezados http backend

Descripción: La aplicación web expone activamente la tecnología y el framework utilizado en la respuesta de sus peticiones mediante el encabezado X-Powered-By, facilitando la identificación exacta de la estructura tecnológica del backend (Fat-Free Framework).

Impacto: Divulgación de la versión exacta del intérprete PHP y del framework backend, facilitando la búsqueda de vulnerabilidades o exploits específicos por parte de un atacante.

- Pruebas para gestionar la configuración y la implementación

Identificador de prueba: Prueba de Métodos HTTP

Ejecución de la prueba:

Métodos HTTP Habilitados (OPTIONS)

La aplicación fue sometida a solicitudes mediante el método HTTP OPTIONS dirigidas al dominio del backend, con el propósito de identificar los métodos HTTP anunciados por el servidor y determinar las operaciones que se encuentran habilitadas para su procesamiento.

API para consulta MongoDB

curl -v -X OPTIONS https://apimongo1.teced.org/

```
(kali@kali)-[~]
$ curl -v -X OPTIONS https://apimongo1.teced.org/
* Host apimongo1.teced.org:443 was resolved.
* Request completely sent off
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 405
< allow: GET
< alt-svc: h3=":443"; ma=2592000
< content-type: application/json
< date: Sat, 08 Aug 2026 01:44:12 GMT
< server: uvicorn
< strict-transport-security: max-age=31536000; includeSubDomains
< content-length: 31
<
* Connection #0 to host apimongo1.teced.org:443 left intact
{"detail":"Method Not Allowed"}
```

Figura 15 Análisis de métodos HTTP OPTIONS en la API Mongo - Curl.

Fuente: Elaboración propia

API REST

curl -v -X OPTIONS https://apiviteced1.teced.org/

```
(kali@kali)-[~]
$ curl -v -X OPTIONS https://apiviteced1.teced.org/
* Host apiviteced1.teced.org:443 was resolved.
> OPTIONS / HTTP/2
> Host: apiviteced1.teced.org
> User-Agent: curl/8.19.0
> Accept: */*
>
* Request completely sent off
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 200
< access-control-allow-headers: Content-Type, Authorization,X-API-KEY, Origin, X-Requested-With, Co
ntent-Type, Accept, Access-Control-Request-Method
< access-control-allow-methods: GET, POST, OPTIONS, PUT,PATCH, DELETE
< access-control-allow-origin: *
< allow: GET
< alt-svc: h3=":443"; ma=2592000
< content-type: application/json; charset=utf-8
< date: Fri, 05 Jun 2026 18:43:24 GMT
< server: Apache/2.4.66 (Debian)
< x-powered-by: PHP/8.3.30
< content-length: 0
<
* Connection #0 to host apiviteced1.teced.org:443 left intact
```

Figura 16 Análisis de métodos HTTP OPTIONS en la API REST – Curl

Fuente: Elaboración propia

La evaluación del método HTTP OPTIONS permitió verificar la información que los servicios proporcionan respecto a los métodos HTTP disponibles en los recursos evaluados. En la Figura 16, correspondiente al dominio de las API apimongo1.teced.org, se observa que la solicitud OPTIONS fue rechazada mediante una respuesta 405 Method Not Allowed. En contraste, en la Figura 17, correspondiente al servicio API REST apiviteced1.teced.org, la solicitud OPTIONS obtuvo una respuesta 200 OK. En ambas respuestas se identificó el encabezado Allow: GET, asociado al recurso consultado. Sin embargo, en la petición de la API REST se evidencio el encabezado Access-Control-Allow-Methods en la cual declara los métodos GET, POST, OPTIONS, PUT, PATCH y

DELETE, métodos que un navegador puede utilizar desde otros orígenes. Asimismo, se observó Access-Control-Allow-Origin: *, lo que indica que el servicio permite solicitudes de origen cruzado conforme a su configuración CORS.

Observación: La diferencia entre Allow y Access-Control-Allow-Methods no representa una inconsistencia de seguridad, ya que Allow refleja los métodos aceptados por el recurso específico consultado, mientras que Access-Control-Allow-Methods corresponde a la configuración CORS global del servicio. Dado que la API posee una arquitectura REST, los diferentes endpoints pueden utilizar distintos métodos HTTP según la operación requerida, por lo que la respuesta OPTIONS sobre la raíz del servicio no representa un inventario completo de todos los métodos disponibles.

Métodos HTTP Peligrosos (TRACE / PUT / DELETE / CONNECT)

Se enviaron solicitudes individuales con cada método peligroso hacia ambos componentes:

API para consulta MongoDB

Comando: curl -v -X TRACE https://apimongo1.teced.org/

Comando: curl -v -X PUT https://apimongo1.teced.org/

Comando: curl -v -X DELETE https://apimongo1.teced.org/

Comando: curl -v -X CONNECT https://apimongo1.teced.org/

API REST

Comando: curl -v -X PUT https://apiviteced1.teced.org/

Comando: curl -v -X DELETE https://apiviteced1.teced.org/

Comando: curl -v -X TRACE https://apiviteced1.teced.org/

Comando: curl -v -X CONNECT https://apiviteced1.teced.org/

Resultados de evaluación de los métodos HTTP peligroso

Método	Componente	Código de Respuesta	Interpretación
TRACE	API Mongo	405 Method Not Allowed	Rechazado. No es posible ejecutar ataques XST.

Método	Componente	Código de Respuesta	Interpretación
PUT	API Mongo	405 Method Not Allowed	Rechazado. No es posible cargar archivos no autorizados.
DELETE	API Mongo	405 Method Not Allowed	Rechazado. No es posible eliminar recursos.
CONNECT	API Mongo	404 Not Found	No procesado. Servidor no operable como proxy.
TRACE	API REST	405 Method Not Allowed	Rechazado. No es posible ejecutar ataques XST.
PUT	API REST	405 Method Not Allowed	Rechazado. No es posible cargar archivos no autorizados.
DELETE	API REST	405 Method Not Allowed	Rechazado. No es posible eliminar recursos.
CONNECT	API REST	404 Not Found	No procesado. Servidor no operable como proxy.

Tabla 42 Resultados de la evaluación de métodos HTTP potencialmente peligrosos en la API de la plataforma

Como parte de la evaluación de los métodos HTTP potencialmente peligrosos, se realizaron solicitudes dirigidas tanto a la API Mongo y a la API REST, cuyos resultados se presentan en la Tabla 42. Las pruebas permitieron verificar el comportamiento del servidor frente a los métodos TRACE, PUT, DELETE y CONNECT, con el propósito de determinar si estos se encontraban habilitados y si podían ser utilizados para ejecutar acciones no autorizadas sobre los recursos del sistema.

En la API Mongo, las solicitudes realizadas mediante los métodos TRACE, PUT y DELETE fueron rechazadas con el código HTTP 405 Method Not Allowed, lo que evidencia que dichos métodos no se encuentran habilitados para su procesamiento. De igual manera, la solicitud mediante CONNECT obtuvo una respuesta HTTP 404 Not Found, por lo que no se evidenció un comportamiento que permitiera utilizar el servidor como proxy.

En la API REST se obtuvo un comportamiento equivalente. Los métodos TRACE, PUT y DELETE fueron rechazados mediante respuestas HTTP 405 Method Not Allowed, mientras que el método CONNECT generó una respuesta HTTP 404 Not Found. Estos resultados indican que no se identificó la habilitación de estos métodos para realizar operaciones no autorizadas sobre los recursos evaluados.

Métodos HTTP Arbitrarios (JEFF / CATS / FOOBAR)

Se enviaron solicitudes con métodos HTTP completamente inventados para verificar si el servidor los procesa internamente como solicitudes GET, lo que permitiría eludir controles de acceso:

Método Arbitrario	Componente	Código de Respuesta	¿Procesado como GET?
JEFF	API Mongo	404 Not Found	No
CATS	API Mongo	404 Not Found	No
FOOBAR	API Mongo	404 Not Found	No
JEFF	API REST	405 Method Not Allowed	No
CATS	API REST	405 Method Not Allowed	No
FOOBAR	API REST	405 Method Not Allowed	No

Tabla 43 Resultados de la evaluación de métodos HTTP arbitrarios en la API y la aplicación web

La evaluación de métodos HTTP arbitrarios permitió analizar el comportamiento de los componentes API Mongo y API REST frente a solicitudes construidas con métodos no estandarizados, utilizando JEFF, CATS y FOOBAR como métodos de prueba. El objetivo fue determinar si estas solicitudes eran aceptadas por los servicios, procesadas de manera inesperada o interpretadas automáticamente como solicitudes GET.

Los resultados correspondientes al componente API Mongo. Las solicitudes realizadas mediante los métodos arbitrarios JEFF, CATS y FOOBAR obtuvieron una respuesta HTTP 404 Not Found, lo que indica que el servidor no identificó un recurso válido asociado a las solicitudes realizadas. Asimismo, no se evidenció un procesamiento alternativo de estos métodos ni su interpretación como solicitudes GET. Por otra parte, los resultados obtenidos en el componente API REST, donde las solicitudes mediante los métodos JEFF, CATS y FOOBAR recibieron una respuesta HTTP 405 Method Not Allowed. Esta respuesta indica que el método HTTP utilizado no está permitido para el recurso solicitado. Al igual que en el componente API Mongo, no se observó que los métodos arbitrarios fueran convertidos o procesados como solicitudes GET.

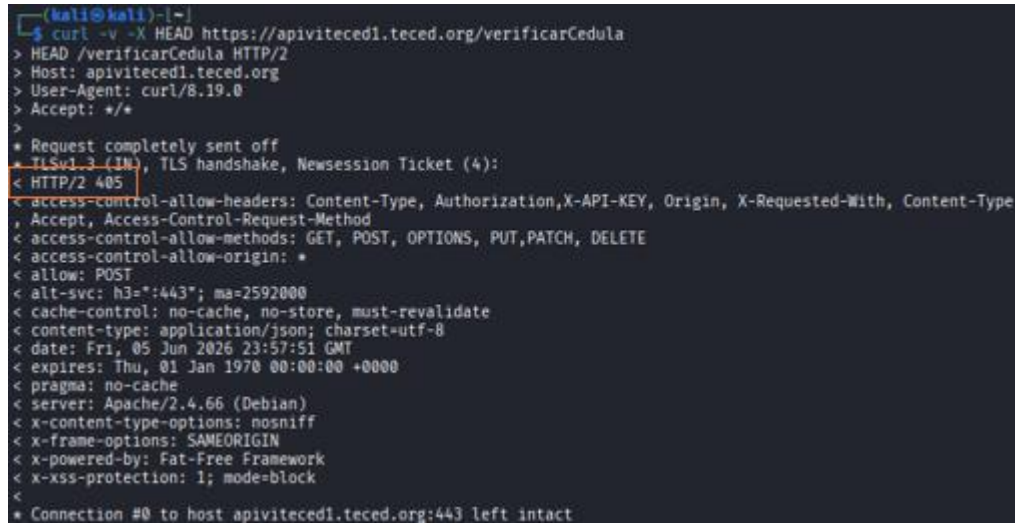
Finalmente, la Tabla 43 los resultados muestran que ninguno de los métodos probados fue procesado como GET y no se evidenció un comportamiento inesperado asociado a la aceptación de métodos HTTP arbitrarios. Por tanto, no se identificó una exposición de

seguridad relacionada con el procesamiento de métodos HTTP no reconocidos en los componentes evaluados.

Control de Acceso mediante el Método HEAD

Se realizaron solicitudes con el método HEAD sobre recursos protegidos para verificar si es posible acceder a información restringida sin autenticación:

`curl -v -X HEAD https://apiviteced1.teced.org/verificarCedula`



```
(kali@kali)~$ curl -v -X HEAD https://apiviteced1.teced.org/verificarCedula
> HEAD /verificarCedula HTTP/2
> Host: apiviteced1.teced.org
> User-Agent: curl/8.19.0
> Accept: */*
>
* Request completely sent off
* TLSv1.3 (IN), TLS handshake, Newsession Ticket (4):
< HTTP/2 405
< access-control-allow-headers: Content-Type, Authorization,X-API-KEY, Origin, X-Requested-With, Content-Type
, Accept, Access-Control-Request-Method
< access-control-allow-methods: GET, POST, OPTIONS, PUT,PATCH, DELETE
< access-control-allow-origin: *
< allow: POST
< alt-svc: h3=":443"; ma=2592000
< cache-control: no-cache, no-store, must-revalidate
< content-type: application/json; charset=utf-8
< date: Fri, 05 Jun 2026 23:57:51 GMT
< expires: Thu, 01 Jan 1970 00:00:00 +0000
< pragma: no-cache
< server: Apache/2.4.66 (Debian)
< x-content-type-options: nosniff
< x-frame-options: SAMEORIGIN
< x-powered-by: Fat-Free Framework
< x-xss-protection: 1; mode=block
<
* Connection #0 to host apiviteced1.teced.org:443 left intact
```

Figura 17 Análisis del método HEAD en la API REST – Curl.

Fuente: Elaboración propia

Resultados obtenidos sobre el método HEAD

Endpoint Evaluado	Método	Código de Respuesta	¿Bypass de Autenticación?
/verificarCedula (API)	HEAD	405 Method Not Allowed	No

Tabla 44 Resultado de Control de Acceso mediante el Método HEAD

La evaluación del método HTTP HEAD se realizó sobre el endpoint /verificarCedula de la API, con el propósito de determinar si este método podía ser utilizado para eludir los mecanismos de autenticación establecidos por el sistema. Como se muestra en la Tabla 44, la solicitud realizada mediante el método HEAD obtuvo una respuesta HTTP 405 Method Not Allowed, lo que indica que dicho método no se encuentra permitido para el endpoint evaluado. En consecuencia, no se evidenció un comportamiento que permitiera acceder al recurso o eludir los controles de autenticación mediante el uso del método HEAD.

Conclusión de la prueba

La evaluación realizada sobre los métodos HTTP permitió comprobar que los servicios analizados mantienen restricciones frente a métodos potencialmente peligrosos y métodos arbitrarios. Asimismo, el método HEAD no evidenció posibilidades de eludir los controles de autenticación. En general, los resultados obtenidos muestran un comportamiento adecuado en el tratamiento de los métodos HTTP evaluados, sin identificarse vulnerabilidades relacionadas con su habilitación, procesamiento o utilización para evadir controles de acceso.

Identificador de prueba:	Prueba HTTP Strict Transport Security
Objetivo:	Verificar la existencia del encabezado HSTS (HTTP Strict Transport Security), el cual corresponde a un mecanismo de seguridad web diseñado para proteger sitios y aplicaciones frente a distintos ataques informáticos, como la degradación de protocolo y el robo de cookies.
Modalidad:	Caja negra
Entornos Evaluados:	Frontend (https://viteced.teced.org), Backend de API REST (https://apiviteced1.teced.org), API (https://apimongo1.teced.org) y Broker MQTT (https://broker.teced.org)
Herramientas Utilizadas:	Curl

Tabla 45 Prueba HTTP Strict Transport Security

Resultados

Se verifico la presencia del encabezado HTTP Strict Transport Security (HSTS) mediante el análisis de los encabezados HTTP, tal como se muestra a continuación:

Verificación mediante Curl

Comando ejecutado:

Frontend: curl -s -D- <https://viteced.teced.org/> | grep -i "Strict-Transport-Security"

API Backend: curl -s -D- <https://apiviteced1.teced.org/> | grep -i "Strict-Transport-Security"

Broker: curl -s -D- <https://broker.teced.org/> | grep -i "Strict-Transport-Security"

Descripción del comando:

- Curl: Realiza una petición HTTP al sitio web.
- -s: Oculta la barra de progreso y mensajes innecesarios.
- -D-: Muestra los encabezados (headers) de la respuesta HTTP en la salida estándar.
- https://dominio/: URL a la que se realiza la petición.
- |: Envía la salida del comando anterior al siguiente comando.
- grep -i "Strict-Transport-Security": Busca, sin distinguir mayúsculas y minúsculas, la cadena Strict-Transport-Security dentro de los encabezados recibidos.

El análisis de los encabezados HTTP determinó que la aplicación web no implementa el mecanismo HTTP Strict Transport Security (HSTS). La ausencia de este encabezado constituye una debilidad de configuración, ya que el navegador no recibe instrucciones para recordar que el sitio debe ser accedido únicamente mediante conexiones HTTPS.

Como consecuencia, la primera conexión realizada por un usuario podría ser susceptible a ataques de degradación del protocolo (SSL Stripping), en los cuales un atacante busca impedir el establecimiento de una conexión cifrada entre el cliente y el servidor, pudiendo interceptar el tráfico en texto claro antes de que se aplique el cifrado. Asimismo, no se identificó la configuración de las directivas max-age e includeSubDomains, las cuales fortalecen la política HSTS al definir el tiempo de aplicación de la política y extenderla a los subdominios asociados al dominio principal.

Si bien la plataforma ViTECED utiliza HTTPS para proteger las comunicaciones, la implementación de HSTS representa una medida adicional de seguridad recomendada para reforzar la integridad del canal de comunicación y prevenir accesos accidentales o forzados mediante el protocolo HTTP sin cifrado.

Hallazgo Identificado

HALLAZGO VULN-002: Ausencia del Encabezado HTTP Strict Transport Security (HSTS): La aplicación no envía el encabezado Strict-Transport-Security en sus respuestas HTTP, lo que impide que los navegadores apliquen de forma automática el uso exclusivo de HTTPS en comunicaciones futuras.

Exposición a ataques de degradación de protocolo (SSL Stripping) y ataques de intermediario (Man-in-the-Middle) durante la primera conexión del usuario.

- **Categoría de prueba: Prueba de administración de identidad**

Identificador de prueba: Prueba de definición de Roles

Ejecución de la prueba:

Identificación de Roles en el Sistema

Se revisó la documentación oficial de la plataforma, correspondiente al Manual de usuario ViTECED v1 y v2, así como el artículo científico Sistema de Monitoreo de parámetros cardiovasculares basado en IoT y MQTT para alertas médicas [1]. Estas fuentes permitieron identificar los roles de usuario definidos en el sistema, sus responsabilidades y los permisos asociados. Adicionalmente, se realizó una revisión funcional de la plataforma para verificar los perfiles existentes y las funcionalidades disponibles para cada rol.

A partir de esta revisión, en la Tabla 46 se presentan los roles identificados en el sistema ViTECED y su alcance operativo principal.

Rol definido	Alcance operativo principal
Super Administrador	Gestión técnica global, administración de infraestructura, auditoría y generación de tokens de acceso.
Administrador	Gestión operativa del sistema, administración de usuarios, centros hospitalarios, especialidades y reportes.
Médico	Gestión clínica, monitoreo biomédico en tiempo real de pacientes asignados, emisión de recetas y telemedicina.
Asistente	Apoyo operativo al médico asignado y actualización de datos físicos de pacientes.
Paciente	Está orientado al monitoreo y gestión de su información personal de salud.

Tabla 46 Roles identificados del sistema

Posteriormente, se verificó la existencia de cada uno de los perfiles mediante sesiones autenticadas independientes en la plataforma. Las interfaces correspondientes a los roles identificados se presentan a continuación:

Rol Administrador



Figura 18 Interfaz del perfil Administrador.

Fuente: Elaboración propia

Rol Super Administrador



Figura 19 Interfaz del perfil Super Administrador.

Fuente: Elaboración propia

Rol Paciente

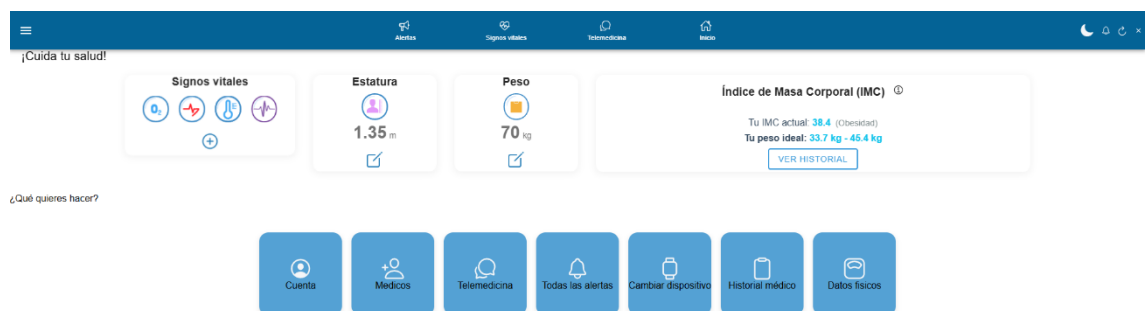


Figura 20 Interfaz del perfil Paciente.

Fuente: Elaboración propia

Rol Medico



Figura 21 Interfaz del perfil Medico.

Fuente: Elaboración propia

Asistente

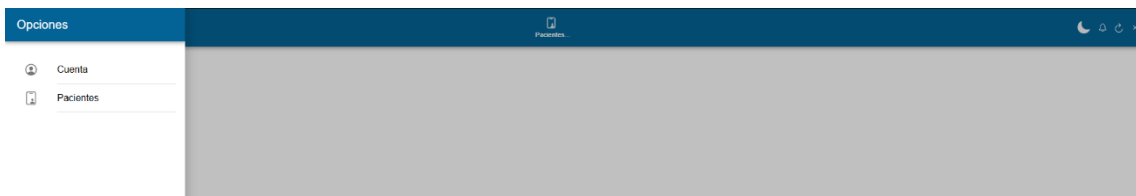


Figura 22 Interfaz del perfil Asistente.

Fuente: Elaboración propia

Como se muestra en las Figuras 19 a 23, se identificaron cinco roles principales dentro del sistema ViTECED: Paciente, Administrador, Médico, Asistente y Super Administrador. La revisión realizada no evidenció perfiles adicionales con acceso a la plataforma que no estuvieran contemplados en la documentación revisada. La existencia de estos cinco perfiles permite establecer una separación inicial de responsabilidades y constituye la base para la posterior verificación de la asignación y aplicación de permisos correspondientes a cada rol.

Matriz de roles y permisos

Con base en la revisión funcional y la documentación oficial, se consolidó en la Tabla 46 la matriz de asignación de funcionalidades y permisos correspondiente a cada perfil de usuario.

Funcionalidad / Permiso Asignado	Super Admini strador	Admin istrad or	Médico	Asistente	Paciente
Configurar perfil personal	✓	✓	✓	✓	✓

Funcionalidad / Permiso Asignado	Super Administrador	Administrador	Médico	Asistente	Paciente
Crear, editar, habilitar e inhabilitar usuarios de todos los roles	✓	✓			
Gestionar centros hospitalarios y especialidades	✓	✓			
Gestionar dispositivos IoT (crear, listar, editar)	✓	✓			
Ver reportes estadísticos de dispositivos, médicos, pacientes e investigaciones exportables (PDF/Excel)	✓	✓			
Crear parámetros biomédicos personalizados	✓	✓			
Modificar información de contacto institucional	✓	✓			
Gestionar / aprobar investigaciones científicas	✓	✓			
Consultar logs e historial del sistema	✓	✓			
Ver y gestionar sugerencias de usuarios	✓	✓			
Ver registros de auditoría y exportar (PDF/Excel)	✓				
Generar tokens de acceso (usuarios nuevos/existentes)	✓				
Consultar lista de pacientes asignados			✓	✓	
Visualizar datos biomédicos en tiempo real			✓		✓ (Propios)
Consultar alertas generadas y agregar notas					✓ (Propias)
Consultar alertas generadas de sus pacientes			✓		✓
Acceder a históricos y reportes clínicos de pacientes			✓		

Funcionalidad / Permiso Asignado	Super Administrador	Administrador	Médico	Asistente	Paciente
Consultar estadísticas de pacientes y consultas.			✓		
Generar recetas médicas y registrar consultas			✓		
Canales de telemedicina (Chat, WhatsApp, Telegram)			✓		✓
Gestionar horarios de atención			✓		
Gestionar un asistente			✓		
Editar datos físicos de pacientes (peso, talla, IMC)				✓	✓ (Propios)
Seleccionar y contactar médicos por especialidad					✓
Habilitar / deshabilitar médicos asignados					✓
Registrar y cambiar dispositivo IoT propio					✓
Consultar recetas médicas e historial de antecedentes personales.					✓

Tabla 47 Matriz de roles y permisos

La comparación entre la matriz elaborada y las funcionalidades visibles en la interfaz permitió verificar que la asignación de permisos mantiene coherencia con los procesos soportados por la plataforma. Los privilegios administrativos permanecen asociados a los usuarios con mayor nivel de acceso (Administrador y Super Administrador), mientras que los perfiles clínicos (Médico, Asistente) y el perfil Paciente poseen accesos limitados y acordes a su función dentro del proceso de telemedicina.

Validación de la matriz contra las políticas de acceso

Se iniciaron sesiones autenticadas e independientes para cada uno de los cinco roles definidos en la plataforma. Para cada sesión se verificó la disponibilidad de las funcionalidades en la interfaz y su correcto funcionamiento, contrastando los resultados

obtenidos con los permisos establecidos en la matriz teórica de roles, presentada en las Tabla 47 a 51. Esta comparación permitió determinar si los accesos observados para cada perfil corresponden con las funcionalidades definidas en la documentación oficial.

Sesión Super Administrador

Funcionalidad definida en el manual	¿Disponible en la interfaz?	¿Funciona?	Coincide
Ver y gestionar sugerencias de usuarios	Si	Si	Si
Ver registro de auditorías y exportar en PDF y Excel	Si	Si	Si
Configurar su perfil personal	Si	Si	Si
Crear, editar, habilitar e inhabilitar usuarios de todos los roles	Si	Si	Si
Ver informes de reportes estadísticos de dispositivos, médicos, pacientes e investigaciones y exportarlo en PDF y Excel	Si	Si	Si
Crear parámetros biomédicos personalizados	Si	Si	Si
Cambiar información de contacto institucional	Si	Si	Si
Ver investigaciones científicas	Si	Si	Si
Gestionar dispositivos IoT (crear, listar, editar)	Si	Si	Si
Consultar logs del sistema y gestión de historial de logs	Si	Si	Si
Gestionar centros hospitalarios (Crear, Editar, Eliminar)	Si	Si	Si
Gestionar especialidades médicas (Crear, Editar, Eliminar)	Si	Si	Si
Generar tokens de acceso para usuarios nuevos y existentes	Si	Si	Si

Tabla 48 Verificación de funcionalidades del perfil Super Administrador

Sesión Administrador

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Ver y gestionar sugerencias de usuarios	Si	Si	Si
Configurar su perfil personal	Si	Si	Si

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Crear, editar, habilitar e inhabilitar usuarios de todos los roles	Si	Si	Si
Ver informes de reportes estadísticos de dispositivos, médicos, pacientes e investigaciones y exportarlo en PDF y Excel	Si	Si	Si
Crear parámetros biomédicos personalizados	Si	Si	Si
Cambiar información de contacto institucional	Si	Si	Si
Aprobar investigaciones científicas	Si	Si	Si
Gestionar dispositivos IoT (crear, listar, editar)	Si	Si	Si
Consultar logs del sistema y gestión de historial de logs	Si	Si	Si
Gestionar centros hospitalarios (Crear, Editar, Eliminar)	Si	Si	Si
Gestionar especialidades médicas (Crear, Editar, Eliminar)	Si	Si	Si

Tabla 49 Verificación de funcionalidades del perfil Administrador

Sesión Médico

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Ver lista de pacientes asignados	Si	Si	Si
Visualizar datos biomédicos (frecuencia cardíaca, temperatura, electrocardiograma y saturación de oxígeno) en tiempo real de sus pacientes.	Si	Si	Si
Ver todas las alertas de sus pacientes	Si	Si	Si
Acceder a históricos clínicos de sus pacientes	Si	Si	Si
Generar reportes clínicos	Si	Si	Si
Comunicarse con pacientes vía chat directo, WhatsApp y Telegram	Si	Si	Si
Registrar consultas médicas	Si	Si	Si
Generar recetas	Si	Si	Si
Gestionar horarios de atención	Si	Si	Si

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Ver estadísticas de pacientes y consultas	Si	Si	Si
Agregar un nuevo asistente	Si	Si	Si
Editar los datos del asistente	Si	Si	Si
Configurar su perfil personal	Si	Si	Si

Tabla 50 Verificación de funcionalidades del perfil Medico

Sesión asistente

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Visualizar el listado de los pacientes vinculados al médico asignado	Si	Si	Si
Editar datos físicos de pacientes	Si	Si	Si
Configurar su perfil personal	Si	Si	Si

Tabla 51 Verificación de funcionalidades del perfil Asistente

Sesión paciente

Funcionalidad	¿Disponible en la interfaz?	¿Funciona?	Coincide
Visualizar en tiempo real sus datos biomédicos (frecuencia cardíaca, temperatura, saturación de oxígeno y electrocardiograma)	Si	Si	Si
Ver alertas generadas y agregar notas a las mismas	Si	Si	Si
Seleccionar y contactar médicos según su especialidad	Si	Si	Si
Habilitar o deshabilitar medico asignados	No	No	No
Usar telemedicina (WhatsApp, Telegram, chat directo)	Si	Si	Si
Ver recetas médicas e historial de antecedentes personales	Si	Si	Si
Registrar y cambiar su dispositivo IoT	Si	Si	Si
Ver, registrar y editar sus datos físicos (peso, talla, IMC)	Si	Si	Si
Configurar su perfil personal	Si	Si	Si

Tabla 52 Verificación de funcionalidades del perfil Paciente

Observación: En el rol Paciente, la opción "Habilitar o deshabilitar médicos asignados" no se encuentra presente en la interfaz, por ende, no pudo ser validada operativamente durante las pruebas, constituyendo una observación menor de verificación.

Identificador de prueba: Prueba de enumeración y adivinanzas de cuentas de usuario

Ejecución de la prueba:

Mensaje de respuesta HTTP

Se evaluó el mecanismo de autenticación de ViTECED para determinar si las respuestas generadas ante diferentes combinaciones de credenciales permitían identificar la existencia de una cuenta registrada. Para ello, se analizaron tres escenarios de autenticación.

Escenario 1: Usuario válido con contraseña incorrecta

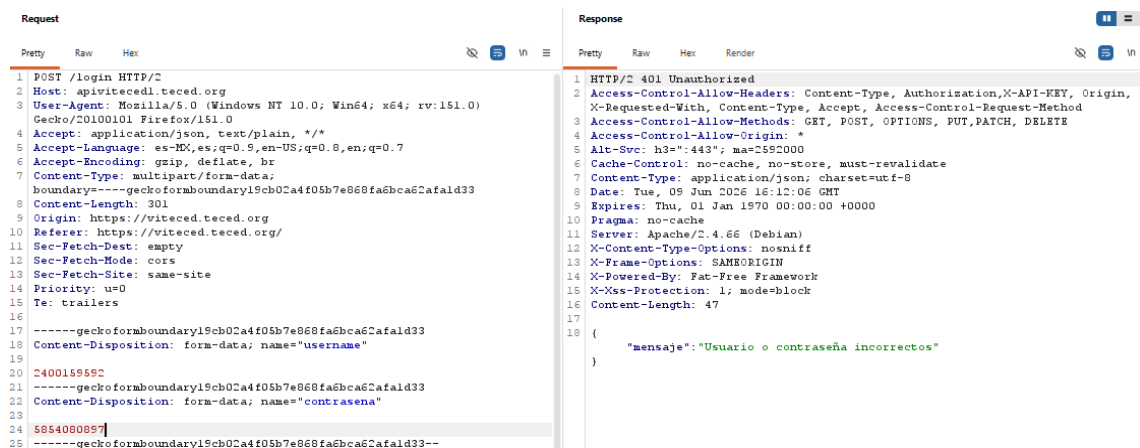


Figura 23 Intento de inicio de sesión con Usuario válido y contraseña incorrecta.

Fuente: Elaboración propia

En la Figura 24 se muestra la solicitud de autenticación realizada utilizando un usuario existente en la plataforma, pero proporcionando una contraseña incorrecta. La aplicación rechazó la autenticación y devolvió una respuesta HTTP 401 Unauthorized. El mensaje mostrado fue genérico y no proporcionó información que permitiera determinar si el usuario ingresado se encontraba registrado en el sistema.

Escenario 3: Usuario inexistente con contraseña incorrecta

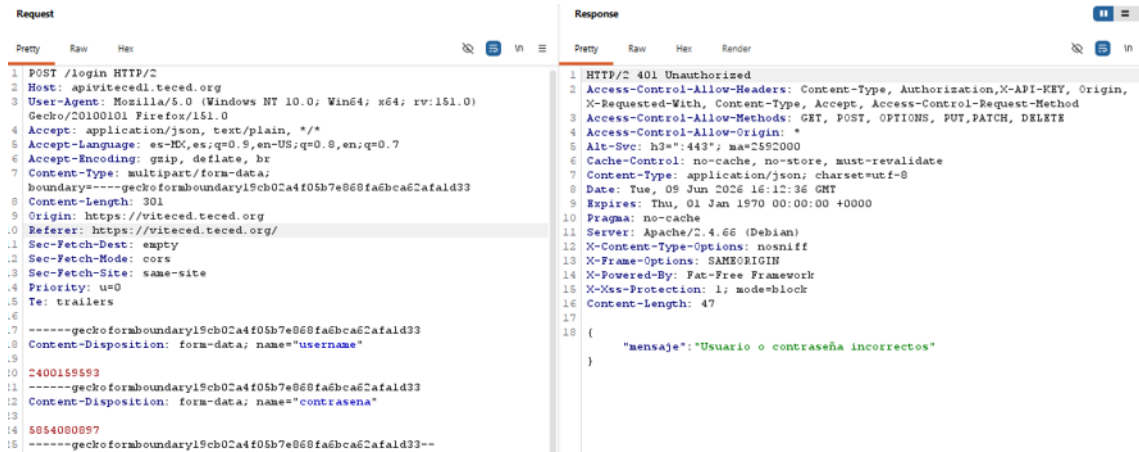


Figura 24 Intento de inicio de sesión con credenciales invalidas.

Fuente: Elaboración propia

En la Figura 25, se muestra la solicitud de autenticación realizada utilizando un usuario que no se encuentra registrado en la plataforma y una contraseña incorrecta. La aplicación rechazó la autenticación mediante una respuesta HTTP 401 Unauthorized. La respuesta obtenida mantuvo el mismo comportamiento observado en el escenario anterior, utilizando el mismo mensaje genérico y presentando una longitud equivalente, sin revelar información que permitiera determinar la existencia de la cuenta.

Mensaje de respuesta HTTP

Escenario	Código HTTP	Mensaje del body	Tamaño (bytes)
Credenciales válidas	200 OK	{"mensaje": "Autenticación exitosa", "token": "eyJ....."}	1616 bytes
Usuario válido, contraseña incorrecta	401 Unauthorized	{"mensaje": "Usuario o contraseña incorrectos"}	47 bytes
Usuario inexistente, contraseña incorrecta	401 Unauthorized	{"mensaje": "Usuario o contraseña incorrectos"}	47 bytes

Tabla 53 Comparación de respuestas del mecanismo de autenticación ante diferentes escenarios de credenciales

Tras el análisis de la comparación entre los escenarios de autenticación fallida como se visualiza en la Tabla 53 demuestra que la aplicación mantiene un comportamiento uniforme cuando las credenciales son incorrectas. Tanto para un usuario existente con contraseña inválida como para un usuario inexistente, el servidor devuelve el mismo código de estado (401 Unauthorized), el mismo mensaje de error y una respuesta con la misma longitud. Este comportamiento impide que un atacante pueda distinguir, únicamente mediante el análisis de la respuesta HTTP, si un nombre de usuario se encuentra registrado en el sistema, reduciendo el riesgo de enumeración de cuentas por diferencias en las respuestas del mecanismo de autenticación.

Analizando el código de error recibido en las páginas de inicio de sesión

Usuario inexistente y contraseña incorrecta



Figura 25 Análisis del mensaje de error tras inicio de sesión con credenciales invalidas.

Fuente: Elaboración propia

Durante la evaluación de pruebas utilizando dos escenarios diferentes como se muestra en la Figura 26, se verificó que la aplicación mantiene el mismo comportamiento visual cuando un usuario existente ingresa una contraseña incorrecta y cuando se utiliza un usuario inexistente. En ambos escenarios la aplicación presentó el mensaje genérico "Error al autenticarse", sin indicar si el nombre de usuario existe o si el error corresponde únicamente a la contraseña.

La ausencia de mensajes diferenciados impide que un atacante pueda identificar la existencia de cuentas válidas únicamente mediante la observación de los mensajes

mostrados durante el proceso de autenticación, por lo que no se evidenció enumeración de usuarios a través de la interfaz de inicio de sesión.

Análisis de los títulos de la Página Web

Usuario válido y contraseña incorrecta

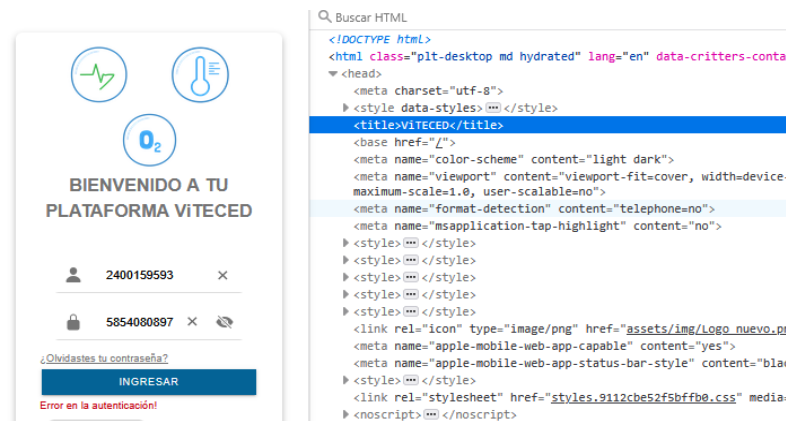


Figura 26 Título de la página durante la autenticación con usuario válido y contraseña incorrecta.

Fuente: Elaboración propia

Durante la evaluación se analizó el comportamiento del título de la página en dos escenarios de autenticación fallida: un usuario existente con contraseña incorrecta y un usuario inexistente con contraseña incorrecta. Como se muestra en la Figura 27, se comprobó que el título de la página permanece constante como “ViTECED” en ambos escenarios.

La aplicación no utiliza títulos diferenciados para indicar la existencia del usuario ni la causa del fallo de autenticación. Por tanto, no se identificó una vulnerabilidad de enumeración de usuarios mediante el análisis del título de la página web, debido a que este elemento no proporciona información que permita determinar si el usuario ingresado se encuentra registrado en el sistema.

Análisis de un mensaje recibido de una instalación de recuperación

Se realizaron dos solicitudes al módulo de recuperación de contraseña utilizando una dirección de correo electrónico registrada en el sistema y una dirección de correo electrónico inexistente. Como se muestra en las Figuras 28 a 30, se observó un comportamiento diferenciado entre ambos escenarios. Para el correo electrónico registrado, la aplicación mostró en el frontend el mensaje “Código de recuperación

enviado”, acompañado de una respuesta HTTP 200 OK y del mensaje del servidor “Correo enviado”.

En cambio, al utilizar un correo electrónico inexistente, el frontend presentó el mensaje “Hubo un problema al enviar el correo. Intenta nuevamente más tarde”. Mediante la inspección de la comunicación con el servidor a través de Burp Suite, se obtuvo una respuesta de HTTP 404 Not Found indicando explícitamente con un mensaje “Correo no encontrado”.

La comparación de ambos escenarios evidencia que el servidor genera respuestas diferenciadas dependiendo de la existencia de la cuenta. Esta condición permite determinar si una dirección de correo electrónico se encuentra registrada en la plataforma mediante el módulo de recuperación de contraseña, constituyendo un mecanismo de enumeración de cuentas de usuario.

Correo inexistente: cinthiangel2@gmail.com

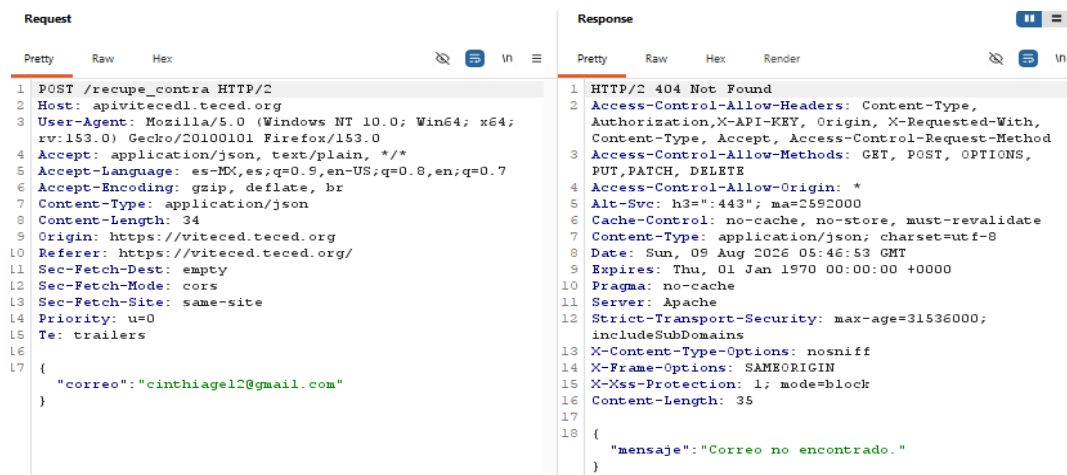


Figura 27 Verificación de cuenta mediante la respuesta del servidor ante un correo inexistente - Burp Suite.

Fuente: Elaboración propia

Correo registrado: cinthiangel2@gmail.com

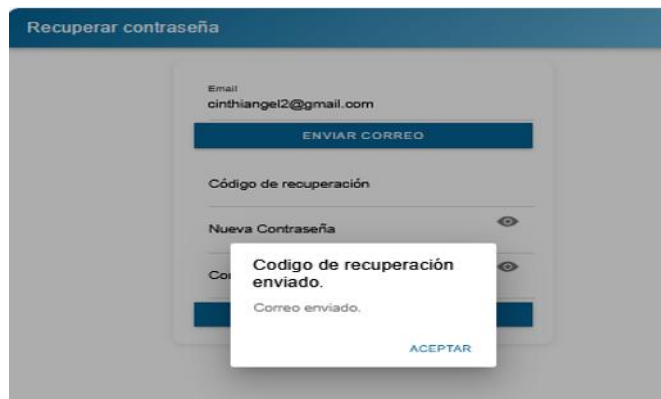


Figura 28 Verificación de cuenta mediante correo electrónico existente.

Fuente: Elaboración propia

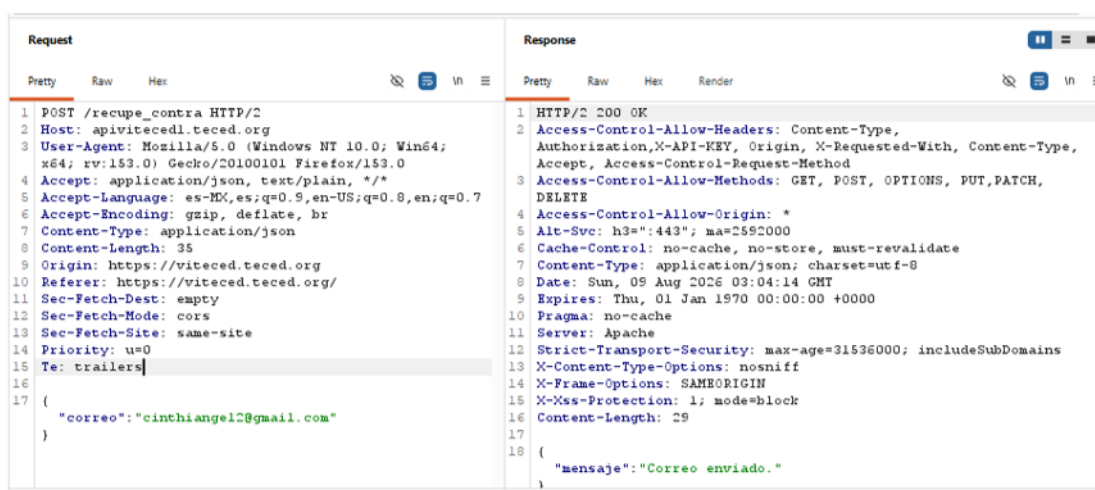


Figura 29 Verificación de cuenta mediante la respuesta del servidor ante un correo existente- Burp Suite.

Fuente: Elaboración propia

Adivinanza de usuarios

Durante la evaluación se identificó que la plataforma ViTECED utiliza el número de cédula de identidad ecuatoriana como identificador de usuario para el proceso de autenticación. Debido a que este identificador se basa en un formato conocido y público, es posible generar posibles números de cédula que podrían utilizarse en intentos de autenticación dirigidos al endpoint POST /login. El uso del número de cédula de identidad como identificador de usuario no constituye por sí mismo una vulnerabilidad de seguridad, debido a que corresponde a una decisión de diseño de la aplicación. No obstante, al tratarse de un identificador basado en un formato conocido y público, puede facilitar la generación de posibles identificadores para realizar intentos de autenticación.

La condición de seguridad se produciría si la aplicación proporcionara respuestas diferentes al utilizar una cédula registrada frente a una inexistente, permitiendo confirmar la existencia de una cuenta. Sin embargo, durante la evaluación se comprobó que el mecanismo de autenticación utiliza un mensaje genérico de error: “Usuario o contraseña incorrecta”. Este comportamiento impide diferenciar, a partir del mensaje mostrado, entre una cuenta existente con una contraseña incorrecta y una cédula que no se encuentra registrada.

En consecuencia, con la evidencia obtenida no se evidenció un mecanismo de enumeración de cuentas mediante el proceso de autenticación.

Modalidad: Caja Gris

Resultados

Mensaje de respuesta HTTP

Backend: controladores/User_Ctrl.php

```
// Generar el JWT
$jwt = JWT::encode($payload, $key, 'HS256');
// Construir la respuesta JSON
// Enviar el token en la respuesta JSON
echo json_encode([
    'mensaje' => 'Autenticación exitosa',
    'token' => $jwt,
    'id_rol' => $this->M_Modelo->id_rol,
    'nombre' => $this->M_Modelo->nombre,
    'id_usuario' => $this->M_Modelo->id_usuario,
    'menu' => $menu,
    'peso' => $peso,
    'topic' => $dispo
]);
} elseif ($this->M_Modelo->estado == 'B') {
    echo json_encode([
        'mensaje' => 'Usuario Bloqueado',
    ]);
} else {
    echo json_encode([
        'mensaje' => 'Usuario inactivo',
    ]);
} else {
    echo json_encode([
        'mensaje' => 'Usuario o contraseña incorrectos',
    ]);
} else {
    echo json_encode([
        'mensaje' => 'Usuario o contraseña incorrectos',
    ]);
}
```

Figura 30 Código backend para la emisión de tokens de sesión y manejo de mensajes de respuesta según el estado del usuario.

Fuente: Elaboración propia

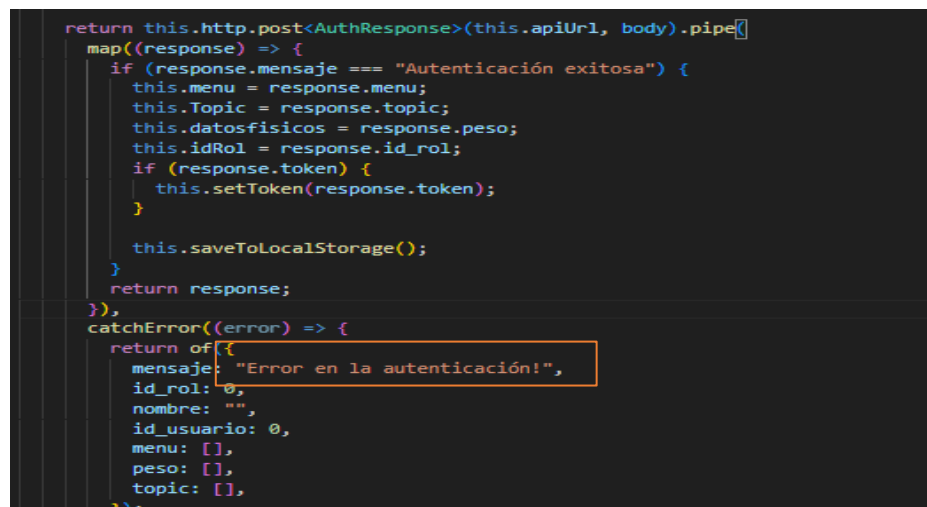
Se revisaron el controlador User_Ctrl.php y el componente autenticacion.page.ts, identificándose que el mecanismo de autenticación utiliza un mensaje genérico cuando las credenciales proporcionadas son incorrectas. Asimismo, en el código fuente se

identificaron respuestas específicas asociadas a determinados estados de la cuenta, como usuarios bloqueados o inactivos.

Estas respuestas diferenciadas podrían proporcionar información sobre el estado de una cuenta si fueran accesibles mediante la interfaz o directamente a través de la API. Sin embargo, durante las pruebas realizadas no se evidenció que dichos mensajes fueran expuestos ante intentos de autenticación con usuarios existentes o inexistentes.

Analizando el código de error recibido en las páginas de inicio de sesión

Backend -> ser-autenticacion.service.ts



```
return this.http.post<AuthResponse>(this.apiUrl, body).pipe(
  map((response) => {
    if (response.mensaje === "Autenticación exitosa") {
      this.menu = response.menu;
      this.topic = response.topic;
      this.datosfisicos = response.peso;
      this.idRol = response.id_rol;
      if (response.token) {
        this.setToken(response.token);
      }
      this.saveToLocalStorage();
    }
    return response;
  }),
  catchError((error) => {
    return of({
      mensaje: "Error en la autenticación!",
      id_rol: 0,
      nombre: "",
      id_usuario: 0,
      menu: [],
      peso: [],
      topic: [],
    });
  })
);
```

Figura 31 Análisis integral de la gestión de respuestas y mensajes de error en el proceso de autenticación de la plataforma.

Fuente: Elaboración propia

En la Figura 32, se muestra la revisión del método authenticate() confirmó que el frontend muestra un mensaje genérico ("Error en la autenticación") para los distintos escenarios de autenticación fallida, comportamiento consistente con las pruebas realizadas durante la evaluación de caja negra.

Análisis de un mensaje recibido de una instalación de recuperación

Frontend -> recuperar-contrasena.page.ts

```

async enviarCorreo() {
  const { correo } = this.emailForm.value;

  try {
    const response = await this.usuarioService.Recuperar_contraseña({ correo }).toPromise();

    if (response.mensaje === 'Correo enviado.') {

      this.isResetVisible = true;
      this.resetForm.patchValue({ correo });
      this.mostrarAlerta('Codigo de recuperación enviado.', response.mensaje);
    } else {

      this.isResetVisible = false;
      this.mostrarAlerta('Error', response.mensaje || 'No se pudo enviar el correo, intenta nuevamente.');
```

Figura 32 Análisis de un mensaje recibido de una instalación de recuperación en el código fuente.

Fuente: Elaboración propia

La revisión del método logout() confirmó que, tras el cierre de sesión o ante una condición de autenticación inválida, como se muestra en la Figura 33, la aplicación mantiene al usuario en la ruta de autenticación, sin generar redireccionamientos diferenciados que permitan inferir información sobre el estado de la cuenta o de la autenticación.

Análisis de los títulos de la Página Web

Frontend -> src/index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>ViTECED</title>

  <base href="/" />

  <meta name="color-scheme" content="light dark" />
  <meta name="viewport" content="viewport-fit=cover, width=device-width, initial-scale=1.0, mini
  <meta name="format-detection" content="telephone=no" />
  <meta name="msapplication-tap-highlight" content="no" />

  <link rel="icon" type="image/png" href="assets/img/Logo nuevo.png" />

```

Figura 33 Análisis del título de la Página Web en el código fuente.

Fuente: Elaboración propia

Como se muestra en la Figura 34 el título de la aplicación está definido de forma estática como "ViTECED" en el archivo index.html y no existe ninguna lógica en el código que lo modifique dinámicamente según el resultado de la autenticación, confirmando que el título permanece invariable en todos los escenarios posibles.

Hallazgo Identificado

HALLAZGO VULN-003: Enumeración de Cuentas de Usuario mediante el Módulo de Recuperación de Contraseña

Descripción: La aplicación muestra un mensaje afirmativo de envío (200 OK) cuando el correo existe y un mensaje de error cuando no existe, permitiendo confirmar correos registrados.

Componente Afectado: Módulo de Recuperación de Contraseña (recuperar-contrasena.page.ts / API de recuperación)

Impacto Potencial: Permite a un atacante construir una lista verificada de usuarios registrados para orientar ataques posteriores de fuerza bruta o phishing.

- **Categoría de prueba: Pruebas de Autenticación**

Identificador de prueba: Prueba de transporte de credenciales en canal encriptado

Ejecución de la prueba:

Resultados

Se evaluó la seguridad del transporte de credenciales mediante cuatro escenarios específicos de interceptación de tráfico:

Ejemplo 1: Envío de datos con el método POST a través de HTTP

Se intentó acceder directamente al endpoint de autenticación mediante el esquema no cifrado: <http://apivitecedl.teced.org/login>.

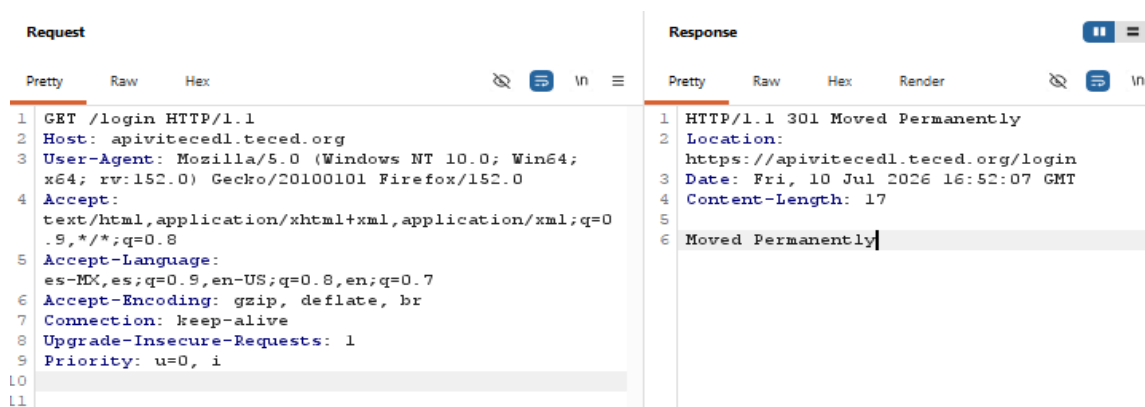


Figura 34 Envío de datos con el método POST a través de HTTP.

Fuente: Elaboración propia

El servidor respondió con un código de estado 301 Moved Permanently, como se muestra en la Figura 35 indica una redirección forzosa hacia la versión cifrada del recurso

(https://apiviteced1.teced.org/login). Este comportamiento indica que el servidor rechaza establecer comunicación en HTTP puro y obliga a la conexión a completarse mediante HTTPS antes de procesar cualquier dato de autenticación.

Ejemplo 2: Envío de datos con el método POST a través de HTTPS



```

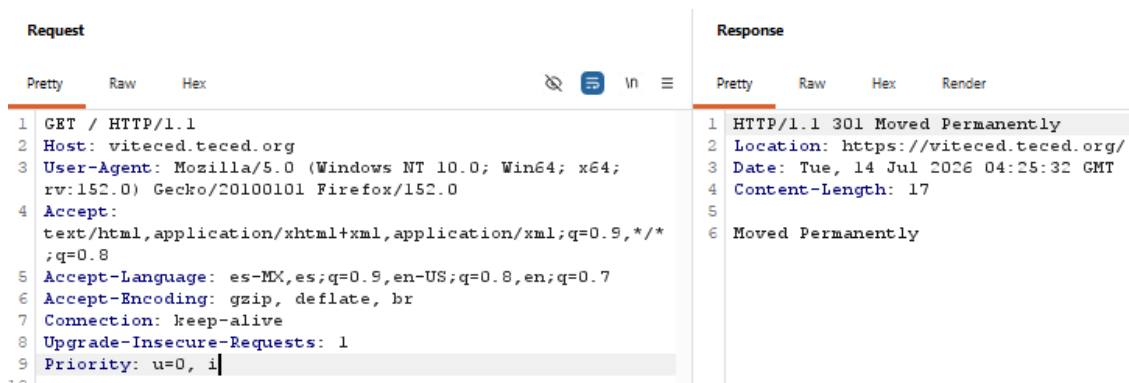
Request
Pretty Raw Hex
1 POST /login HTTP/2
2 Host: apiviteced1.teced.org
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:152.0) Gecko/20100101 Firefox/152.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: es-MX, es; q=0.9, en-US; q=0.8, en; q=0.7
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: multipart/form-data; boundary=----geckoformboundarybca93e61ba6c69193be6219764a7eaeef
8 Content-Length: 301
9 Origin: https://viteced.teced.org
10 Referer: https://viteced.teced.org/
11 Sec-Fetch-Dest: empty
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Site: same-site
14 Priority: u=0
15 Te: trailers
16 Connection: keep-alive
17
18 -----geckoformboundarybca93e61ba6c69193be6219764a7eaeef
19 Content-Disposition: form-data; name="username"
20
21 0928197532
22 -----geckoformboundarybca93e61ba6c69193be6219764a7eaeef
23 Content-Disposition: form-data; name="contrasena"
24
25 0999125283
26 -----geckoformboundarybca93e61ba6c69193be6219764a7eaeef--
27

```

Figura 35 Envío de datos con el método POST a través de HTTPS.

Fuente: Elaboración propia

Ejemplo 3: Envío de datos con el método POST a través de HTTPS en una página accesible a través de HTTP



```

Request
Pretty Raw Hex
1 GET / HTTP/1.1
2 Host: viteced.teced.org
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:152.0) Gecko/20100101 Firefox/152.0
4 Accept: text/html, application/xhtml+xml, application/xml; q=0.9, */*; q=0.8
5 Accept-Language: es-MX, es; q=0.9, en-US; q=0.8, en; q=0.7
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9 Priority: u=0, i

Response
Pretty Raw Hex Render
1 HTTP/1.1 301 Moved Permanently
2 Location: https://viteced.teced.org/
3 Date: Tue, 14 Jul 2026 04:25:32 GMT
4 Content-Length: 17
5
6 Moved Permanently

```

Figura 36 Envío de datos con el método POST a través de HTTPS en una página accesible a través de HTTP.

Fuente: Elaboración propia

En la Figura 37 se muestra la solicitud POST /login mediante HTTP/2 dirigida al host apiviteced1.teced.org. Los encabezados Origin: https://viteced.teced.org y Referer: https://viteced.teced.org/ confirman que la solicitud de autenticación se origina desde la aplicación web mediante una conexión HTTPS. El cuerpo de la petición, enviado

mediante multipart/form-data, contiene los campos username y contraseña correspondientes a los valores utilizados durante la prueba.

Aunque las credenciales pudieron visualizarse durante la interceptación, este comportamiento es consecuencia de la utilización de una herramienta de análisis que actúa como intermediario controlado y permite inspeccionar la comunicación mediante la terminación controlada de la sesión TLS. Posteriormente, la herramienta establece una nueva conexión TLS con el servidor de destino.

Por tanto, la visualización de las credenciales durante esta prueba no constituye, por sí misma, una vulnerabilidad de transmisión en texto plano. La evidencia obtenida confirma que la comunicación entre la aplicación y el servidor se realiza mediante HTTPS/TLS, manteniendo las credenciales protegidas durante su transmisión en condiciones normales de operación.

Ejemplo 4: Envío de datos con el método GET a través de HTTPS

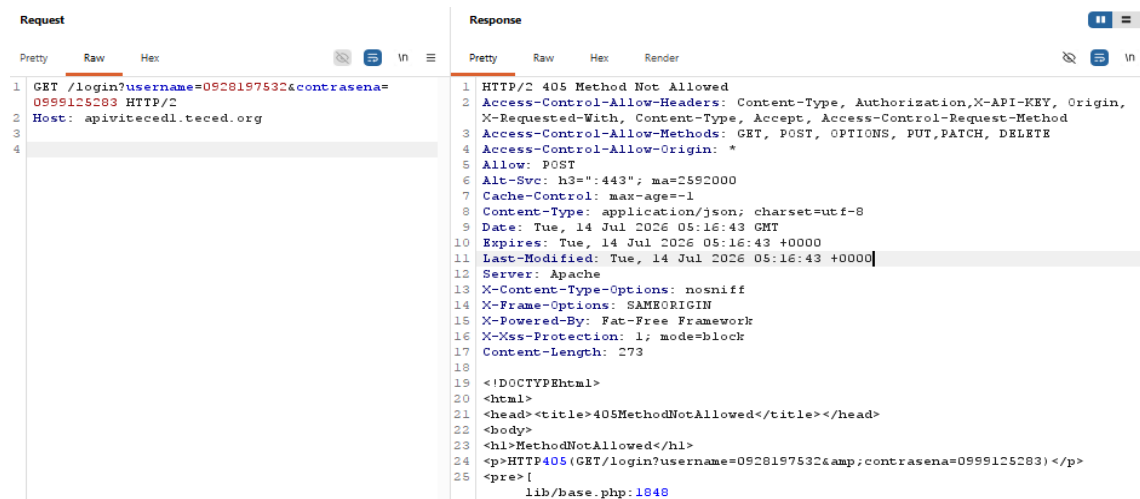


Figura 37 Envío de datos con el método GET a través de HTTPS.

Fuente: Elaboración propia

En la Figura 38, al solicitar <http://viteced.teced.org/>, el servidor respondió con el código de estado 301 Moved Permanently e incluyó el encabezado Location: <https://viteced.teced.org/>, redirigiendo automáticamente la solicitud hacia la versión protegida mediante HTTPS antes de entregar el contenido de la aplicación. Este mismo comportamiento fue identificado previamente en el subdominio de la API apiviteced1.teced.org, donde las solicitudes realizadas mediante HTTP son redirigidas hacia su equivalente en HTTPS mediante una respuesta 301 Moved Permanently.

Los resultados evidencian que tanto el frontend como el backend disponen de mecanismos de redirección de HTTP hacia HTTPS, favoreciendo el establecimiento de las comunicaciones mediante el canal cifrado.

Modalidad: Caja Gris

Resultados

A respuestas proporcionadas por el equipo de desarrollo mediante una entrevista (Ver Anexo 1), se determinó que existe conocimiento sobre las diferencias entre HTTP y HTTPS y sobre los riesgos asociados a la transmisión de información sensible mediante canales no cifrados. El equipo reconoce la necesidad de proteger las comunicaciones debido a que ViTECED gestiona credenciales, información personal y datos biomédicos.

Asimismo, se indicó que el sistema utiliza HTTPS mediante TLS para proteger las comunicaciones entre la aplicación y los servicios backend, incluyendo el intercambio de información sensible. El equipo manifestó que los servicios cuentan con certificados digitales y que se mantienen mecanismos orientados a dirigir las comunicaciones hacia el canal seguro.

En consecuencia, las respuestas obtenidas evidencian que el uso de HTTPS forma parte de las medidas de seguridad consideradas para proteger la información durante su transmisión. No obstante, las respuestas de la entrevista deben contrastarse con la verificación técnica de la configuración del servidor, certificados TLS, redirecciones HTTP-HTTPS y solicitudes observadas durante la evaluación, debido a que la entrevista permite conocer las medidas declaradas por el equipo, pero no sustituye la comprobación técnica de su correcta aplicación.

Identificador de prueba: Pruebas para buscar la debilidad de memoria caché del navegador

Ejecución de la prueba:

Historia del navegador (botón "Atrás")

Al presionar el botón "Atrás", el sistema no mostró la información sensible previamente cargada. En su lugar, redirigió al usuario a una página de "Acceso Denegado" (viteced.teced.org/no-autorizado), indicando que no se tienen permisos para acceder al recurso solicitado, tal como se muestra en la Figura 39. Este comportamiento indica que la aplicación valida la sesión activa en el servidor en cada solicitud, incluso cuando el navegador intenta recuperar el recurso desde el historial de navegación.



Figura 38 Resultado al presionar "Atrás" tras cerrar sesión - pantalla de "Acceso Denegado".

Fuente: Elaboración propia

Identificador de prueba: Prueba de credenciales por defecto

Modalidad: Caja Negra

Resultados

Se evaluó la presencia de credenciales predeterminadas, predecibles o débiles a través de los cuatro escenarios técnicos definidos en la metodología:

Credenciales por defecto de aplicaciones comunes

- **Probar nombres de usuario y contraseñas administrativas comunes y basadas en el nombre de la aplicación u organización**

Como se muestra en la Figura 40, se revisó el mecanismo de autenticación de la plataforma ViTECED para determinar la aplicabilidad de credenciales administrativas o de prueba comúnmente utilizadas, como admin, administrator, root, user, guest, operator, test y testing. Las solicitudes realizadas con estas credenciales fueron rechazadas por el servidor mediante una respuesta HTTP 401 Unauthorized, indicando que las credenciales proporcionadas no fueron aceptadas para la autenticación.

Adicionalmente, en la Figura 41 se puede observar que el mecanismo de creación de cuentas, la plataforma no permite seleccionar libremente el nombre de usuario. El identificador es asignado automáticamente por el sistema y corresponde al número de

cédula del usuario. Por esta razón, los términos administrativos o de prueba utilizados en esta comprobación no forman parte del esquema de identificación de usuarios de ViTECED. De igual manera, se verificó si el nombre de la aplicación u organización, como ViTECED o TECED, es utilizado para generar o seleccionar nombres de usuario. No se identificó esta práctica, debido a que el sistema asigna automáticamente el número de cédula como identificador de usuario.

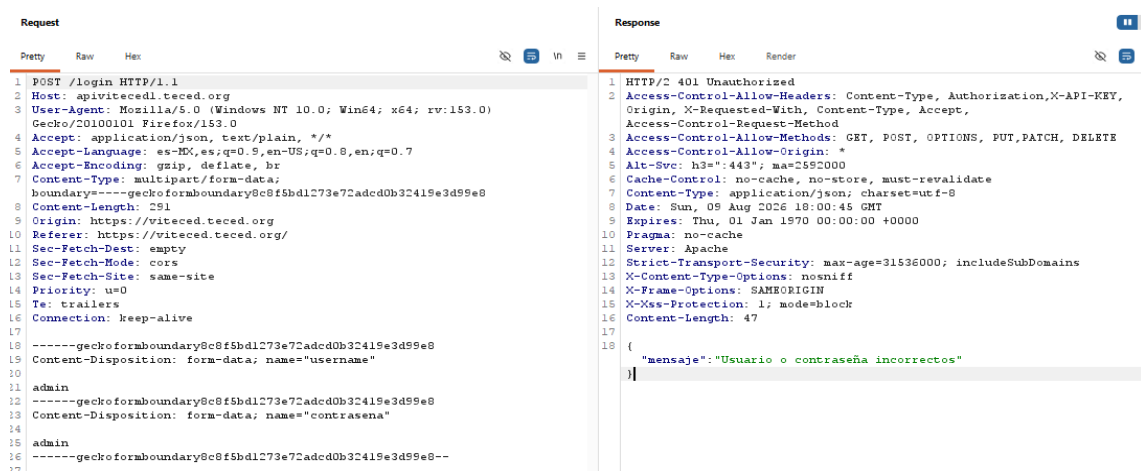


Figura 39 Verificación de nombres de usuario y contraseñas administrativas comunes.

Fuente: Elaboración propia

En consecuencia, no se identificaron cuentas con nombres de usuario administrativos, de prueba o derivados directamente del nombre de la aplicación u organización que pudieran ser utilizados como credenciales predeterminadas. La comprobación permitió determinar que este tipo de credenciales no forma parte del mecanismo de asignación de usuarios de la plataforma.

Contraseñas por defecto en cuentas nuevas

- **Analizar la página de registro para determinar el formato de credenciales**

Como se visualiza en la Figura 41, se revisó el formulario de registro de usuarios con el propósito de identificar las características y restricciones establecidas para las credenciales de acceso. Se verificó que el sistema establece una longitud mínima de 8 caracteres para la contraseña, requisito que debe cumplirse para completar el registro de una cuenta. Asimismo, durante la revisión no se identificó una restricción de longitud máxima de la contraseña en el formulario analizado. Este resultado permite establecer las características básicas de validación aplicadas por la plataforma para la creación de

credenciales y sirve como referencia para las comprobaciones posteriores relacionadas con contraseñas predecibles o triviales.

The screenshot shows a registration form titled 'Registro de nuevo Usuario'. It has two columns of input fields. The left column includes: 'Nombre *' (m), 'Apellido *' (m), a tabbed section with 'CÉDULA' selected (showing '78' and an error 'La cédula ingresada no es válida.'), 'Género *', and 'Fecha de Nacimiento *' (dd/mm/aaaa). The right column includes: 'Correo Electrónico *' (m), 'País de origen*', 'Provincia de residencia*', 'Cantón de residencia*', 'Teléfono *' (with error 'El teléfono debe tener 10 dígitos.'), and an 'Autenticación' section with 'Usuario *' (78) and 'Contraseña *' (1254, with error 'La contraseña es obligatoria y debe tener al menos 8 caracteres.'). A 'SIGUIENTE' button is at the bottom right.

Figura 40 Verificación de formato de credenciales.

Fuente: Elaboración propia

- **Determinar cómo se generan los nombres de usuario**

This screenshot shows the same registration form, but with different data. The 'Nombre *' is 'Ana' and 'Apellido *' is 'Lainez'. The 'CÉDULA' tab shows '0928197532' (highlighted with an orange box). The 'Teléfono *' field is empty. In the 'Autenticación' section, the 'Usuario *' field contains '0928197532' (highlighted with an orange box), demonstrating that the system automatically generates the username from the ID number. The 'Contraseña *' field shows masked characters. The 'SIGUIENTE' button is at the bottom.

Figura 41 Verificación del mecanismo de generación del nombre de usuario.

Fuente: Elaboración propia

De acuerdo con los resultados en la Figura 42, se verificó que el sistema genera automáticamente el nombre de usuario utilizando el número de cédula del usuario registrado. Asimismo, se constató que el usuario no tiene la posibilidad de seleccionar o modificar dicho identificador. Debido a que el formato empleado es numérico y predecible, cualquier persona que conozca el número de cédula de un usuario puede inferir directamente su nombre de usuario, lo que facilita procesos de enumeración de cuentas y ataques de adivinación de credenciales.

- **Determinar si las contraseñas generadas por el sistema son predecibles**

Mediante el análisis presentado se verificó el mecanismo utilizado por la plataforma ViTECED para la asignación de contraseñas durante el registro de nuevas cuentas. Se determinó que el sistema no genera automáticamente contraseñas iniciales o temporales para los usuarios. En su lugar, la contraseña es definida directamente por el usuario durante el proceso de creación de la cuenta, de acuerdo con las restricciones establecidas por la plataforma.

Por lo tanto, no existe una contraseña generada por el sistema que pueda ser comparada entre diferentes cuentas para identificar patrones secuenciales, fijos o derivados de información conocida. En consecuencia, no se identificó un mecanismo de generación automática de contraseñas predecibles, por lo que esta comprobación no resulta aplicable al funcionamiento actual de la plataforma.

- **Probar contraseña vacía o igual al nombre de usuario**

Es un error frecuente en aplicaciones académicas no validar que la contraseña sea diferente al usuario o que no esté vacía.

Intentar autenticación con contraseña en blanco o usando el mismo valor del nombre de usuario como contraseña en cuentas recién creadas, verificando si el sistema carece de validación mínima.

Contraseñas por defecto en cuentas nuevas

- **Determinar cómo se generan los nombres de usuario**

De acuerdo con los resultados en la Figura 43, se verificó que el sistema genera automáticamente el nombre de usuario utilizando el número de cédula del usuario registrado. Asimismo, se constató que el usuario no tiene la posibilidad de seleccionar o modificar dicho identificador. Debido a que el formato empleado es numérico y predecible, cualquier persona que conozca el número de cédula de un usuario puede inferir directamente su nombre de usuario, lo que facilita procesos de enumeración de cuentas y ataques de adivinación de credenciales.

- **Probar contraseña vacía o igual al nombre de usuario**

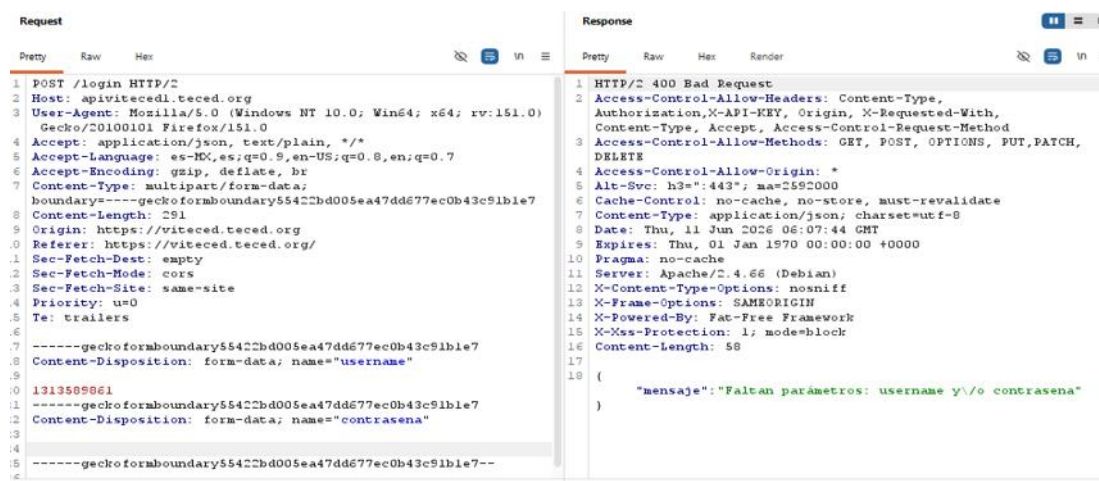


Figura 42 Validación de parámetros en el endpoint de autenticación ante el envío de un campo de contraseña vacío.

Fuente: Elaboración propia

Los resultados obtenidos muestran comportamientos diferentes frente a las dos comprobaciones realizadas. En el caso de la contraseña vacía, la solicitud fue rechazada mediante una respuesta HTTP 400 Bad Request, acompañada del mensaje “*Faltan parámetros: username y/o contrasena*”. Esto demuestra que la aplicación valida la presencia del campo de contraseña antes de procesar la autenticación y, por tanto, no permite iniciar sesión utilizando una contraseña vacía.

En cambio, al utilizar como contraseña el mismo valor correspondiente al número de cédula empleado como nombre de usuario, la aplicación respondió con HTTP 200 OK, el mensaje “*Autenticación exitosa*” y un token JWT, confirmando que la autenticación fue completada correctamente. Este comportamiento demuestra que la plataforma permite utilizar el identificador de usuario como contraseña, sin impedir el uso de una credencial trivial y fácilmente predecible.

Considerando que el nombre de usuario en ViTECED corresponde al número de cédula y que este identificador puede ser conocido o inferido, permitir que el mismo valor sea utilizado como contraseña reduce la dificultad necesaria para obtener acceso a una cuenta cuando un usuario haya establecido esta combinación. Por ello, el resultado constituye una debilidad en la política de contraseñas, específicamente relacionada con la aceptación de credenciales idénticas entre usuario y contraseña.

Modalidad: Caja Gris

Resultados

Mediante una entrevista con el equipo de desarrollo (Ver Anexo 2), el equipo de desarrollo indicó que las cuentas de usuario de ViTECED no utilizan credenciales administrativas predeterminadas para el acceso al sistema. El nombre de usuario es asignado automáticamente mediante el número de cédula, mientras que la contraseña es definida por el usuario durante el registro. De acuerdo con la información proporcionada por el equipo de desarrollo, no se mantienen cuentas de usuario con credenciales predeterminadas habilitadas para el acceso a la plataforma.

Las cuentas son creadas mediante el proceso establecido por el sistema y requieren la definición de una contraseña por parte del usuario. Además, mediante la revisión de los registros de usuarios se verificó el mecanismo de almacenamiento y asignación de las credenciales. No se identificaron cuentas con contraseñas vacías ni registros asociados a credenciales administrativas predeterminadas.

El nombre de usuario se encuentra asociado al número de cédula, mientras que la contraseña es definida durante el registro. Por otro lado, la revisión del mecanismo de registro permitió determinar que ViTECED no genera automáticamente contraseñas para las nuevas cuentas. La contraseña es definida directamente por el usuario durante el registro y el sistema establece restricciones sobre su formato. Por tanto, no existe un procedimiento de generación automática cuya predictibilidad deba ser evaluada.

- **Categoría de prueba: Pruebas de Autorización**

Identificador de prueba: Pruebas para determinar el escalamiento de privilegios

Manipulación de parámetros de rol/grupo en peticiones POST

Para comprobar la existencia de posibles mecanismos de escalamiento de privilegios, inicialmente se accedió a la plataforma con una cuenta con rol de administrador y se ejecutaron diferentes funciones disponibles para este perfil, cuyas solicitudes HTTP fueron interceptadas mediante Burp Suite para identificar los endpoints, métodos, encabezados y parámetros asociados a operaciones administrativas.

Posteriormente, se cerró la sesión del administrador y se inició sesión con una cuenta de paciente, correspondiente a un nivel de privilegios inferior. A partir de las solicitudes previamente capturadas, se seleccionaron dos operaciones administrativas para comprobar si podían ejecutarse utilizando las credenciales del usuario paciente: la

creación de una especialidad médica y la creación de un nuevo usuario administrador. Para realizar la comprobación, se reemplazó el token de autenticación del administrador por el token correspondiente a la cuenta de paciente y se modificaron los parámetros de la solicitud POST.

Al enviar la solicitud al servidor con las credenciales del paciente, se obtuvo como respuesta el código HTTP 200 OK y el mensaje “Especialidad creada”, confirmando que la operación fue procesada correctamente. Este resultado evidenció que una cuenta con rol paciente podía ejecutar una función administrativa que debía estar restringida al rol administrador, demostrando una deficiencia en la validación de privilegios del lado del servidor y la existencia de una vulnerabilidad de escalamiento vertical de privilegios.

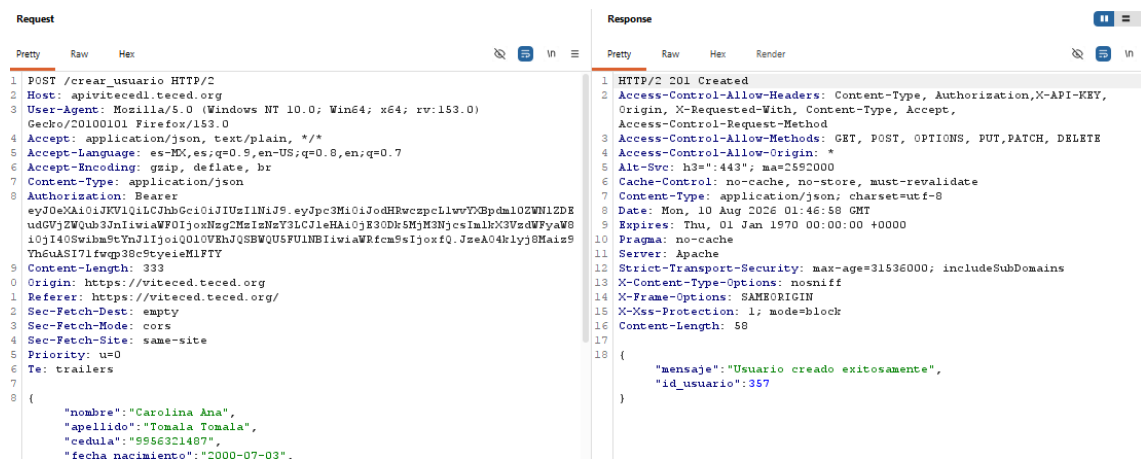


Figura 43 Solicitud de crear un usuario administrador con token de paciente.

Fuente: Elaboración propia

Como se muestra en la Figura 44, se capturó mediante Burp Suite la solicitud utilizada por el administrador para crear un nuevo usuario con privilegios administrativos. Muestra que el token del administrador fue sustituido por el token del paciente y se modificaron los parámetros enviados mediante el método POST para establecer los datos del nuevo usuario administrador. La solicitud modificada fue enviada al servidor utilizando únicamente la sesión correspondiente al usuario paciente. El servidor respondió con el código HTTP 201 Created y el mensaje “Usuario creado exitosamente”, confirmando que la operación fue ejecutada correctamente. El resultado confirma que una cuenta con rol paciente pudo ejecutar una función restringida al rol administrador mediante la manipulación de la solicitud HTTP.

Hallazgo Identificado

HALLAZGO VULN-004: Escalamiento vertical de privilegios por ausencia de control de acceso en el backend

Descripción: El servidor no valida el rol del usuario autenticado al procesar peticiones HTTP en endpoints administrativos. Se confirmó que usuarios con el rol Paciente pueden crear especialidades médicas y aprovisionar nuevos usuarios Administradores reenviando peticiones con su propio token JWT.

Impacto Potencial: Compromiso total de la confidencialidad, integridad y disponibilidad del sistema. Un atacante con una cuenta básica de paciente puede tomar el control completo de la plataforma creando su propia cuenta de Administrador.

Recomendación: Implementar un middleware de autorización estricto a nivel de backend que verifique el rol del usuario (`checkRole('administrador')`) en cada endpoint administrativo antes de procesar la solicitud, devolviendo un código 403 Forbidden cuando el rol no coincida.

- **Categoría de prueba: Pruebas de Administración e sesión**

Identificador de prueba: Pruebas de los atributos de las cookies

Ejecución de la prueba:

Análisis de cookies

Durante la evaluación de la aplicación se analizaron las solicitudes y respuestas HTTP generadas durante su funcionamiento, se inspeccionaron los encabezados HTTP de las solicitudes y respuestas con el propósito de identificar mecanismos utilizados para la gestión de sesiones y el intercambio de información entre el cliente y el servidor.

Como resultado de la revisión, presentada en la Figura 45, no se identificó el encabezado Set-Cookie en las respuestas HTTP analizadas, ni se evidenció el uso de cookies como mecanismo para la gestión de sesiones o autenticación de los usuarios. Por lo tanto, no fue posible identificar cookies sobre las cuales fuera posible verificar la configuración de los atributos de seguridad Secure, HttpOnly, Domain, Path y Expires. De acuerdo con las características observadas durante la evaluación, la aplicación utiliza un mecanismo de autenticación basado en tokens, por lo que la verificación específica de los atributos de seguridad de cookies no resulta aplicable en este escenario.

En consecuencia, la ausencia del atributo Set-Cookie no se considera por sí misma una vulnerabilidad de seguridad, debido a que dicho atributo únicamente resulta necesario cuando la aplicación utiliza cookies para mantener información de sesión. La seguridad del mecanismo alternativo utilizado para la autenticación debe ser evaluada mediante las pruebas correspondientes a la gestión, almacenamiento, transmisión, expiración y protección de los tokens.

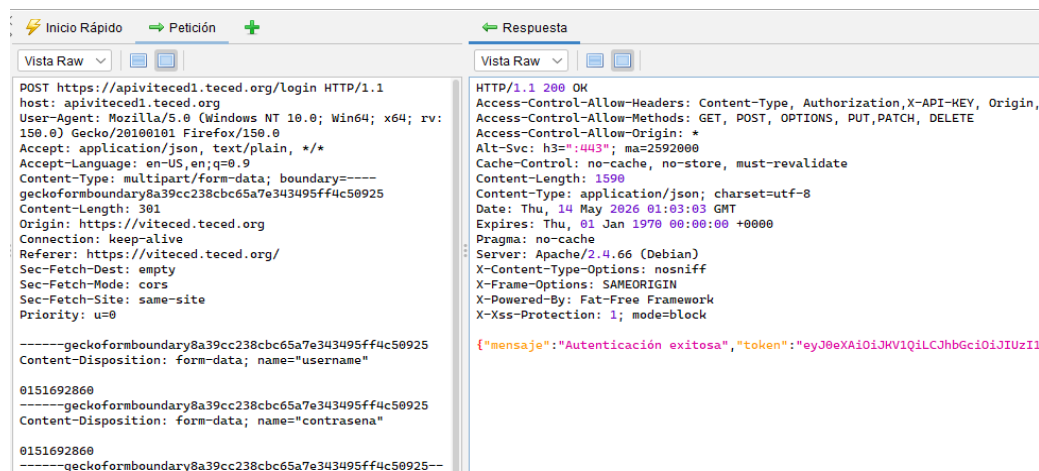


Figura 44 Verificación del atributo de cookie - OWASP ZAP.

Fuente: Elaboración propia

- **Categoría de prueba: Prueba de Administración e sesión**

Identificador de prueba: Prueba para falsificación de petición de sitio cruzado (CSRF)

Ejecución de la prueba:

Modalidad: Caja Negra

Resultados

Para verificar la existencia de una posible vulnerabilidad de tipo Cross-Site Request Forgery (CSRF), se evaluó el comportamiento del endpoint utilizado para el cambio de contraseña de la plataforma ViTECED. Inicialmente, se inició sesión en la aplicación web utilizando una cuenta con rol de paciente y, mediante la herramienta Burp Suite, se interceptó la solicitud legítima generada al realizar el cambio de contraseña. Durante el análisis de dicha solicitud se identificó el endpoint POST `/editar_contra`, así como los parámetros utilizados para efectuar la operación, correspondientes al identificador del usuario, la contraseña actual y la nueva contraseña. Asimismo, se verificó que la solicitud legítima incorporaba un mecanismo de autenticación mediante un token, como se muestra en la Figura 79.

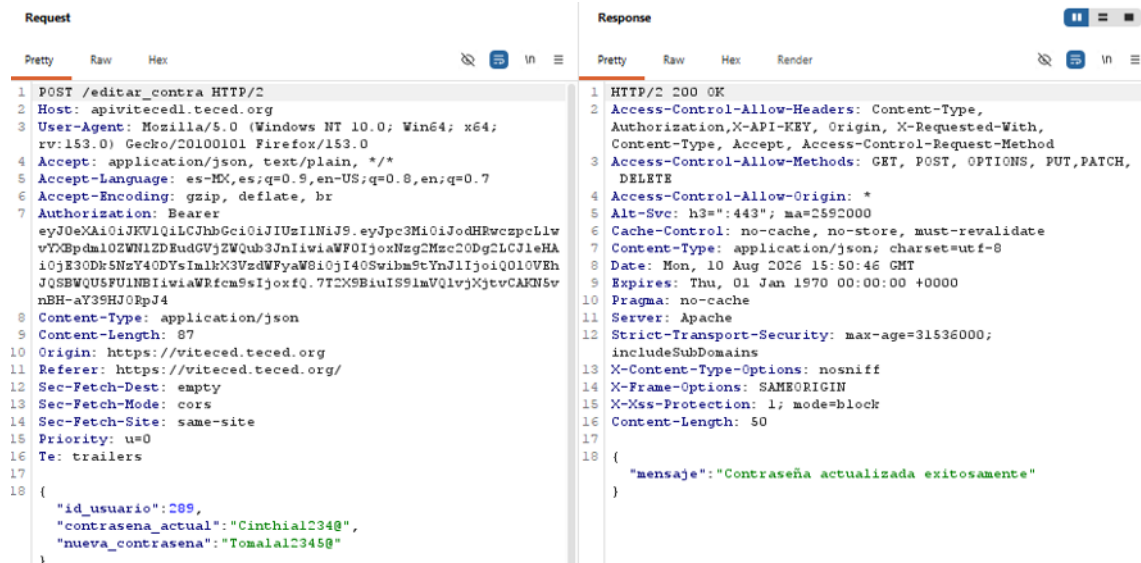


Figura 45 Solicitud legítima POST /editar_contra capturada mediante Burp Suite.

Fuente: Elaboración propia

Posteriormente, con el propósito de simular el comportamiento de un sitio externo que intentara inducir al navegador de un usuario autenticado a ejecutar una operación sobre la plataforma, se creó una página HTML independiente de ViTECED, como se muestra en la Figura 46. En esta página se incorporó un código JavaScript mediante la función `fetch()`, utilizando la URL del endpoint `/editar_contra`, el método POST y los parámetros previamente identificados en la solicitud legítima. No se incluyó el token de autenticación observado en la petición original, debido a que la prueba buscaba determinar si la operación podía ser ejecutada desde un origen externo sin disponer de las credenciales requeridas para autenticar la solicitud.

La página HTML fue ejecutada desde el origen local `http://127.0.0.1:5500`, simulando un sitio externo a la aplicación ViTECED. Al cargar la página, el código ejecutó automáticamente la solicitud hacia la API `apiviteced1.teced.org`, permitiendo observar su comportamiento mediante las herramientas de desarrollador del navegador. Como se muestra en la Figura 81, se identificó una solicitud previa OPTIONS con respuesta HTTP 200 OK, asociada a la comunicación cross-origin, seguida de la solicitud POST `/editar_contra`.

En la solicitud POST /editar_contra se obtuvo una respuesta HTTP 401 Unauthorized, lo que indica que la API no aceptó la solicitud como una operación autenticada. De igual manera, la consola del navegador confirmó la recepción del código HTTP 401. En consecuencia, la solicitud generada desde el origen externo no logró ejecutar el cambio de contraseña.

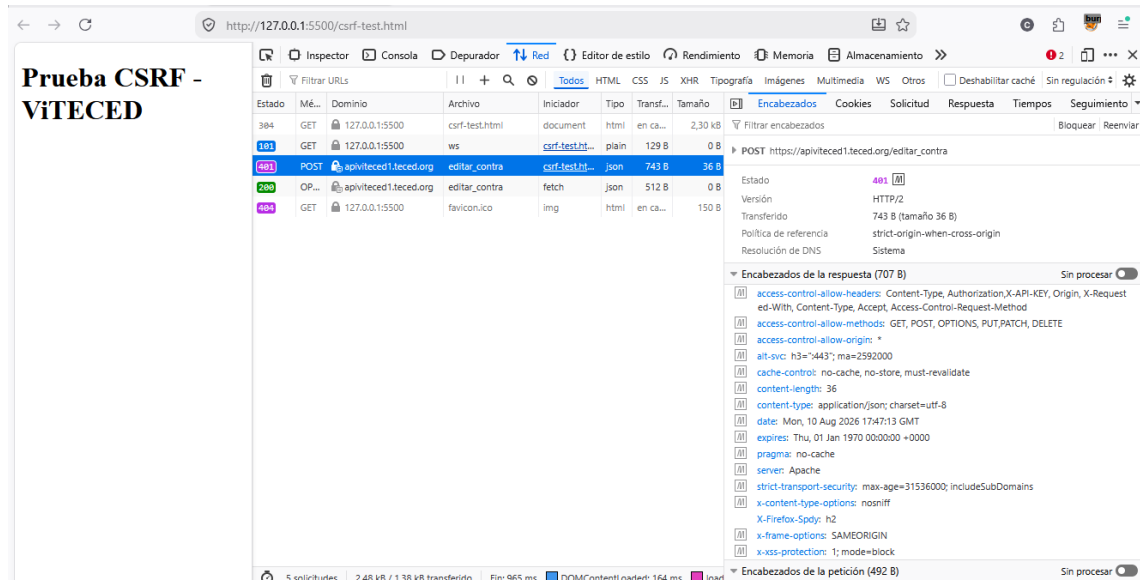


Figura 46 Solicitud generada desde el origen externo.

Fuente: Elaboración propia

Los resultados obtenidos permiten determinar que, bajo las condiciones evaluadas, no se evidenció la ejecución no autorizada del cambio de contraseña mediante una solicitud generada desde un origen externo. El rechazo de la solicitud estuvo asociado a la ausencia del mecanismo de autenticación requerido por el endpoint, específicamente el token presente en la solicitud legítima capturada mediante Burp Suite. Por tanto, durante la prueba realizada no se evidenció una explotación CSRF exitosa sobre el endpoint POST /editar_contra.

Modalidad: Prueba de Caja Gris

Resultados

Gestión de sesiones

Como parte de la evaluación de la falsificación de petición de sitio cruzado (CSRF), se realizó una prueba de caja gris orientada a determinar el mecanismo utilizado por ViTECED para gestionar la sesión y establecer si la autenticación depende de

Inicialmente, se realizó el proceso de autenticación en la aplicación utilizando una cuenta de prueba. Mediante la captura del tráfico HTTP se observó que, después de un inicio de sesión exitoso, el servidor proporcionó un JSON Web Token (JWT) como credencial de autenticación, sin identificarse el establecimiento de una cookie de sesión mediante el encabezado Set-Cookie, como se muestra en la Figura 47. El token obtenido constituye el mecanismo utilizado por la aplicación para identificar y autenticar las solicitudes posteriores realizadas por el usuario.



Posteriormente, se realizó una navegación normal dentro de la aplicación y se capturó mediante Burp Suite una solicitud correspondiente a un recurso protegido. En dicha solicitud se identificó el encabezado Authorization: Bearer <TOKEN>, evidenciando que la credencial de autenticación es incorporada explícitamente por el cliente en las solicitudes que requieren autenticación, como se muestra en la Figura 48. No se identificó el uso de una cookie de sesión para transmitir automáticamente la credencial al servidor.

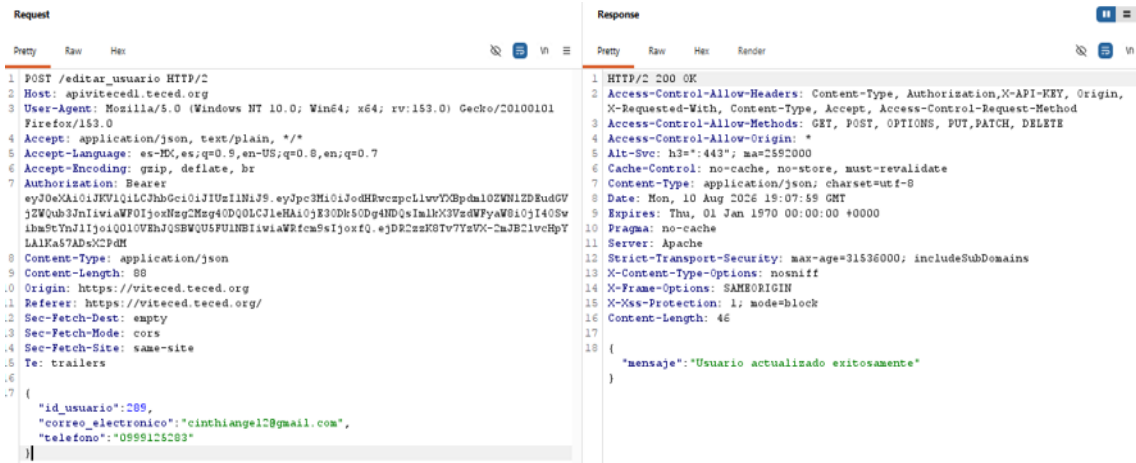


Figura 48 Solicitud autenticada mediante el encabezado Autorización.

Fuente: Elaboración propia

A continuación, se analizó el comportamiento del mecanismo de autenticación ante modificaciones controladas de la credencial. Para ello, se modificó el contenido del token presente en una solicitud previamente capturada y se procedió a reenviarla mediante Burp Suite. El servidor respondió con el código HTTP 401 Unauthorized, indicando que la credencial presentada no era válida, como se evidencia en la Figura 49. Este resultado demuestra que el servidor no acepta un token alterado y que realiza una validación de la credencial antes de procesar la solicitud protegida.

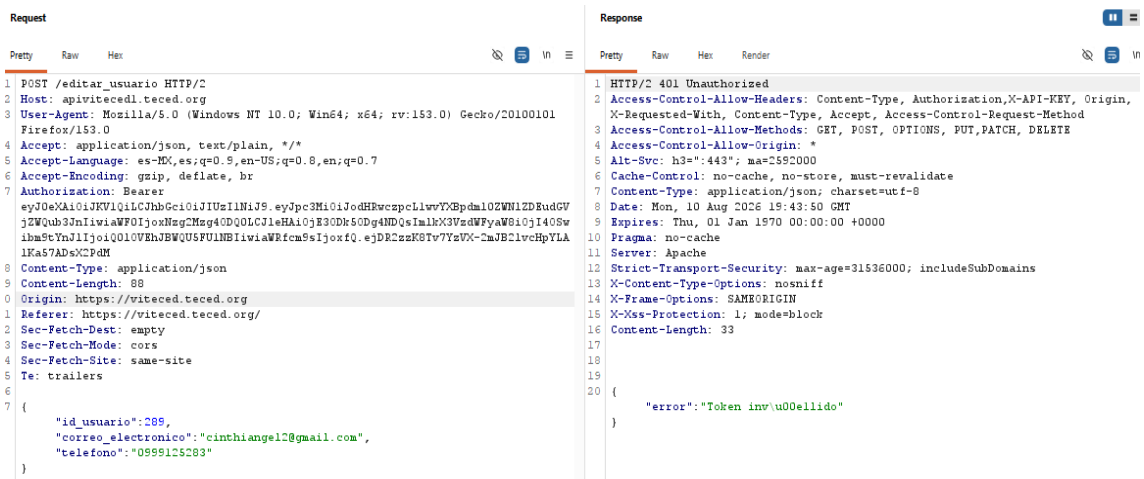


Figura 49 Rechazo de una solicitud con un token JWT modificado.

Fuente: Elaboración propia

Posteriormente, se eliminó completamente el encabezado de autenticación de la solicitud y se procedió a reenviarla. En este escenario, el servidor respondió nuevamente con el código HTTP 401 Unauthorized, acompañado de un mensaje que indicaba la ausencia del

token, como se muestra en la Figura 50. Este comportamiento confirma que la presencia de una credencial de autenticación válida es necesaria para acceder al recurso protegido.

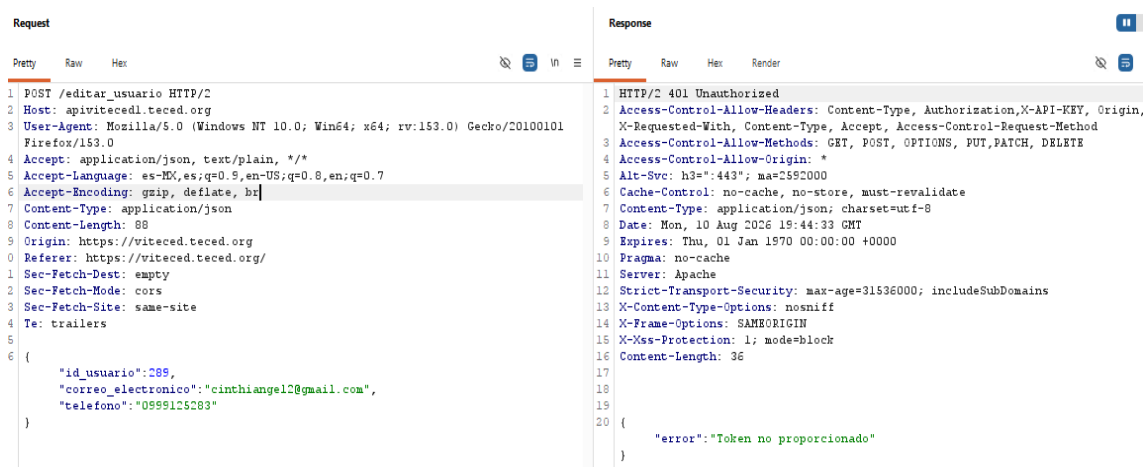


Figura 50 Rechazo de una solicitud sin token de autenticación.

Fuente: Elaboración propia

Finalmente, se analizó la estructura del token identificado como se muestra en la Figura 51, donde se determinó que corresponde a un JSON Web Token (JWT) compuesto por las secciones Header, Payload y Signature. En el encabezado se identificó el algoritmo de firma HS256 (HMAC-SHA256), mientras que en el contenido del token se observaron atributos relacionados con la sesión y el usuario, entre ellos el emisor (iss), la fecha de emisión (iat), la fecha de expiración (exp), el identificador del usuario (id_usuario) y el identificador del rol (id_rol). La posibilidad de visualizar o decodificar estos campos no fue considerada, por sí sola, como una vulnerabilidad, debido a que la información contenida en el Payload no constituye el mecanismo utilizado para garantizar la integridad del token; dicha función corresponde a la firma.

- **Categoría de prueba: Pruebas de manejo de errores**

Identificador de prueba: Pruebas de errores de código

Análisis de manejo de errores

Errores HTTP 404 Not Found

Como se muestra en la Figura 52 se envió una solicitud GET hacia una ruta inexistente de la aplicación, utilizando un nombre de recurso arbitrario que no corresponde a ninguna funcionalidad conocida. El servidor respondió con el código HTTP 404 Not Found. Se revisó el cuerpo de la respuesta y sus encabezados HTTP, considerando campos como

Server y X-Powered-By, así como cualquier contenido que pudiera revelar información sobre el servidor web, sistema operativo, módulos instalados, framework, rutas internas u otros componentes tecnológicos. Como resultado, no se evidenció divulgación de información técnica o sensible mediante la respuesta HTTP 404. La respuesta se limitó a indicar que el recurso solicitado no fue encontrado, sin exponer información adicional sobre la infraestructura de la plataforma.

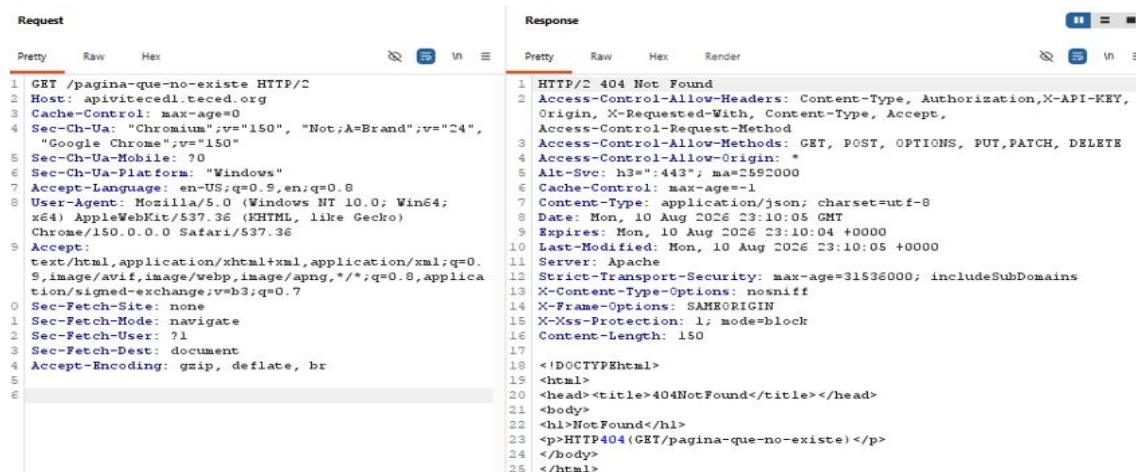


Figura 51 Análisis de manejo de error HTTP 404 Not Found.

Fuente: Elaboración propia

Errores HTTP 400 Bad Request

Se modificó una solicitud HTTP válida eliminando la cabecera Host y posteriormente se envió la petición modificada al servidor. En lugar del código esperado HTTP 400 Bad Request, el servidor respondió con HTTP 404 Not Found. Se revisaron el cuerpo y los encabezados de la respuesta para identificar información relacionada con la versión del servidor web, lenguaje de programación, módulos, framework o configuración interna. Como resultado, no se reprodujo el código HTTP 400 y no se evidenció divulgación de información técnica o sensible en la respuesta 404 obtenida.

Errores HTTP 405 Method Not Allowed

Como se muestra en la Figura 53, se seleccionó un endpoint conocido de la aplicación y se modificó el método HTTP de la solicitud original de POST a DELETE. La solicitud modificada fue enviada al servidor, que respondió con el código HTTP 405 Method Not Allowed. Se revisaron el cuerpo y los encabezados de la respuesta para determinar si se exponían los métodos permitidos, la versión del servidor, módulos instalados u otra información relacionada con la infraestructura.

Como resultado, se confirmó el rechazo del método no permitido mediante HTTP 405, pero no se evidenció divulgación de información técnica o sensible adicional asociada con la versión o configuración del servidor.

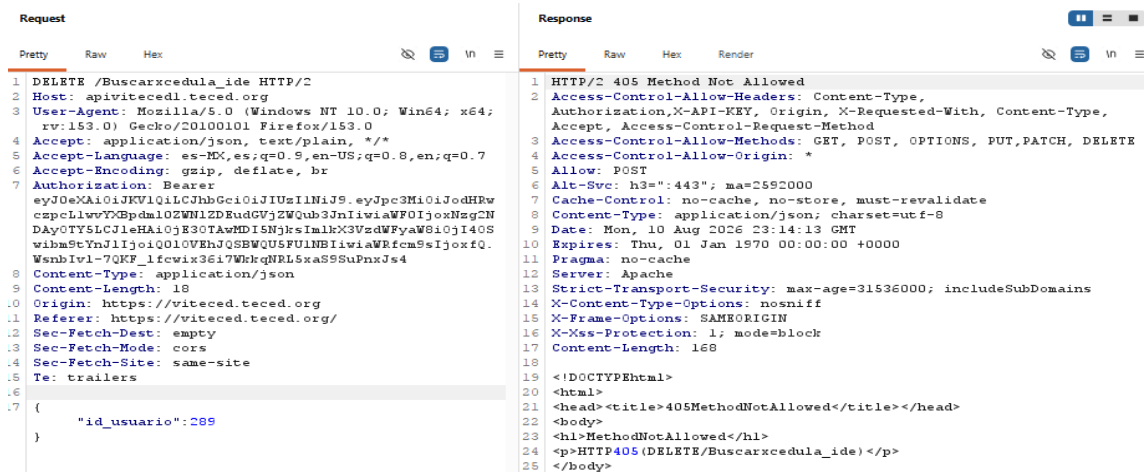


Figura 52 Análisis de manejo de error HTTP 405 Method Not Allowed.

Fuente: Elaboración propia

Errores HTTP 408 Request Timeout

La evidencia obtenida durante la prueba, se estableció una conexión TLS directa contra apiviteced1.teced.org:443 y viteced.teced.org mediante openssl s_client y se envió una solicitud HTTP intencionalmente incompleta, incluyendo la línea de solicitud y la cabecera Host: pero omitiendo el doble salto de línea final que indica al servidor el fin de la cabecera manteniéndola abierta sin enviar datos adicionales durante el tiempo de espera del servidor.

El servidor no genera una respuesta HTTP 408 ante solicitudes incompletas; simplemente termina la conexión TCP/TLS sin enviar datos. Este comportamiento es favorable desde el punto de vista de seguridad, ya que no divulga ninguna información sobre el software del servidor, su versión ni componentes del stack tecnológico en situaciones de timeout. El código 408 no es reproducible en este servidor.

Errores HTTP 501 Method Not Implemented

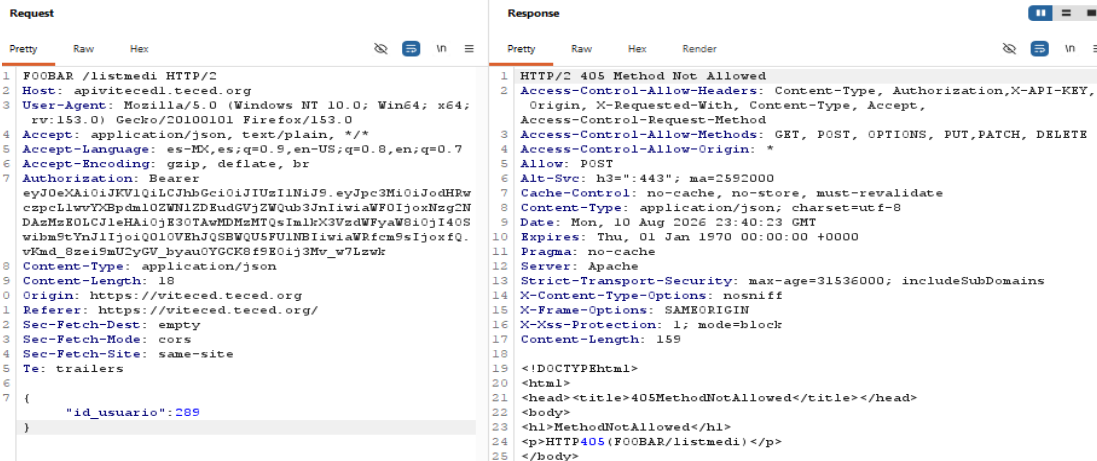


Figura 53 Respuesta HTTP 405 ante la solicitud evaluada para el manejo de errores HTTP 501 Method Not Implemented.

Fuente: Elaboración propia

En la Figura 54, se observa que se construyó una solicitud dirigida al componente backend apiviteced1.teced.org, utilizando el método HTTP arbitrario FOOBAR. El servidor no respondió con el código esperado HTTP 501 Method Not Implemented; en su lugar, respondió con HTTP 405 Method Not Allowed. Se revisaron el cuerpo y los encabezados de la respuesta para identificar información relacionada con la versión del servidor, módulos activos u otros componentes tecnológicos. Como resultado, el código HTTP 501 no fue reproducible y no se evidenció divulgación de información técnica o sensible adicional en la respuesta HTTP 405 generada ante el método arbitrario.

Errores de conectividad con la base de datos

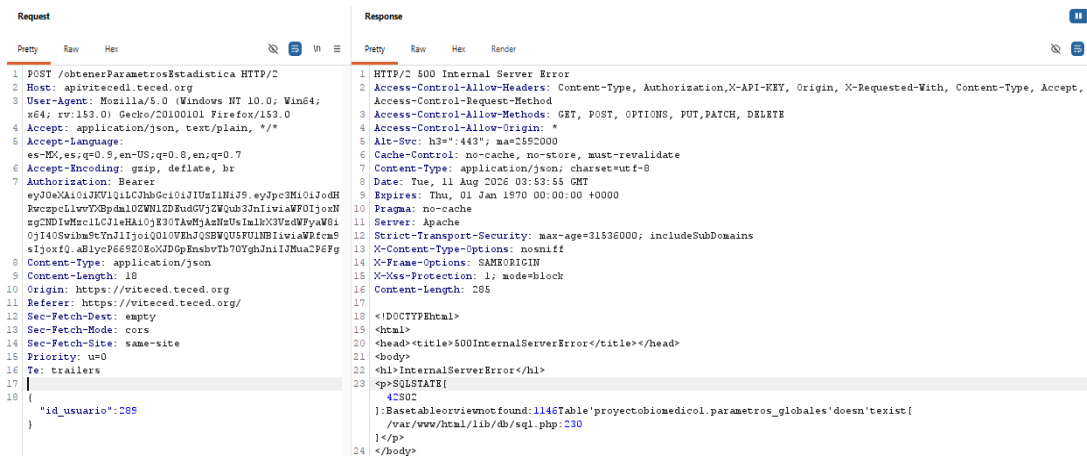


Figura 54 Análisis de manejo de error de conectividad con la base de datos.

Fuente: Elaboración propia

Durante la evaluación del endpoint POST /obtenerParametrosEstadistica se envió una solicitud autenticada con el parámetro id_usuario. El servidor respondió con el código HTTP 500 Internal Server Error, indicando que no pudo completar el procesamiento de la solicitud debido a un error interno relacionado con la base de datos.

La respuesta del servidor expuso directamente el mensaje de excepción SQL:

SQLSTATE[42S02]: Base table or view not found: 1146 Table 'proyectobiomedico1.parametros_globales' doesn't exist

Este mensaje evidencia que la aplicación intentó acceder a la tabla parametros_globales de la base de datos proyectobiomedico1, pero dicha tabla no se encuentra disponible. Adicionalmente, la respuesta revela información interna del servidor, incluyendo la ruta del archivo /var/www/html/lib/db/sql.php:230, lo que permite identificar la ubicación del componente que generó el error. Por tanto, se evidencia una gestión inadecuada de errores de base de datos, debido a que los detalles de la excepción SQL son enviados directamente al cliente en lugar de presentar un mensaje genérico y registrar el detalle técnico únicamente en los mecanismos internos de monitoreo y registro.

Exposición de datos personales de usuario en stack trace

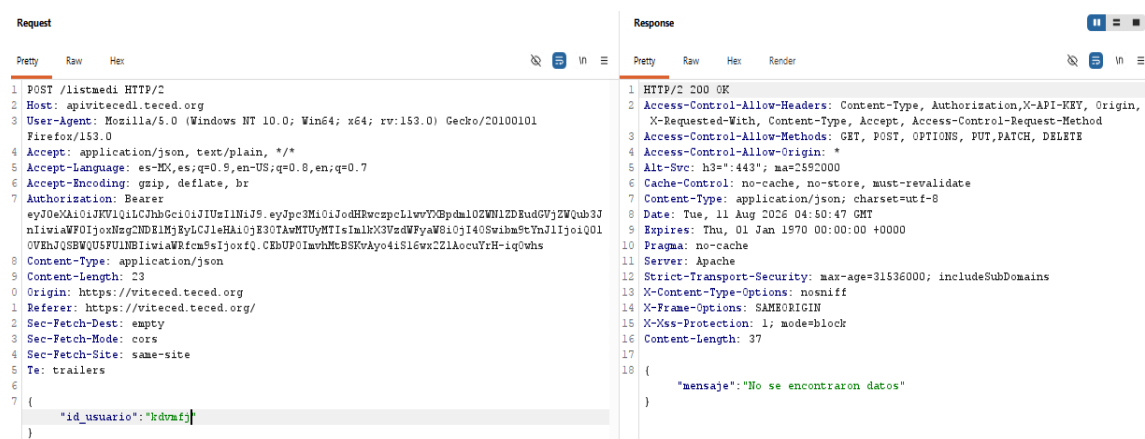


Figura 55 Análisis de manejo de error de exposición de datos personales de usuario en stack trace.

Fuente: Elaboración propia

De acuerdo con la Figura 56, se observa que se identificó un campo cuyo valor esperado correspondía a un dato numérico y se modificó su contenido enviando un valor de tipo texto. La aplicación respondió con HTTP 200 OK y un mensaje indicando que no se encontraron datos. Se revisó el contenido de la respuesta para identificar posibles trazas de pila, rutas internas, nombres de clases o funciones, información de configuración y

datos personales de usuarios. Como resultado, no se evidenció divulgación de datos personales ni información técnica mediante una traza de pila. La aplicación no devolvió mensajes de excepción ni información interna asociada con el valor utilizado durante la prueba.

Errores de autenticación y autorización de componentes intermedios

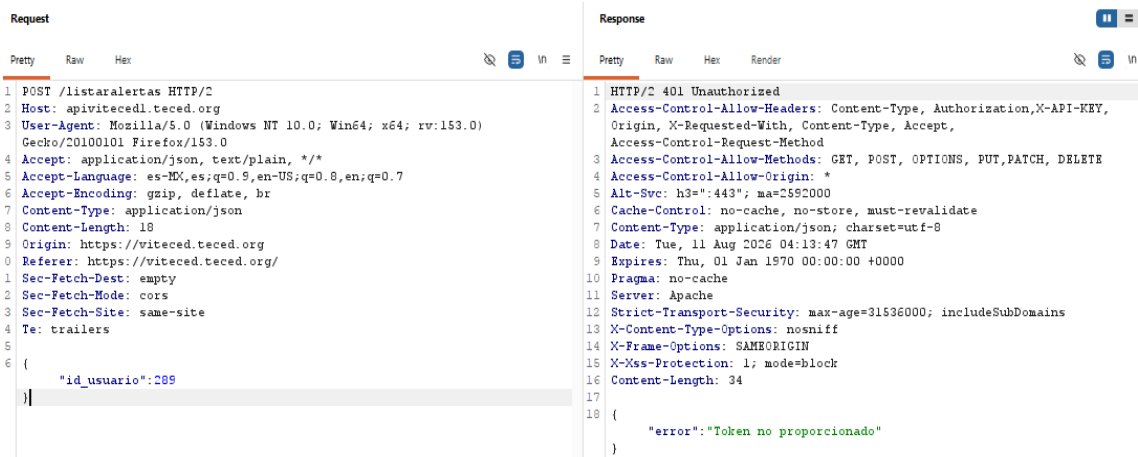


Figura 56 Análisis de petición ante una solicitud sin token.

Fuente: Elaboración propia

Según lo observado en la Figura 57, se envió una solicitud POST al endpoint /listaralertas del componente backend apivitecedl.teced.org sin incluir la cabecera Authorization, manteniendo el parámetro id_usuario: 209 en el cuerpo. El servidor respondió con HTTP 401 Unauthorized y el mensaje “Token no proporcionado”.

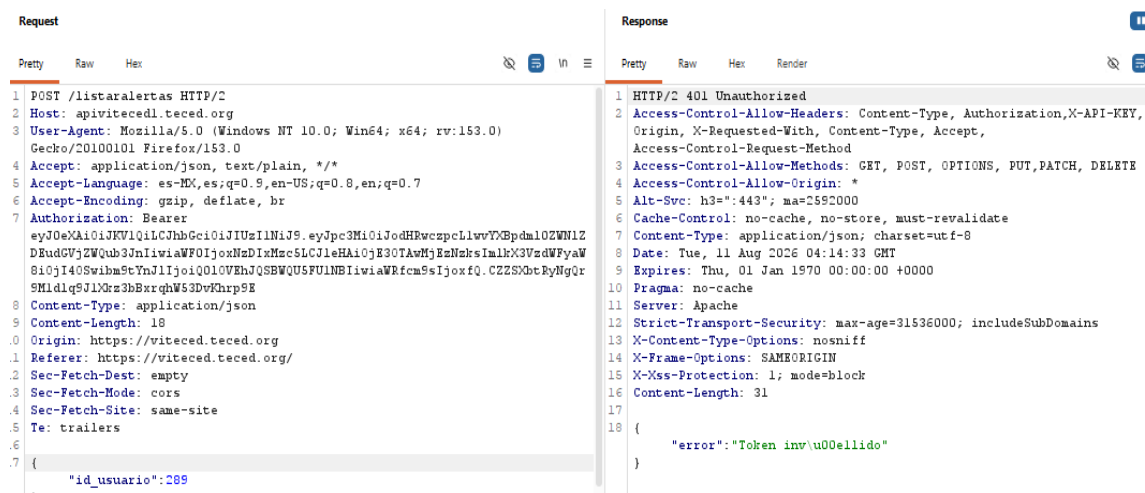


Figura 57 Análisis de solicitud ante un token modificado.

Fuente: Elaboración propia

Tal como se evidencia en la Figura 58, se envió una segunda solicitud al mismo endpoint utilizando la cabecera: Authorization: Bearer inválido En este caso, el servidor respondió nuevamente con HTTP 401 Unauthorized, pero indicó “Token inválido”. En ambas respuestas se identificó el encabezado Server: Apache. Asimismo, se observó que la aplicación diferencia entre la ausencia de un token y la presentación de un token inválido mediante mensajes de respuesta diferentes. Como resultado, se evidenció divulgación de información técnica mediante el encabezado Server: Apache y se identificó información adicional sobre el estado de autenticación mediante los mensajes diferenciados. No obstante, no se evidenció la versión de Apache ni información que permitiera identificar un componente intermedio específico.

Errores HTTP 403 Forbidden y listado de directorios

Mediante la navegación por la plataforma y el uso de un proxy en la cual se captura todas las peticiones del sistema, se evidencio las siguientes direcciones:

Directorios descubiertos

Contenido	Tipo
/assets/	Recursos estáticos (imágenes, estilos, scripts)
/assets/BD/	Archivos de base de datos en JSON
/assets/img/	Imágenes de la aplicación
/svg/	Archivos SVG (iconos)
/img/	Imágenes adicionales

Tabla 54 Direcciones identificadas en el sistema

Verificación de cada directorio

Se realizó un acceso directo a cada ruta, ante la cual el servidor respondió con HTTP 403 Forbidden. Se revisó la respuesta para determinar si el acceso denegado permitía identificar la existencia del recurso, información sobre la estructura de directorios o datos relacionados con la tecnología utilizada por el servidor. Como resultado, sí se evidenció divulgación de información técnica, debido a que la respuesta HTTP 403 permitió identificar nginx y su versión. Sin embargo, no se evidenció un listado del contenido del directorio.

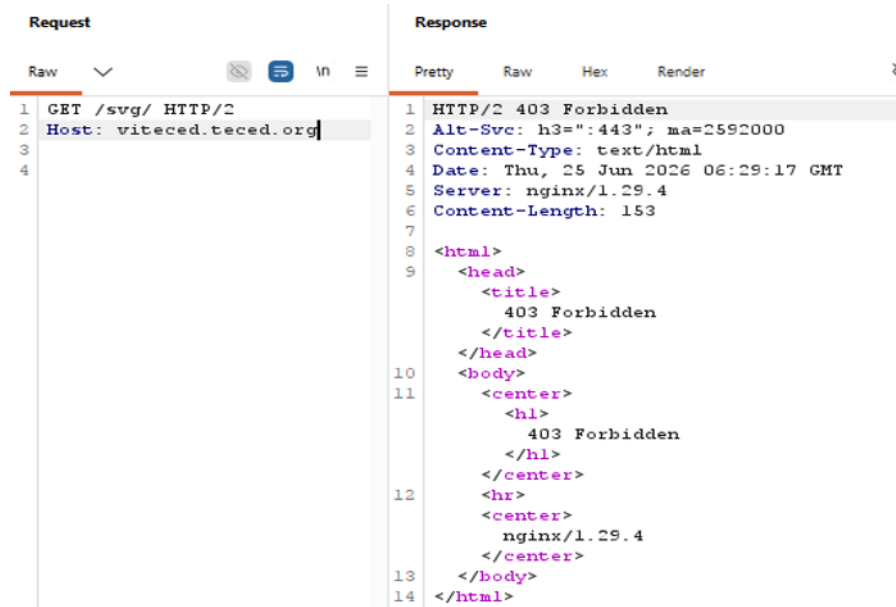


Figura 58 Verificación de acceso al directorio /svg/.

Fuente: Elaboración propia

Conclusión de la prueba

La evaluación evidenció que la plataforma ViTECED gestiona de manera segura las peticiones malformadas e inexistentes en la mayoría de sus controladores. Sin embargo, presenta una exposición parcial de información técnica sensible mediante mensajes de error de base de datos en el backend.

Se recomienda corregir el controlador de excepciones de la base de datos para suprimir los mensajes descriptivos de SQLState y rutas internas, así como ocultar las versiones de los servidores web Apache y Nginx en las cabeceras HTTP de respuesta.

Hallazgo identificado

HALLAZGO VULN-005: Divulgación de información mediante mensajes de error

Descripción: Durante la evaluación se identificó que la aplicación devuelve al cliente información detallada cuando se produce un error durante la interacción con la base de datos. La respuesta HTTP 500 expuso el nombre de la base de datos proyectobiomedico1, el nombre de la tabla parametros_globales y la ruta interna /var/www/html/lib/db/sql.php:230, además del mensaje de excepción SQL. Esta información corresponde a detalles internos de la infraestructura y del funcionamiento de la aplicación que no deberían ser visibles para el usuario.

- **Categoría de prueba: Pruebas para criptografía débil**

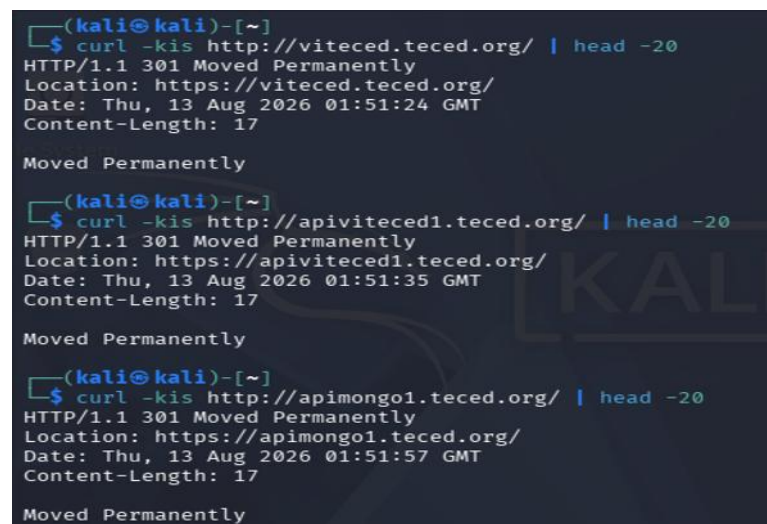
Identificador de prueba: Pruebas para el envío de información sensible por canales sin encriptar

Ejecución de la prueba:

Uso de HTTPS en la comunicación

Comando:

- `curl -kis http://viteced.teced.org/ | head -20`
- `curl -kis http://apiviteced1.teced.org/ | head -20`
- `curl -kis http://apimongo1.teced.org/ | head -20`



```
(kali㉿kali)-[~]
$ curl -kis http://viteced.teced.org/ | head -20
HTTP/1.1 301 Moved Permanently
Location: https://viteced.teced.org/
Date: Thu, 13 Aug 2026 01:51:24 GMT
Content-Length: 17

Moved Permanently

(kali㉿kali)-[~]
$ curl -kis http://apiviteced1.teced.org/ | head -20
HTTP/1.1 301 Moved Permanently
Location: https://apiviteced1.teced.org/
Date: Thu, 13 Aug 2026 01:51:35 GMT
Content-Length: 17

Moved Permanently

(kali㉿kali)-[~]
$ curl -kis http://apimongo1.teced.org/ | head -20
HTTP/1.1 301 Moved Permanently
Location: https://apimongo1.teced.org/
Date: Thu, 13 Aug 2026 01:51:57 GMT
Content-Length: 17

Moved Permanently
```

Figura 59 Análisis de políticas de seguridad en la capa de transporte (HTTPS)

Fuente: Elaboración propia

Site map

Scope

Issues

Site map filter: Hiding not found items; hiding CSS, image and general binary content; hiding 4xx responses; hiding empty folders

	Host	Method	URL	Params	Status code	Length	MIME type
>	apimongo1.teced.org						
>	apiviteced1.teced.org						
>	broker.teced.org						
>	viteced.teced.org						
	https://apimongo1.teced.org	GET	/datos/usuario				
	https://apimongo1.teced.org	GET	/logs/guardarQUITAESTO				
	https://apimongo1.teced.org	OPTIONS	/datos/usuario		200	502	text
	https://apimongo1.teced.org	OPTIONS	/logs/guardarQUITAESTO		200	502	text
	https://apimongo1.teced.org	POST	/datos/usuario	✓	200	358	JSON
	https://apiviteced1.teced.org	GET	/BuscarCedula_id				
	https://apiviteced1.teced.org	GET	/Guardammedico				
	https://apiviteced1.teced.org	GET	/LISTARCONSULTAS				
	https://apiviteced1.teced.org	GET	/LISTARECETAS				
	https://apiviteced1.teced.org	GET	/LIS_horarios				
	https://apiviteced1.teced.org	GET	/actualizarchat				
	https://apiviteced1.teced.org	GET	/asignarAsistente				
	https://apiviteced1.teced.org	GET	/buscararticulosponusuarios				
	https://apiviteced1.teced.org	GET	/crear_datosfisicos				
	https://apiviteced1.teced.org	GET	/listacentro		200	2029	JSON
	https://apiviteced1.teced.org	GET	/listadoima				
	https://apiviteced1.teced.org	GET	/listadoima?id_usuario=166	✓	200	511	JSON
	https://apiviteced1.teced.org	GET	/listadoima?id_usuario=289	✓	200	511	JSON
	https://apiviteced1.teced.org	GET	/listaespe		200	1648	JSON
	https://apiviteced1.teced.org	GET	/listanota		200	3373	JSON
	https://apiviteced1.teced.org	GET	/listaciencias				
	https://apiviteced1.teced.org	GET	/listapais		200	3755	JSON
	https://apiviteced1.teced.org	GET	/listaparametros		200	1208	JSON
	https://apiviteced1.teced.org	GET	/listaprovincias				
	https://apiviteced1.teced.org	GET	/listarHistorialDatosFisicos				
	https://apiviteced1.teced.org	GET	/listar_datosfisicos				
	https://apiviteced1.teced.org	GET	/listar_dispositivo				
	https://apiviteced1.teced.org	GET	/listaralertas				
	https://apiviteced1.teced.org	GET	/listaralertasfecha				
	https://apiviteced1.teced.org	POST	/listarantepir				
	https://broker.teced.org	GET	/mott		500	212	text
	https://apimongo1.teced.org	GET	/datos/usuario				
	https://apimongo1.teced.org	GET	/logs/guardarQUITAESTO				
	https://apimongo1.teced.org	OPTIONS	/datos/usuario		200	502	text
	https://apimongo1.teced.org	OPTIONS	/logs/guardarQUITAESTO		200	502	text
	https://apimongo1.teced.org	POST	/datos/usuario	✓	200	358	JSON

Figura 60 Verificación del uso del protocolo HTTPS

Fuente: Elaboración propia

Interpretación de los resultados

Dominio	Código	Redirección a
viteced.teced.org	301 Moved Permanently	https://viteced.teced.org/
apiviteced1.teced.org	301 Moved Permanently	https://apiviteced1.teced.org/
apimongo1.teced.org	301 Moved Permanently	https://apimongo1.teced.org/

Tabla 55 Interpretación de resultados

Mediante la Figura 60 y 61 y la Tabla 56 se evidencia que los dominios evaluados redirigen correctamente las solicitudes realizadas mediante HTTP hacia sus respectivas versiones HTTPS. En los tres dominios se obtuvo el código de respuesta 301 Moved Permanently, indicando una redirección permanente hacia el canal de comunicación seguro. Adicionalmente, se realizó un análisis del tráfico generado durante la navegación de la aplicación mediante Burp Suite, donde se revisaron las solicitudes y respuestas registradas en el historial HTTP. Como resultado, se verificó que las comunicaciones realizadas con el frontend y los servicios evaluados se efectuaron mediante HTTPS, sin identificarse solicitudes que transmitieran información sensible mediante HTTP. Por lo

tanto, no se evidenció una condición de riesgo asociada al envío de información sensible a través de canales sin cifrar durante la prueba realizada.

Transmisión de credenciales

Mediante la Figura 62 se evidencia la solicitud POST /login generada durante el proceso de autenticación de la aplicación. En el cuerpo de la solicitud se identifican los campos username y contraseña, correspondientes a las credenciales utilizadas para iniciar sesión. La solicitud fue interceptada mediante Burp Suite para verificar el mecanismo empleado para transmitir esta información hacia el servidor. Se comprobó que la solicitud corresponde al endpoint `apiviteced1.teced.org/login` y que es generada mediante el dominio seguro `https://viteced.teced.org`. Aunque Burp Suite permite visualizar los valores enviados en el cuerpo de la solicitud, esta visualización corresponde a la inspección realizada por el proxy y no implica que dichos datos sean transmitidos en texto claro a través de la red.

Como prueba complementaria, se realizó una solicitud mediante curl, tal como se muestra en la Figura 63, utilizando explícitamente el protocolo HTTP hacia `http://apiviteced1.teced.org/login`, obteniéndose como respuesta el código HTTP 308 (Permanent Redirect). Este resultado indica que el servidor redirige la solicitud realizada mediante HTTP, por lo que no se evidenció el procesamiento de las credenciales mediante un canal HTTP sin cifrado.

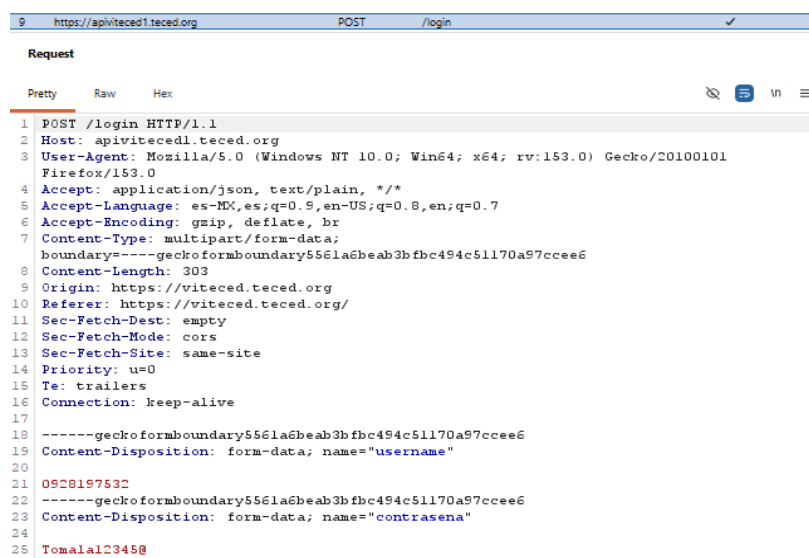


Figura 61 Transmisión de credenciales durante el inicio de sesión

Fuente: Elaboración propia

Comando: curl -kis -X POST <http://apiviteced1.teced.org/login> \ -H "Content-Type: application/json" \ -d '{"usuario":"0928197532","contrasena":"0928197532"}'

```
(kali@kali)-[~]
$ curl -kis -X POST http://apiviteced1.teced.org/login \
-H "Content-Type: application/json" \
-d '{"usuario":"0928197532","contrasena":"0928197532"}'
HTTP/1.1 308 Permanent Redirect
Location: https://apiviteced1.teced.org/login
Date: Fri, 26 Jun 2026 03:16:37 GMT
Content-Length: 18

Permanent Redirect
```

Figura 62 Procesamiento de solicitudes de inicio de sesión sin establecer una conexión segura

Fuente: Elaboración propia

Como evidencia adicional, mediante Wireshark se verificó el tráfico de red generado durante la comunicación con la aplicación. En la Figura 64, se observa el Client Hello desde 192.168.100.3 hacia la dirección 72.61.11.127, incluyendo el SNI viteced.teced.org, seguido del Server Hello y posteriormente paquetes identificados como Application Data, todos correspondientes al protocolo TLSv1.3. Esto confirma el establecimiento de un canal de comunicación cifrado entre el cliente y el servidor, impidiendo que los datos intercambiados sean observables directamente en texto claro mediante la captura de red.

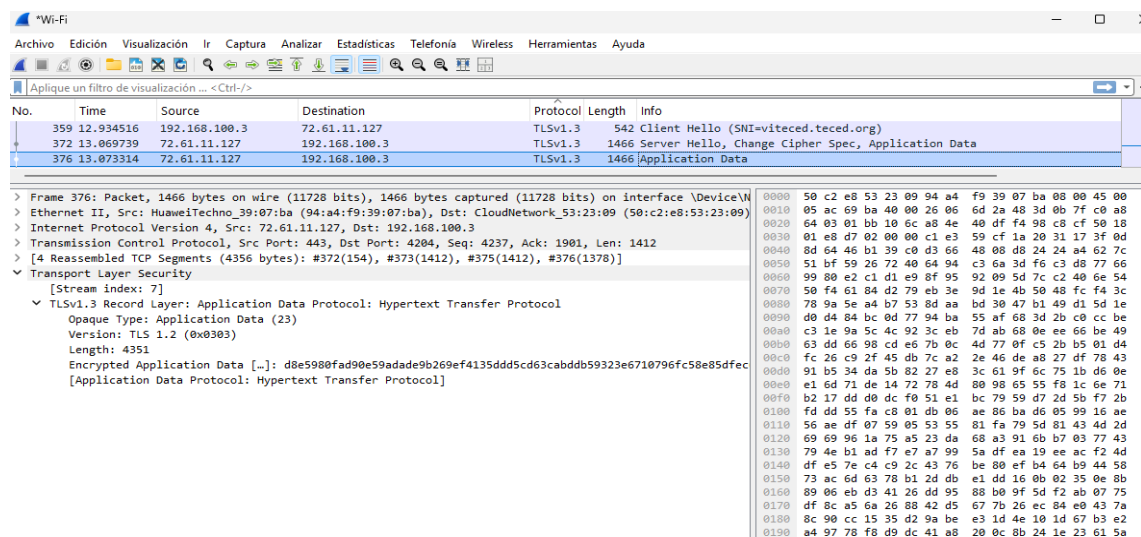


Figura 63 Análisis de información cifrada mediante Wireshark

Fuente: Elaboración propia

Protección de tokens e identificadores de sesión

La Figura 65, muestra una solicitud autenticada en la que se utiliza el token de acceso mediante la cabecera Authorization. Durante la revisión del historial de Burp Suite se

analizaron las solicitudes posteriores que requieren autenticación, como se muestra en la Figura 66, verificándose que todas las solicitudes que emplean el token se realizan mediante HTTPS. No se identificaron solicitudes en las que el token fuera transmitido mediante HTTP sin cifrado. Por lo tanto, durante la prueba realizada se determinó que el token de autenticación se encuentra protegido durante su transmisión mediante el uso de conexiones HTTPS.

Figura 64 Verificación de la protección del token mediante HTTPS

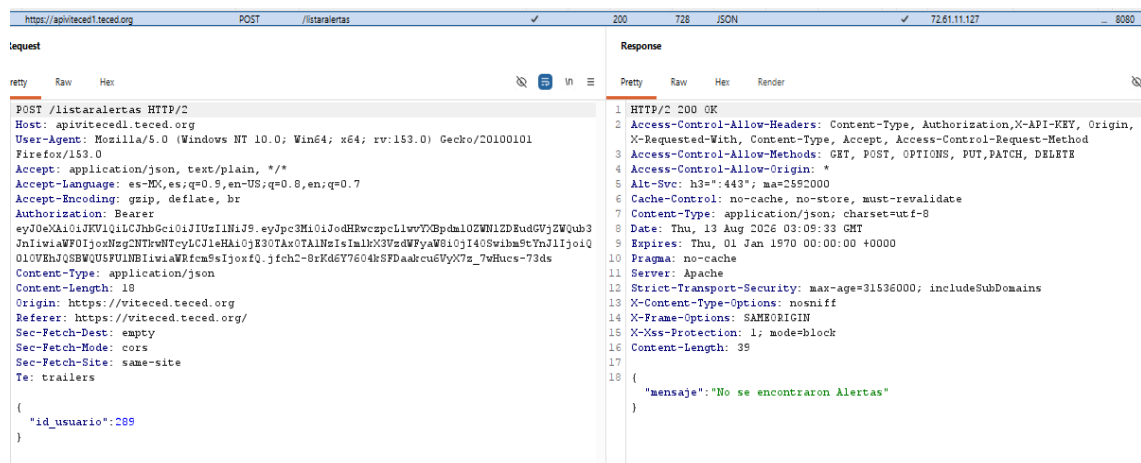


Figura 65 Verificación del uso del token de autenticación en solicitudes HTTPS.

Conclusión de la prueba

(viteced.teced.org, apiviteced1.teced.org y apimongo1.teced.org) imponen el uso de conexiones cifradas HTTPS mediante redirecciones permanentes (301 y 308).

Las credenciales de acceso y los tokens de autenticación JWT se encuentran protegidos durante toda su transmisión en la red. El análisis de paquetes mediante Wireshark confirmó el establecimiento de canal cifrado sobre el protocolo moderno TLSv1.3, impidiendo la captura de contraseñas o tokens en texto claro mediante técnicas de inspección de tráfico.

Evaluación de Riesgo

La metodología OWASP establece que cada una de las pruebas de seguridad debe ser evaluada en función de su nivel de riesgo, considerando el grado de criticidad de los hallazgos identificados [159]. Esta evaluación se fundamenta en un conjunto de factores que permiten determinar el impacto y la probabilidad de explotación de cada vulnerabilidad detectada durante el proceso de análisis.

Para la determinación del nivel de riesgo, se consideran diversos factores de evaluación definidos por OWASP, los cuales sirven como base para calcular de manera sistemática el riesgo asociado a cada una de las pruebas de seguridad realizadas [160]. Estos factores se describen a continuación y serán utilizados posteriormente para la valoración de los resultados obtenidos en cada prueba.

Factores de Probabilidad

- **Factores del agente de amenaza**

Nivel de habilidad: Determina las habilidades requeridas para descubrir una posible vulnerabilidad. Sus valores corresponden a: sin habilidades técnicas (1), algunas habilidades técnicas (3), usuario avanzado de computadoras (5), habilidades de redes y programación (6), habilidades de penetración de seguridad (9).

Motivación: Determina la relevancia de la información encontrada. Sus valores son: recompensa baja o nula (1), recompensa posible (4), recompensa alta (9).

Oportunidad: Evalúa la necesidad de permisos para acceder y recopilar información sobre una posible vulnerabilidad. Sus valores son: Acceso completo o recursos

costosos (0), acceso o recursos especiales (4), acceso o recursos limitados (7), ningún acceso o recursos (9).

Tamaño: Indica la cantidad de usuarios que podrían verse afectados por una posible amenaza. Sus valores son: desarrolladores (2), administradores de sistemas (2), usuarios de intranet (4), socios (5), usuarios autenticados (6), usuarios anónimos de Internet (9).

- **Factores de vulnerabilidad**

Facilidad de detección: El nivel de dificultad para identificar una vulnerabilidad. Sus valores son: prácticamente imposible (1), difícil (3), fácil (7), herramientas automatizadas disponibles (9).

Facilidad de explotación: El nivel de dificultad para explotar una vulnerabilidad. Sus valores son: teórica (1), difícil (3), fácil (5), herramientas automatizadas disponibles (9).

Conocimiento: Indica qué tan conocida es una vulnerabilidad por las personas que podrían intentar aprovecharla. Sus valores son: desconocida (1), oculta (4), obvia (6), conocimiento público (9).

Detección de intrusiones: Evalúa la posibilidad de detectar un ataque que aproveche una vulnerabilidad. Sus valores son: Detección activa en la aplicación (1), registrada y revisada (3), registrada sin revisión (8), no registrada (9).

Factores de Impacto

- **Factores de impacto técnico**

Pérdida de confidencialidad: Evalúa el nivel de afectación a la confidencialidad de la información ocasionado por una posible vulnerabilidad. Sus valores son: datos no sensibles mínimos divulgados (2), datos críticos mínimos divulgados (6), datos no sensibles extensos divulgados (6), datos críticos extensos divulgados (7), todos los datos divulgados (9).

Pérdida de integridad: Se refiere a la pérdida o modificación de la información ocasionada por una posible vulnerabilidad. Sus valores son: datos mínimos

ligeramente dañados (1), datos mínimos gravemente dañados (3), datos extensos ligeramente dañados (5), datos extensos gravemente dañados (7), todos los datos totalmente dañados (9).

Pérdida de disponibilidad: Determina si una posible vulnerabilidad podría ocasionar la pérdida o inaccesibilidad de la información. Sus valores son: interrupción mínima de servicios secundarios (1), interrupción mínima de servicios primarios (5), interrupción extensa de servicios secundarios (5), interrupción extensa de servicios primarios (7), pérdida total de todos los servicios (9).

Pérdida de responsabilidad: Determina si las acciones realizadas por un usuario pueden ser identificadas y relacionadas con una persona específica. Sus valores son: totalmente rastreables (1), posiblemente rastreables (7), completamente anónimos (9).

- **Factores de impacto empresarial**

Daños financieros: Determina las posibles pérdidas financieras que podrían generarse debido a una vulnerabilidad. Sus valores son: Menos que el costo de solucionar la vulnerabilidad (1), efecto menor en las ganancias anuales (3), efecto significativo en las ganancias anuales (7), quiebra (9).

Daño a la reputación: Indica si una posible vulnerabilidad podría causar una mala imagen o afectar la confianza en la organización. Sus valores son: Daño mínimo (1), pérdida de clientes importantes (4), pérdida de prestigio (5), daño a la marca (9).

Incumplimiento: Indica si una amenaza dentro de la aplicación web podría ocasionar una infracción. Sus valores son: infracción menor (2), infracción clara (5), infracción grave (7).

Violación de la privacidad: Indica el número de usuarios que podrían resultar afectados por un posible ataque a la aplicación web. Sus valores son: un individuo (3), cientos de personas (5), miles de personas (7), millones de personas (9).

Es importante considerar estos factores para establecer el nivel de severidad de una vulnerabilidad, ya que permiten identificar y priorizar los problemas que requieren atención. Su correcta evaluación facilita la aplicación de medidas de seguridad orientadas

a reducir los riesgos y proteger la información gestionada por una aplicación web frente a posibles ataques derivados de configuraciones o controles de seguridad inadecuados.

La valoración de estos factores permite obtener el nivel de severidad asociado a cada vulnerabilidad identificada. Los criterios de evaluación se determinan mediante el análisis de los resultados obtenidos en cada prueba de seguridad realizada, utilizando los valores de clasificación establecidos por la metodología de evaluación de riesgos de OWASP.

Probabilidad y niveles de impacto	
0 a <3	BAJO
3 a <6	MEDIO
6 a 9	ALTO

Tabla 56 Nivel de gravedad del riesgo

Fuente: Adaptado de [161]

En la Tabla 57, se presenta la estimación del nivel de riesgo, donde se combinan la probabilidad de ocurrencia y el impacto generado para determinar la severidad general de una vulnerabilidad.

Es importante estimar la severidad de los riesgos asociados a una aplicación, con el propósito de tomar decisiones fundamentadas sobre las vulnerabilidades identificadas y establecer cuáles requieren mayor atención.

Gravedad del riesgo general				
Impacto	ALTO	Medio	Alto	Crítico
	MEDIO	Bajo	Medio	Alto
	BAJO	Nota	Bajo	Medio
		BAJO	MEDIO	ALTO
	Probabilidad			

Tabla 57 Determinación de la gravedad de Riesgo

Fuente: Adaptado de [161]

En la Tabla 58, se presenta la clasificación de la gravedad de los riesgos, considerando los valores obtenidos de la probabilidad de ocurrencia y el nivel de impacto

Análisis estadístico de los resultados

Para el análisis estadístico se consolidaron las pruebas de seguridad realizadas sobre la aplicación web y móvil del sistema ViTECED. Se debe considerar que cada prueba principal se encuentra estructurada en diferentes subpruebas o verificaciones específicas, de acuerdo con los criterios establecidos por las guías OWASP utilizadas. Por esta razón, una prueba principal puede comprender varias evaluaciones técnicas y, consecuentemente, puede generar uno o más hallazgos de seguridad.

En este sentido, para el análisis general se contabilizó cada prueba principal como una unidad de evaluación, independientemente del número de subpruebas que la conforman. De esta manera, se obtuvo un total de 33 pruebas principales, de las cuales 14 presentaron uno o más hallazgos y 19 no presentaron hallazgos.

Resultado	Frecuencia	Porcentaje
Pruebas con hallazgos	14	42,42 %
Pruebas sin hallazgos	19	57,58 %
Total	33	100,00 %

Tabla 58 Frecuencia de los resultados de las pruebas

Los resultados muestran que el 42,42 % de las pruebas principales presentó al menos un hallazgo, mientras que el 57,58 % no presentó hallazgos. Esta distribución describe la cobertura general de la evaluación; sin embargo, no debe interpretarse como el número de subpruebas ejecutadas, debido a que cada prueba principal agrupa varias verificaciones técnicas. La identificación de hallazgos permitió establecer la línea base para las fases posteriores de remediación y reevaluación de la metodología.

Estado de las vulnerabilidades identificadas

Para evaluar la efectividad de la metodología se analizaron los 14 hallazgos consolidados identificados en las pruebas principales. Después de aplicar las medidas de mitigación y efectuar la reevaluación, 11 se registraron como mitigados o resueltos, 1 como parcialmente mitigado y 2 permanecieron pendientes. Para el cálculo de los indicadores, únicamente se contabilizaron como corregidas las vulnerabilidades resueltas

completamente. La vulnerabilidad parcialmente mitigada se mantiene dentro de las vulnerabilidades persistentes hasta completar su corrección.

Estado	Cantidad	Porcentaje sobre hallazgos
Mitigadas / resueltas	11	78,6 %
Parcialmente mitigadas	1	7,1 %
Pendientes	2	14,3 %
Total de vulnerabilidades	14	100,00 %

Tabla 59 Estado Cantidad Porcentaje sobre hallazgos

Aplicación del método de análisis

Con base en el método establecido en la investigación, se utilizaron las vulnerabilidades identificadas inicialmente (Vi), las vulnerabilidades corregidas (Vc), las vulnerabilidades persistentes (Vp) y las vulnerabilidades finales (Vf). La vulnerabilidad parcialmente mitigada y las dos pendientes se consideran persistentes, debido a que todavía requieren acciones de remediación.

Tasa de mitigación (TM): $11 / 14 \times 100 = 78,6 \%$

Tasa de persistencia (TP): $3 / 14 \times 100 = 21,4 \%$

Reducción absoluta (RA): $14 - 3 = 11$ vulnerabilidades

Reducción relativa (RR): $(14 - 3) / 14 \times 100 = 78,6 \%$

Indicador	Resultado
Vulnerabilidades identificadas inicialmente (Vi)	14
Vulnerabilidades corregidas (Vc)	11
Vulnerabilidades persistentes (Vp)	2
Vulnerabilidades finales (Vf)	3
Tasa de mitigación (TM)	78,6 %
Tasa de persistencia (TP)	21,4 %
Reducción absoluta (RA)	11 vulnerabilidades

Indicador	Resultado
Reducción relativa (RR)	78.6 %

Tabla 60 Indicadores obtenidos

Interpretación de los resultados

La aplicación de la metodología permitió integrar en un solo análisis las pruebas realizadas sobre los componentes web y móvil del sistema ViTECED. De las 33 pruebas principales, 14 presentaron hallazgos y 19 no presentaron hallazgos. Debido a que cada prueba principal comprende diferentes subpruebas o verificaciones técnicas, esta frecuencia representa la clasificación general de las pruebas y no el número total de verificaciones efectuadas.

En la fase de remediación y reevaluación, 11 de los 14 hallazgos consolidados fueron corregidos completamente, 1 quedó parcialmente mitigado y 2 permanecieron pendientes. En consecuencia, la tasa de mitigación alcanzó el 78.6 %, mientras que la tasa de persistencia fue del 21,4 %.

La reducción absoluta fue de 11 vulnerabilidades, al pasar de 14 hallazgos iniciales a 3 vulnerabilidades que continúan presentes en algún grado. En términos relativos, la reducción fue del 78.6 %. Estos resultados evidencian una mejora cuantificable del estado de seguridad después de la aplicación de las medidas de mitigación.

Aunque la metodología produjo una reducción de los hallazgos, las tres vulnerabilidades persistentes dos pendientes y una parcialmente mitigada requieren continuar con el proceso de remediación y una nueva reevaluación. Por tanto, los resultados no implican la ausencia total de vulnerabilidades, sino que demuestran la efectividad alcanzada por las medidas implementadas dentro del alcance y versión del sistema evaluado.

CONCLUSIONES

Se identificó la arquitectura y los principales componentes del sistema ViTECED, incluyendo las aplicaciones web y móvil, APIs, servicios, bases de datos y mecanismos de comunicación. Esta caracterización permitió reconocer los activos críticos y delimitar la superficie de ataque sobre la cual se desarrolló la evaluación de seguridad.

Se diseñó una metodología de seguridad informática estructurada en cinco fases: recolección de información, identificación de vulnerabilidades, remediación, reevaluación y elaboración del informe. La integración de OWASP Testing Guide, OWASP MASVS, OWASP MASTG y OWASP SAMM permitió adaptar el proceso de evaluación a los diferentes componentes del sistema ViTECED y mantener la trazabilidad de los hallazgos desde su identificación hasta su reevaluación. Estas son precisamente las cinco fases establecidas en la metodología del proyecto.

La aplicación de la metodología permitió consolidar 33 pruebas principales de seguridad correspondientes a los componentes web y móvil del sistema ViTECED. Cada prueba principal comprendió diferentes subpruebas o verificaciones técnicas, permitiendo realizar un análisis más detallado de los aspectos de seguridad evaluados. Del total de pruebas principales, 14 presentaron uno o más hallazgos, equivalentes al 42,42 %, mientras que 19 no presentaron hallazgos, correspondientes al 57,58 %. Los resultados obtenidos permitieron establecer una línea base del estado de seguridad del sistema y determinar los hallazgos que requerían medidas de tratamiento.

Durante la fase de remediación se aplicaron medidas correctivas sobre las vulnerabilidades identificadas y posteriormente se realizó su reevaluación. De los 14 hallazgos consolidados, 11 fueron mitigados o resueltos completamente, 1 quedó parcialmente mitigado y 2 permanecieron pendientes. La vulnerabilidad parcialmente mitigada se consideró persistente debido a que todavía requiere acciones para completar su corrección. Estos resultados evidencian una reducción de las vulnerabilidades presentes en el sistema, aunque permanecen hallazgos que requieren continuar con el proceso de remediación y reevaluación.

El análisis estadístico comparativo entre el estado inicial y final determinó una tasa de mitigación del 78,6 % y una tasa de persistencia del 21,4 %. La reducción absoluta fue

de 11 vulnerabilidades y la reducción relativa del 78,6 %, al pasar de 14 hallazgos iniciales a 3 vulnerabilidades que continúan presentes en algún grado. En consecuencia, la aplicación de la metodología permitió cuantificar la efectividad de las medidas de seguridad implementadas y evidenciar una mejora en el estado de seguridad del sistema ViTECED, manteniéndose la necesidad de continuar el tratamiento de las vulnerabilidades pendientes y parcialmente mitigadas.

RECOMENDACIONES

Completar la remediación de las 3 vulnerabilidades pendientes y finalizar la corrección de la vulnerabilidad parcialmente mitigada, priorizando las acciones de acuerdo con su nivel de riesgo y el impacto potencial sobre la confidencialidad, integridad y disponibilidad de la información biomédica gestionada por el sistema ViTECED.

Ejecutar una nueva reevaluación de seguridad una vez implementadas las correcciones pendientes, utilizando las mismas pruebas, subpruebas y criterios de evaluación aplicados durante la evaluación inicial, con el propósito de comprobar técnicamente el cierre de los hallazgos y actualizar la tasa de mitigación, tasa de persistencia, reducción absoluta y reducción relativa.

Incorporar la metodología propuesta como parte del ciclo de desarrollo y mantenimiento del sistema ViTECED, realizando evaluaciones periódicas mediante OWASP Testing Guide, OWASP MASVS, OWASP MASTG y herramientas SAST y DAST, especialmente después de modificaciones en el código, actualizaciones de dependencias, cambios en la infraestructura o incorporación de nuevas funcionalidades. Esto mantiene coherencia con las herramientas y guías contempladas en la metodología desarrollada.

Fortalecer las prácticas de desarrollo seguro evaluadas mediante OWASP SAMM, estableciendo procedimientos para la gestión y seguimiento de vulnerabilidades, actualización de librerías y componentes, revisión de código, gestión de registros, respaldo de información y tratamiento de incidentes de seguridad.

Realizar una evaluación integral adicional antes del despliegue del sistema ViTECED en un entorno real, en un ambiente controlado que reproduzca las condiciones de producción,

verificando especialmente los mecanismos de autenticación y autorización, cifrado, protección de datos biomédicos, comunicaciones, APIs, aplicación móvil y mecanismos de recuperación ante fallos. Esta recomendación es consistente con el alcance del proyecto, que limita la evaluación realizada a un entorno de preproducción.

Mantener un registro centralizado y trazable de las vulnerabilidades identificadas en las subpruebas o verificaciones técnicas, incluyendo la prueba principal a la que pertenecen, evidencia obtenida, nivel de riesgo, categoría OWASP asociada, medida de mitigación aplicada, estado de corrección y resultado de la reevaluación. Este registro facilitará la comparación de futuras evaluaciones y permitirá determinar la evolución del nivel de seguridad del sistema ViTECED.

BIBLIOGRAFIA

- [1] E. Sánchez Duarte, “LAS TECNOLOGÍAS DE INFORMACIÓN Y COMUNICACIÓN (TIC) DESDE UNA PERSPECTIVA SOCIAL,” pp. 1409–1451, 2008.
- [2] С. О. Колб and М. І. Пирога, “Інформаційна безпека: нові виклики та стратегії захисту,” *Науковий вісник Ужгородського національного університету. Серія: Право*, vol. 5, no. 90, pp. 437–441, Oct. 2025, doi: 10.24144/2307-3322.2025.90.5.58.
- [3] A. Gite *et al.*, “Data Breaches: Financial and Reputational Impacts of Vulnerabilities on Organizations to Enhance Cybersecurity Strategies,” *3rd International Conference on Automation, Computing and Renewable Systems, ICACRS 2024 - Proceedings*, pp. 684–689, 2024, doi: 10.1109/ICACRS62842.2024.10841577.
- [4] “2024 3rd International Conference on Automation, Computing and Renewable Systems,” 2024.
- [5] “Aspectos destacados de las amenazas a las aplicaciones en 2024 | Prensa Releases | Digital.ai.” Accessed: May 04, 2026. [Online]. Available: <https://digital.ai/es/press-releases/digitalai-2024-application-security-threat-report/>
- [6] “Informe de seguridad de aplicaciones,” 2024.
- [7] S. (Sy) Banerjee, T. Hemphill, and P. Longstreet, “Wearable devices and healthcare: Data sharing and privacy,” *The Information Society*, vol. 34, no. 1, pp. 49–57, Jan. 2018, doi: 10.1080/01972243.2017.1391912.
- [8] H. N. Atalay and Ş. Yücel, “Decoding privacy concerns: the role of perceived risk and benefits in personal health data disclosure,” *Archives of Public Health*, vol. 82, no. 1, Dec. 2024, doi: 10.1186/S13690-024-01416-Z.
- [9] “Los ciberataques al sector sanitario crecen un 17 % en 2025 - Ciberseguridadtic.” Accessed: Apr. 26, 2026. [Online]. Available: <https://ciberseguridadtic.es/mercado/los-ciberataques-al-sector-sanitario-crecen-un-17-en-2025-check-point-2025091910245.htm>

- [10] “Aumentan un 17% los ciberataques contra el sector sanitario en 2025 - CyberSecurity News.” Accessed: Apr. 26, 2026. [Online]. Available: <https://cybersecuritynews.es/alerta-de-un-aumento-del-17-en-los-ciberataques-contra-el-sector-sanitario-en-2025/>
- [11] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, “A Survey on Security and Privacy Issues in Modern Healthcare Systems,” *ACM Trans. Comput. Healthc.*, vol. 2, no. 3, Jul. 2021, doi: 10.1145/3453176;PAGE:STRING:ARTICLE/CHAPTER.
- [12] A. Cervera García and A. Goussens, “Ciberseguridad y uso de las TIC en el Sector Salud,” *Aten. Primaria*, vol. 56, no. 3, p. 102854, Mar. 2024, doi: 10.1016/j.aprim.2023.102854.
- [13] “Panorama de las amenazas móviles en 2024 | Securelist.” Accessed: May 04, 2026. [Online]. Available: <https://securelist.lat/mobile-threat-report-2024/99586/>
- [14] “¿Qué es un ciberataque y los tipos de ataques en la red? | Fortinet.” Accessed: May 04, 2026. [Online]. Available: <https://www.fortinet.com/lat/resources/cyberglossary/types-of-cyber-attacks>
- [15] M. Elizabeth, A. V. Segura, M. Verónica, and T. Veloz, “Vulnerabilidades en aplicaciones web, amenazas y ataques,” *Revista Científica Multidisciplinar Generando*, vol. 5, no. 1, pp. 384-393–384-393, Jun. 2024, doi: 10.60100/RCMG.V5I1.201.
- [16] “Ataques a la cadena de suministro digital en ensayos clínicos.” Accessed: May 04, 2026. [Online]. Available: <https://www.esedsl.com/blog/ataques-a-la-cadena-de-suministro-digital-en-ensayos-clinicos>
- [17] <https://www.facebook.com/IURENOVUM>, “La vulnerabilidad de los datos médicos en el mundo digital - Iure Novum,” <https://iurenovum.com/>, Accessed: May 04, 2026. [Online]. Available: <https://iurenovum.com/la-vulnerabilidad-de-los-datos-medicos-en-el-mundo-digital/>
- [18] “A qué riesgos de privacidad pueden exponerte las aplicaciones de salud y fitness.” Accessed: May 04, 2026. [Online]. Available:

<https://www.welivesecurity.com/es/privacidad/riesgos-privacidad-app-salud-fitness/>

- [19] “Cristian Pillajo | LinkedIn.” Accessed: May 04, 2026. [Online]. Available: <https://www.linkedin.com/in/cristian-pillajo/?isSelfProfile=false>
- [20] C. Aydin, M. González Woge, and P. P. Verbeek, “Technological Environmentality: Conceptualizing Technology as a Mediating Milieu,” *Philos. Technol.*, vol. 32, no. 2, pp. 321–338, Jun. 2019, doi: 10.1007/S13347-018-0309-3/METRICS.
- [21] A. Cervera García and A. Goussens, “Ciberseguridad y uso de las TIC en el Sector Salud,” *Aten. Primaria*, vol. 56, no. 3, p. 102854, Mar. 2024, doi: 10.1016/J.APRIM.2023.102854.
- [22] A. I. Newaz, A. K. Sikder, M. A. Rahman, and A. S. Uluagac, “A Survey on Security and Privacy Issues in Modern Healthcare Systems,” *ACM Transactions on Computing for Healthcare*, vol. 2, no. 3, Jul. 2021, doi: 10.1145/3453176.
- [23] T. Hayajneh, B. J. Mohd, M. Imran, G. Almashaqbeh, and A. V. Vasilakos, “Secure Authentication for Remote Patient Monitoring with Wireless Medical Sensor Networks,” *Sensors 2016, Vol. 16, Page 424*, vol. 16, no. 4, p. 424, Mar. 2016, doi: 10.3390/S16040424.
- [24] M. Willing *et al.*, “Behavioral responses to a cyber attack in a hospital environment,” *Sci. Rep.*, vol. 11, no. 1, pp. 1–15, Dec. 2021, doi: 10.1038/S41598-021-98576-7;SUBJMETA.
- [25] K. Azbeg, O. Ouchetto, and S. Jai Andaloussi, “Access Control and Privacy-Preserving Blockchain-Based System for Diseases Management,” *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 4, pp. 1515–1527, Aug. 2023, doi: 10.1109/TCSS.2022.3186945.
- [26] M. M. Bayas, M. Alomoto Tomalá, M. Montaña Blacio, O. W. Gomez, and R. H. Rovira Jurado, “Comprehensive Home Monitoring System for Cardiovascular Health Management,” *Lecture Notes in Computer Science*, vol. 15891 LNCS, pp. 355–369, 2026, doi: 10.1007/978-3-031-97617-9_23.

- [27] P. Fernández Angela María, I. Comp Bayas Sampedro Marcia Marisol, and S. Elena, “Diseño de una plataforma informática para la monitorización de parámetros biomédicos.,” Jan. 23, 2025, *La Libertad: Universidad Estatal Península de Santa Elena*, 2025. Accessed: Aug. 26, 2025. [Online]. Available: <https://repositorio.upse.edu.ec/handle/46000/12824>
- [28] I. Comp Bayas Sampedro Marcia Marisol, S. Elena, and E. Año, “Desarrollo de una plataforma informática para el monitoreo remoto del electrocardiograma y detección de falsos positivos.,” Aug. 19, 2025, *La Libertad: Universidad Estatal Península de Santa Elena*, 2025. Accessed: Aug. 26, 2025. [Online]. Available: <https://repositorio.upse.edu.ec/handle/46000/14214>
- [29] C. M. Mejía-Granda, J. L. Fernández-Alemán, J. M. Carrillo-de-Gea, and J. A. García-Berná, “Security vulnerabilities in healthcare: an analysis of medical devices and software,” *Med. Biol. Eng. Comput.*, vol. 62, no. 1, pp. 257–273, Jan. 2024, doi: 10.1007/S11517-023-02912-0/FIGURES/5.
- [30] “Riesgos y vulnerabilidades del software sin parches | Splashtop.” Accessed: Apr. 15, 2026. [Online]. Available: <https://www.splashtop.com/es/blog/risks-and-vulnerabilities-of-unpatched-software>
- [31] L. Bracciale, P. Loreti, and G. Bianchi, “Cybersecurity vulnerability analysis of medical devices purchased by national health services,” *Sci. Rep.*, vol. 13, no. 1, pp. 1–12, Dec. 2023, doi: 10.1038/S41598-023-45927-1;SUBJMETA.
- [32] J. R. Tadhani, V. Vekariya, V. Sorathiya, S. Alshathri, and W. El-Shafai, “Securing web applications against XSS and SQLi attacks using a novel deep learning approach,” *Sci. Rep.*, vol. 14, no. 1, pp. 1–17, Dec. 2024, doi: 10.1038/S41598-023-48845-4;SUBJMETA=117,166,639,705;KWRD=COMPUTER+SCIENCE,ENGINEERING.
- [33] “¿Qué es el Cross Site Scripting (XSS)?” Accessed: Apr. 15, 2026. [Online]. Available: <https://www.sentinelone.com/es/cybersecurity-101/cybersecurity/cross-site-scripting/>

- [34] “¿Por qué realizar pruebas de seguridad y escalabilidad antes de lanzar un aplicativo?” Accessed: Apr. 15, 2026. [Online]. Available: <https://mtpinternational.mx/por-que-realizar-pruebas-de-seguridad-y-escalabilidad-antes-de-lanzar-un-aplicativo/>
- [35] W. Wardana, A. Almaarif, and A. Widjajarto, “Vulnerability Assessment and Penetration Testing On The Xyz Website Using Nist 800-115 Standard,” *Syntax Literate; Jurnal Ilmiah Indonesia*, vol. 7, no. 1, p. 520, Jan. 2022, doi: 10.36418/SYNTAX-LITERATE.V7I1.5800.
- [36] R. M. Wibowo and A. Sulaksono, “Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd,” *Indonesian Journal of Information Systems*, pp. 149–159, Feb. 2021, doi: 10.24002/IJIS.V3I2.4192.
- [37] “Security Assessment for Digital Wallet Payment Partner Applications using the OWASP Method: A Case Study in Indonesia,” *Journal of System and Management Sciences*, vol. 14, no. 3, Feb. 2024, doi: 10.33168/JSMS.2024.0319.
- [38] Á. S. C. Enríquez, J. V. H. Guijarro, and C. A. G. Cárdenas, “Pruebas de penetración para la seguridad informática al servidor web del laboratorio de ciberseguridad en la Universidad Politécnica Estatal del Carchi, 2021,” *SATHIRI*, vol. 17, no. 2, pp. 177–189, Jul. 2022, doi: 10.32645/13906925.1138.
- [39] F. De Sistemas, Y. Telecomunicaciones, C. De, T. De, and L. A. Información, “Análisis de seguridad controlado en aplicaciones web de una institución financiera utilizando herramientas de ciberseguridad y buenas prácticas de OWASP,” Oct. 12, 2022, *La Libertad: Universidad Estatal Península de Santa Elena*. 2022. Accessed: Sep. 20, 2025. [Online]. Available: <https://repositorio.upse.edu.ec/handle/46000/8646>
- [40] “Líneas.” Accessed: Aug. 26, 2025. [Online]. Available: <https://incyt.upse.edu.ec/index.php/investigacion/lineas>
- [41] S. Ghafur, S. Kristensen, K. Honeyford, G. Martin, A. Darzi, and P. Aylin, “A retrospective impact analysis of the WannaCry cyberattack on the NHS,” *NPJ Digit. Med.*, vol. 2, no. 1, Dec. 2019, doi: 10.1038/S41746-019-0161-6.

- [42] S. Liu *et al.*, “A hybrid method of recurrent neural network and graph neural network for next-period prescription prediction,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 12, pp. 2849–2856, Dec. 2020, doi: 10.1007/S13042-020-01155-X.
- [43] “Vista de Metodología OWASP: Un Enfoque para la Prevención y Resolución de Vulnerabilidades.” Accessed: Oct. 06, 2025. [Online]. Available: <https://ciencialatina.org/index.php/cienciala/article/view/16991/24390>
- [44] A. O. Adeniyi, J. O. Arowoogun, C. A. Okolo, R. Chidi, and O. Babawarun, “Ethical considerations in healthcare IT: A review of data privacy and patient consent issues,” <https://wjarr.com/sites/default/files/WJARR-2024-0593.pdf>, vol. 21, no. 2, pp. 1660–1668, Feb. 2024, doi: 10.30574/WJARR.2024.21.2.0593.
- [45] C. Thapa and S. Camtepe, “Precision health data: Requirements, challenges and existing techniques for data security and privacy,” *Comput. Biol. Med.*, vol. 129, p. 104130, Feb. 2021, doi: 10.1016/J.COMPBIOMED.2020.104130.
- [46] “Plan Nacional de Desarrollo Ecuador No Se Detiene 2025 – 2029 – Secretaría Nacional de la Administración Pública y Planificación.” Accessed: Sep. 15, 2025. [Online]. Available: <https://www.planificacion.gob.ec/plan-nacional-de-desarrollo-2025-2029-ecuador-no-se-detiene/>
- [47] H. Nassaji, “Qualitative and descriptive research: Data type versus data analysis,” *Language Teaching Research*, vol. 19, no. 2, pp. 129–132, Mar. 2015, doi: 10.1177/1362168815572747;ISSUE:ISSUE:DOI.
- [48] A. Muse and J. M. Baldwin, “Quasi-Experimental Research Design,” *The Encyclopedia of Research Methods in Criminology and Criminal Justice: Volume II: Parts 5-8*, pp. 307–310, Jan. 2021, doi: 10.1002/9781119111931.CH61;SUBPAGE:STRING:ACCESS.
- [49] R. Heale and A. Twycross, “What is a case study?,” *Evid. Based. Nurs.*, vol. 21, no. 1, pp. 7–8, Jan. 2018, doi: 10.1136/EB-2017-102845.
- [50] E. R. Tanjung and M. Albina, “Penelitian Deskriptif dalam Pendidikan,” *BLAZE : Jurnal Bahasa dan Sastra dalam Pendidikan Linguistik*

- dan Pengembangan*, vol. 3, no. 3, pp. 168–176, Jun. 2025, doi: 10.59841/BLAZE.V3I3.2972.
- [51] E. T. Koh and W. L. Owen, “Descriptive Research and Qualitative Research,” *Introduction to Nutrition and Health Research*, pp. 219–248, 2000, doi: 10.1007/978-1-4615-1401-5_12.
- [52] M. L. Maciejewski, “Quasi-experimental design,” *Biostat. Epidemiol.*, vol. 4, no. 1, pp. 38–47, Jan. 2020, doi: 10.1080/24709360.2018.1477468.
- [53] C. J. Miller, S. N. Smith, and M. Pugatch, “Experimental and quasi-experimental designs in implementation research,” *Psychiatry Res.*, vol. 283, p. 112452, Jan. 2020, doi: 10.1016/J.PSYCHRES.2019.06.027.
- [54] A. Ratnasari and I. Sudradjat, “Case study approach in post-occupancy evaluation research,” *ARTEKS: Jurnal Teknik Arsitektur*, vol. 8, no. 3, pp. 427–434, Dec. 2023, doi: 10.30822/ARTEKS.V8I3.2584.
- [55] D. Fucci, E. Alégroth, M. Felderer, and C. Johannesson, “Evaluating software security maturity using OWASP SAMM: Different approaches and stakeholders perceptions,” *Journal of Systems and Software*, vol. 214, p. 112062, Aug. 2024, doi: 10.1016/J.JSS.2024.112062.
- [56] “¿Qué es NIST Cibersecurity Framework? | GSS.” Accessed: Oct. 23, 2025. [Online]. Available: <https://www.globalsuitesolutions.com/es/que-es-nist-cibersecurity-framework/>
- [57] H. K. Mohajan, “Quantitative Research: A Successful Investigation in Natural and Social Sciences,” *Journal of Economic Development, Environment and People*, vol. 9, no. 4, pp. 50–79, Dec. 2020, doi: 10.26458/JEDEP.V9I4.679.
- [58] W. M. Lim, “What Is Quantitative Research? An Overview and Guidelines,” *Australasian Marketing Journal*, Aug. 2025, doi: 10.1177/14413582241264622.
- [59] M. R. Fadli, “Memahami desain metode penelitian kualitatif,” *HUMANIKA*, vol. 21, no. 1, pp. 33–54, Apr. 2021, doi: 10.21831/HUM.V21I1.38075.
- [60] R. Dawn, “Quantitative and Qualitative Analysis,” pp. 95–110, 2018, doi: 10.1007/978-981-10-6644-3_4.

- [61] P. A. Kovačić, “Scientific method as the foundation of scientific research,” *International Review*, no. 1–2, pp. 13–17, 2021, doi: 10.5937/INTREV2102013K.
- [62] I. Etikan, S. A. Musa, and R. S. Alkassim, “Comparison of Convenience Sampling and Purposive Sampling,” *American Journal of Theoretical and Applied Statistics* 2016, Volume 5, Page 1, vol. 5, no. 1, pp. 1–4, Dec. 2015, doi: 10.11648/J.AJTAS.20160501.11.
- [63] P. Bhardwaj, “Types of sampling in research,” *Journal of the Practice of Cardiovascular Sciences*, vol. 5, no. 3, p. 157, 2019, doi: 10.4103/JPCS.JPCS_62_19.
- [64] “Guía de pruebas de seguridad web de OWASP | Fundación OWASP.” Accessed: Oct. 07, 2025. [Online]. Available: <https://owasp.org/www-project-web-security-testing-guide/>
- [65] “¿Qué es la seguridad informática y cómo implementarla?” Accessed: Oct. 13, 2025. [Online]. Available: <https://www.grupocibernos.com/blog/que-es-la-seguridad-informatica-y-como-implementarla>
- [66] “Seguridad de la Información – Seguridad de la Información | Redes de Voz y Datos | Diseño de Proyectos.” Accessed: May 18, 2026. [Online]. Available: <https://www.4itn.mx/ciberseguridad/>
- [67] “Principios Básicos de Seguridad de la Información: Confidencialidad, Integridad y Disponibilidad (CIA).” Accessed: Oct. 13, 2025. [Online]. Available: <https://www.auditool.org/blog/auditoria-de-ti/principios-basicos-de-seguridad-de-la-informacion-confidencialidad-integridad-y-disponibilidad-cia>
- [68] M. Genesis, S. Rodríguez, N. J. Quimis Peñafiel, Y. Paulette Gonzalez Baque, J. Mario, and M. Marcillo, “CALIDAD DEL SOFTWARE APLICADO A LA EVALUACIÓN DE LA SEGURIDAD EN SISTEMAS DE INFORMACIÓN,” *UNESUM - Ciencias. Revista Científica Multidisciplinaria*, vol. 5, no. 4, pp. 23–30, Jun. 2021, doi: 10.47230/UNESUM-CIENCIAS.V5.N4.2021.411.
- [69] J. Liu, Y. Zhang, S. Zhou, J. Yang, Y. Lu, and X. Zhong, “Autonomous penetration testing using reinforcement learning: A review and perspectives,” *Expert Syst. Appl.*, vol. 300, Mar. 2026, doi: 10.1016/j.eswa.2025.130219.

- [70] M. M. Ahmed, Y. Souissi, O. Trabelsi, S. Guilley, A. Bouvet, and S. Takarabt, "Towards a Black-Box Security Evaluation Framework," *Communications in Computer and Information Science*, vol. 1497 CCIS, pp. 79–92, 2021, doi: 10.1007/978-3-030-90553-8_6.
- [71] H. ; Zhang *et al.*, "ICVTest: A Practical Black-Box Penetration Testing Framework for Evaluating Cybersecurity of Intelligent Connected Vehicles," *Applied Sciences* 2024, Vol. 14, Page 204, vol. 14, no. 1, p. 204, Dec. 2023, doi: 10.3390/APP14010204.
- [72] N. De *et al.*, "Maestría en Trabajo previo a la obtención de título de Magister en Ciberseguridad Ciberseguridad TUTOR/ES".
- [73] M. E. Khan and F. Khan, "A Comparative Study of White Box, Black Box and Grey Box Testing Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 3, no. 6, Jul. 2012, doi: 10.14569/IJACSA.2012.030603.
- [74] V. M. KOZEL, IE. A. DROZDOVA, O. I. IVANCHUK, and O. O. PRYKHODKO, "ДОСЛІДЖЕННЯ МЕТОДІВ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ," *Вісник Херсонського національного технічного університету*, vol. 3, no. 3(90), pp. 221–227, Nov. 2024, doi: 10.35546/KNTU2078-4481.2024.3.28.
- [75] D. Fucci, E. Alégroth, M. Felderer, and C. Johannesson, "Evaluating software security maturity using OWASP SAMM: Different approaches and stakeholders perceptions," *Journal of Systems and Software*, vol. 214, p. 112062, Aug. 2024, doi: 10.1016/J.JSS.2024.112062.
- [76] R. A. Khan, S. U. Khan, M. Alzahrani, and M. Ilyas, "Security Assurance Model of Software Development for Global Software Development Vendors," *IEEE Access*, vol. 10, pp. 58458–58487, 2022, doi: 10.1109/ACCESS.2022.3178301.
- [77] J. Tu, Z. Wang, W. Li, L. Tan, L. Huang, and F. Liu, "SAMM: A general-purpose segmentation model for material micrographs based on the segment anything model 2," *Advanced Powder Materials*, vol. 5, no. 4, p. 100404, Aug. 2026, doi: 10.1016/J.APMATE.2026.100404.

- [78] “OWASP SAMM | Fundación OWASP.” Accessed: Jul. 21, 2026. [Online]. Available: <https://owasp.org/www-project-samm/>
- [79] J. M. B. Ontiveros, N. R. R. Morales, M. P. Reyes, C. A. Barrios, and S. A. C. Cardenas, “Metodología OWASP: Un Enfoque para la Prevención y Resolución de Vulnerabilidades,” *Ciencia Latina Revista Científica Multidisciplinar*, vol. 9, no. 2, pp. 1558–1576, Apr. 2025, doi: 10.37811/CL_RCM.V9I2.16991.
- [80] F. Hilario, D. Chang, C. Zafra, Y. Vasquez, and L. Chipana, “Application of the OWASP Framework to Identify and Remediate Vulnerabilities in Java Web Applications,” *Online) Journal of System and Management Sciences*, vol. 14, no. 7, pp. 406–425, 2024, doi: 10.33168/JSMS.2024.0722.
- [81] A. H. Jeronimo, P. M. Moreno, J. A. V. Camacho, and G. C. Vega, “Techniques of SAST Tools in the Early Stages of Secure Software Development: A Systematic Literature Review,” *2024 IEEE International Conference on Engineering Veracruz, ICEV 2024*, 2024, doi: 10.1109/ICEV63254.2024.10766004.
- [82] M. Y. Darus, M. Farhan Bin Bolhan, A. Kurniawan, Y. Muliono, C. R. Pardomuan, and M. Mohamad Hata, “Enhancing Web Application Penetration Testing with a Static Application Security Testing (SAST) Tool,” *8th International Conference on Recent Advances and Innovations in Engineering: Empowering Computing, Analytics, and Engineering Through Digital Innovation, ICRAIE 2023*, 2023, doi: 10.1109/ICRAIE59459.2023.10468317.
- [83] R. Singh, M. Kumar Gupta, D. R. Patil, and S. Maruti Patil, “Analysis of Web Application Vulnerabilities using Dynamic Application Security Testing,” *2024 IEEE 9th International Conference for Convergence in Technology, I2CT 2024*, 2024, doi: 10.1109/I2CT61223.2024.10543484.
- [84] F. M. Tudela, J. R. B. Higuera, J. B. Higuera, J. A. S. Montalvo, and M. I. Argyros, “On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications,” *Applied Sciences 2020, Vol. 10, Page 9119*, vol. 10, no. 24, p. 9119, Dec. 2020, doi: 10.3390/APP10249119.

- [85] S. Qadir, E. Waheed, A. Khanum, and S. Jehan, "Comparative evaluation of approaches & tools for effective security testing of Web applications," *PeerJ Comput. Sci.*, vol. 11, pp. 1–42, Apr. 2025, doi: 10.7717/PEERJ-CS.2821/TABLE-21.
- [86] "Qué es Kali Linux y para qué sirve." Accessed: Apr. 18, 2026. [Online]. Available: <https://www.campusciberseguridad.com/blog/que-es-kali-linux-y-para-que-sirve/>
- [87] "Kali Linux: qué es y para que se utiliza en ciberseguridad - DigitechFP." Accessed: Apr. 18, 2026. [Online]. Available: <https://digitechfp.com/blog/ciberseguridad/kali-linux-que-es-y-para-que-se-utiliza-en-ciberseguridad/>
- [88] "DuckDuckGo: Qué es y cómo protege tu privacidad al buscar online." Accessed: Apr. 18, 2026. [Online]. Available: <https://www.godaddy.com/resources/latam/desarrollo/duckduckgo-privacidad-online>
- [89] "DuckDuckGo: Qué es, cómo funciona y por qué proteger tu privacidad." Accessed: Apr. 18, 2026. [Online]. Available: <https://www.polimetro.com/que-es-duckduckgo/>
- [90] "Bing vs. Google: Comparación de los dos motores de búsqueda." Accessed: Apr. 18, 2026. [Online]. Available: <https://es.semrush.com/blog/comparacion-bing-vs-google/>
- [91] "Cómo proporciona Bing los resultados de la búsqueda | Microsoft Support." Accessed: Apr. 18, 2026. [Online]. Available: <https://support.microsoft.com/es-es/bing/how-bing-delivers-search-results>
- [92] "Google hacking: averigua cuanta información sobre ti o tu empresa aparece en los resultados." Accessed: Apr. 18, 2026. [Online]. Available: <https://www.welivesecurity.com/es/recursos-herramientas/google-hacking-averigua-informacion-aparece-resultados/>

- [93] “Google Hacking Database: guía detallada con más de 30 ejemplos de Google dorks.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.securitylab.lat/blog/personal/HackerES/357496.php>
- [94] “¿Qué es un comando Curl y cómo usarlo?” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.hostinger.com/es/tutoriales/comando-curl>
- [95] “Verifica redirecciones 301 desde tu ordenador en 2021 - SEOCOM - Agencia SEO y Marketing Digital.” Accessed: Apr. 18, 2026. [Online]. Available: <https://seocom.agency/blog/verifica-redirecciones-301-desde-la-consola-de-tu-ordenador/>
- [96] “¿Qué es WhatWeb? | KeepCoding Bootcamps.” Accessed: Apr. 18, 2026. [Online]. Available: <https://keepcoding.io/blog/que-es-whatweb/>
- [97] “WhatWeb: Escáner Web de Nueva Generación » EsGeeks.” Accessed: Apr. 18, 2026. [Online]. Available: https://esgeeks.com/whatweb-escaner-web-nueva-generacion/#google_vignette
- [98] “Qués y Cómo usar el Comando Wget Linux + 12 Ejemplpos.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.hostinger.com/mx/tutoriales/usar-comando-wget>
- [99] “Wget, la herramienta para descargar páginas web en Linux.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.redeszone.net/tutoriales/servidores/wget-herramienta-descargar-web-linux/>
- [100] “Burp Suite: Herramienta Esencial para la Seguridad de Aplicaciones Web - Ciberseguridad.” Accessed: Apr. 18, 2026. [Online]. Available: <https://ciberseguridad.art/2024/11/19/burp-suite-herramienta-esencial-para-la-seguridad-de-aplicaciones-web/>
- [101] “¿Qué es Burp Suite? - Tarlogic security.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.tarlogic.com/es/glosario-ciberseguridad/burp-suite/>
- [102] “Burp Suite Community Edition para Windows - Descarga gratis en Uptodown.” Accessed: Apr. 18, 2026. [Online]. Available: <https://burp-suite-community-edition.uptodown.com/windows>

- [103] “El poder de Wfuzz - Territorio Hacker 2024.” Accessed: Apr. 18, 2026. [Online]. Available: <https://territoriohacker.com/herramientas-de-pentesting-3-wfuzz/>
- [104] “F1r0x | Blog sobre Ciberseguridad.” Accessed: Apr. 18, 2026. [Online]. Available: <https://f1r0x.github.io/Web/wfuzz.html>
- [105] “NMAP - Escaneo de puertos | Ciberseguridad.” Accessed: Apr. 18, 2026. [Online]. Available: <https://albertoestrada.es/hacking/nmap-escaneo-de-puertos/>
- [106] “Escanear puertos con Nmap - Listado de comandos Nmap.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.redeszone.net/tutoriales/configuracion-puertos/nmap-escanear-puertos-comandos/>
- [107] “Recopilación de información · notas-de-seguridad-informatica-ofensiva.” Accessed: Apr. 18, 2026. [Online]. Available: <https://epasan.gitbooks.io/notas-de-seguridad-informatica-ofensiva/content/chapter1.html>
- [108] “Extensión Netcraft Chrome: Análisis de hosting y seguridad web.” Accessed: Apr. 18, 2026. [Online]. Available: <https://raiolanetworks.com/ayuda/netcraft-chrome-extension/>
- [109] “Herramientas de propiedad intelectual para pruebas de seguridad y redes | HackerTarget.com.” Accessed: Apr. 18, 2026. [Online]. Available: <https://hackertarget.com/ip-tools/>
- [110] “HackerTarget – Kit de herramientas e inteligencia de red para ayudar a prevenir ataques - HackWise.” Accessed: Apr. 18, 2026. [Online]. Available: <https://hackwise.mx/hackertarget-conjunto-de-herramientas/>
- [111] “ViewDNS - Osintux.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.osintux.org/documentacion/viewdns>
- [112] “Herramienta de revisión automatizada de código y solución de software de análisis | Sonar.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.sonarsource.com/es/solutions/code-review/>
- [113] “Refactorización de código con SonarQube para mejorar la calidad - Sentrío.” Accessed: Apr. 18, 2026. [Online]. Available: <https://sentrío.io/blog/refactorizacion-de-codigo-con-sonarqube/>

- [114] “Escaneo de vulnerabilidades automático con OWASP ZAP.” Accessed: May 28, 2026. [Online]. Available: <https://academy.seguridadcero.com.pe/blog/escaneo-vulnerabilidades-autom%C3%A1tico-OWASP-ZAP>
- [115] “OWASP ZAP: El aliado en seguridad web que necesitas | CyScope.” Accessed: Apr. 18, 2026. [Online]. Available: <https://cyscope.io/es/owasp-zap-el-aliado-en-seguridad-web-que-necesitas/>
- [116] “Escaneo de vulnerabilidades automático con OWASP ZAP.” Accessed: Apr. 18, 2026. [Online]. Available: <https://academy.seguridadcero.com.pe/blog/escaneo-vulnerabilidades-autom%C3%A1tico-OWASP-ZAP>
- [117] “Descubre con qué tecnologías se construyen los sitios web: Wappalyzer.” Accessed: Apr. 18, 2026. [Online]. Available: <https://www.wappalyzer.com/>
- [118] “gobuster | Herramientas de Kali Linux.” Accessed: May 28, 2026. [Online]. Available: <https://www.kali.org/tools/gobuster/>
- [119] “GoBuster | Ciberseguridad.” Accessed: Apr. 18, 2026. [Online]. Available: <https://ciberseguridad.alberlome.com/herramientas/fuzzing-extraer-directorios-ocultos/gobuster>
- [120] “Nikto: un práctico escáner de vulnerabilidades de sitios web – Ciberseguridad.” Accessed: May 28, 2026. [Online]. Available: <https://ciberseguridad.com/herramientas/software/nikto/>
- [121] “Conceptos Básicos de Nikto - Técnicas de escaneo de Servidores y Aplicaciones Web - The Hacker Way.” Accessed: Apr. 18, 2026. [Online]. Available: <https://thehackerway.es/2011/05/12/conceptos-basicos-de-nikto-tecnicas-de-escaneo-de-servidores-y-aplicaciones-web/>
- [122] “Enumeración por fuerza bruta con gobuster - Byte Mind.” Accessed: Apr. 18, 2026. [Online]. Available: <https://byte-mind.net/enumeracion-por-fuerza-bruta-con-gobuster/>
- [123] “Gobuster: descubrimiento de directorios y archivos ocultos | Ciberseguridad.” Accessed: Apr. 18, 2026. [Online]. Available: <https://albertoestrada.es/hacking/gobuster-fuzzing-web/>

- [124] Cooksey RW. Estadísticas descriptivas para resumir datos. Ilustrando procedimientos estadísticos: encontrar significado en los datos cuantitativos. 2020 mayo 15:61–139. doi: 10.1007/978-981-15-2537-7_5. PMCID: PMC7221239.
- [125] J. P. Jones, L. Farrell, and L. M. Benson, Informational Scientific Primers for Officers of the Court: Intended to Strengthen Use of Forensic Science Evidence, NIST Special Publication 1500-28. Gaithersburg, MD, USA: National Institute of Standards and Technology, Mar. 2025, doi: 10.6028/NIST.SP.1500-28
- [126] J. E. A. Chang and <https://dialnet.unirioja.es/servlet/autor?codigo=4698282>, “Análisis de ataques cibernéticos hacia el Ecuador,” *Revista Científica Aristas*, ISSN-e 2600-5662, Vol. 2, N°. 1, 2020 (*Ejemplar dedicado a: Edición Mayo 2020*), págs. 18-27, vol. 2, no. 1, pp. 18–27, 2020, Accessed: Apr. 20, 2026. [Online]. Available: <https://dialnet.unirioja.es/servlet/articulo?codigo=10322885&info=resumen&idoma=ENG>
- [127] “MAPA | Mapa interactivo de ciberamenazas de Kaspersky.” Accessed: Apr. 20, 2026. [Online]. Available: <https://cybermap.kaspersky.com/>
- [128] “Ecuador es el tercer país con las mayores amenazas cibernéticas en América Latina, según Check Point | Ecuador | Noticias | El Universo.” Accessed: Apr. 20, 2026. [Online]. Available: <https://www.eluniverso.com/noticias/ecuador/ecuador-enfrenta-un-creciente-riesgo-de-ciberataques-en-el-2024-nota/>
- [129] A. Cervera García and A. Goussens, “Cybersecurity and use of ICT in the health sector,” *Aten. Primaria*, vol. 56, no. 3, p. 102854, Mar. 2024, doi: 10.1016/J.APRIM.2023.102854.
- [130] H. T. Neprash *et al.*, “Trends in Ransomware Attacks on US Hospitals, Clinics, and Other Health Care Delivery Organizations, 2016-2021,” *JAMA Health Forum*, vol. 3, no. 12, pp. e224873–e224873, Dec. 2022, doi: 10.1001/JAMAHEALTHFORUM.2022.4873.
- [131] “Informática: prevención del phishing en el sector sanitario,” *Revista Electrónica de PortalesMedicos.com*, May 2020.

- [132] A. Ghubaish, T. Salman, M. Zolanvari, D. Unal, A. Al-Ali, and R. Jain, “Recent Advances in the Internet-of-Medical-Things (IoMT) Systems Security,” *IEEE Internet Things J.*, vol. 8, no. 11, pp. 8707–8718, Jun. 2021, doi: 10.1109/JIOT.2020.3045653.
- [133] S. Ray, K. N. Mishra, and S. Dutta, “Detection and prevention of DDoS attacks on M-healthcare sensitive data: a novel approach,” *International Journal of Information Technology 2022 14:3*, vol. 14, no. 3, pp. 1333–1341, Feb. 2022, doi: 10.1007/S41870-022-00869-1.
- [134] D. P. P. Sajan, “A Comprehensive Analysis of WannaCry Ransomware,” *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, vol. 08, no. 008, pp. 1–5, Aug. 2024, doi: 10.55041/IJSREM37242.
- [135] L. J. Trautman and P. Ormerod, “Wannacry, Ransomware, and the Emerging Threat to Corporations,” *SSRN Electronic Journal*, Aug. 2018, doi: 10.2139/SSRN.3238293.
- [136] J. M. Ehrenfeld, “WannaCry, Cybersecurity and Health Information Technology: A Time to Act,” *J. Med. Syst.*, vol. 41, no. 7, pp. 104–, Jul. 2017, doi: 10.1007/S10916-017-0752-1/METRICS.
- [137] Y. Deng, A. Lambrecht, and C. Tucker, “Asymmetric Consequences of Cyber-Vulnerability on Health Services,” 2020, Accessed: Apr. 20, 2026. [Online]. Available: <https://ssrn.com/abstract=3642485>
- [138] “¿Cuáles son los riesgos de seguridad de las aplicaciones? - OWASP Top 10:2025.” Accessed: Apr. 26, 2026. [Online]. Available: https://owasp.org/Top10/2025/0x02_2025-What_are_Application_Security_Risks/
- [139] “Las vulnerabilidades de los sistemas informativos sanitarios.” Accessed: Apr. 26, 2026. [Online]. Available: https://www.redseguridad.com/sectores/sanitario/las-vulnerabilidades-de-los-sistemas-sanitarios-siguen-siendo-un-gran-reto-para-la-ciberseguridad_20241209.html

- [140] A. Cervera García and A. Goussens, “Ciberseguridad y uso de las TIC en el Sector Salud,” *Aten. Primaria*, vol. 56, no. 3, p. 102854, Mar. 2024, doi: 10.1016/j.aprim.2023.102854.
- [141] P. Ewoh and T. Vartiainen, “Vulnerability to Cyberattacks and Sociotechnical Solutions for Health Care Systems: Systematic Review,” *J. Med. Internet Res.*, vol. 26, no. 1, p. e46904, May 2024, doi: 10.2196/46904.
- [142] D. Giansanti, “Cybersecurity and the digital-health: The challenge of this millennium,” *Healthcare (Switzerland)*, vol. 9, no. 1, 2021, doi: 10.3390/HEALTHCARE9010062.
- [143] “▷ Seguridad informática hospitalaria - Ocronos - Editorial Científico-Técnica.” Accessed: Apr. 26, 2026. [Online]. Available: <https://revistamedica.com/seguridad-informatica-hospitalaria/>
- [144] “Ciberataques a infraestructuras críticas: vulnerabilidades en Hospitales | Ciberseguridad.” Accessed: Apr. 26, 2026. [Online]. Available: <https://www.iniseg.es/blog/ciberseguridad/ciberataques-a-infraestructuras-criticas-vulnerabilidades-en-hospitales/>
- [145] J. P. S. Chalco, M. A. C. Molina, <https://dialnet.unirioja.es/servlet/autor?codigo=5675956>, and <https://dialnet.unirioja.es/servlet/autor?codigo=4447901>, “Detección de vulnerabilidades informáticas en estaciones de trabajo: Caso de estudio Hospital de Especialidades José Carrasco Arteaga,” *Polo del Conocimiento: Revista científico - profesional*, ISSN-e 2550-682X, Vol. 7, N°. 4 (ABRIL 2022), 2022, vol. 7, no. 4, p. 73, 2022, doi: 10.23857/pc.v7i4.3835.
- [146] M. Panagioti *et al.*, “Prevalence, severity, and nature of preventable patient harm across medical care settings: Systematic review and meta-analysis,” *The BMJ*, vol. 366, 2019, doi: 10.1136/BMJ.L4185.
- [147] “The economics of patient safety in primary and ambulatory care,” vol. 106, Nov. 2018, doi: 10.1787/BAF425AD-EN.
- [148] B. Riskhan, M. A. Ullah Sheikh, M. S. Hossain, K. Hussain, Z. Zainol, and N. Z. Jhanjh, “Major Vulnerabilities of Web Application in Real World Scenarios and

- Their Prevention,” *ICoICC 2025 - 3rd International Conference on Intelligent and Cloud Computing*, 2025, doi: 10.1109/ICOICC64033.2025.11052016.
- [149] F. Faisal Fadlalla and H. T. Elshoush, “Input Validation Vulnerabilities in Web Applications: Systematic Review, Classification, and Analysis of the Current State-of-the-Art,” *IEEE Access*, vol. 11, pp. 40128–40161, 2023, doi: 10.1109/ACCESS.2023.3266385.
- [150] A. K. Priyanka and S. Sai Smruthi, “Web Application Vulnerabilities: Exploitation and Prevention,” *Proceedings - ICOECS 2020: 2020 International Conference on Electrotechnical Complexes and Systems*, Oct. 2020, doi: 10.1109/ICOECS50468.2020.9278437.
- [151] I. Luján *et al.*, “MOBILE APPLICATIONS AND THEIR IMPORTANCE IN THE COMMERCIAL WORLD,” *Revista Género e Interdisciplinaridade*, vol. 4, no. 05, pp. 797–811, Nov. 2023, doi: 10.51249/gei.v4i05.1705.
- [152] R. Kumar, “Vulnerability Assessment of Mobile Applications,” *International Journal of Innovative Science and Research Technology (IJISRT)*, pp. 900–905, Aug. 2024, doi: 10.38124/IJISRT/IJISRT24AUG464.
- [153] “OWASP Top 10:2021.” Accessed: Apr. 19, 2026. [Online]. Available: <https://owasp.org/Top10/2021/>
- [154] “Los 10 principales problemas de OWASP en dispositivos móviles | Fundación OWASP.” Accessed: Apr. 19, 2026. [Online]. Available: <https://owasp.org/www-project-mobile-top-10/>
- [155] “Estándar de Verificación de Seguridad de Aplicaciones (ASVS) de OWASP | Fundación OWASP.” Accessed: Oct. 14, 2025. [Online]. Available: <https://owasp.org/www-project-application-security-verification-standard/>
- [156] “Comprensión del estándar de verificación de seguridad de aplicaciones OWASP | SecureFlag.” Accessed: Oct. 14, 2025. [Online]. Available: <https://blog.secureflag.com/2024/11/15/understanding-owasp-asvs/>
- [157] “Entendiendo el top 10 de OWASP: Cómo usarlo como estándar.” Accessed: Oct. 14, 2025. [Online]. Available: <https://beaglesecurity.substack.com/p/understanding-owasp-top-10-how-to>

- [158] M. B. Sampedro *et al.*, “Sistema de monitoreo de parámetros cardiovasculares basado en IoT y MQTT para alertas médicas,” *Ingenius: Revista de Ciencia y Tecnología*, ISSN 1390-650X, ISSN-e 1390-860X, N°. 34, 2025 (Ejemplar dedicado a: july-december), págs. 75-88, no. 34, pp. 75–88, 2025, doi: 10.17163/ings.n34.2025.06.
- [159] “Metodología de evaluación de riesgos de OWASP | Fundación OWASP.” Accessed: Jul. 22, 2026. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology
- [160] “La metodología de calificación de riesgos de OWASP y SimpleRisk.” Accessed: Jul. 22, 2026. [Online]. Available: <https://www.simplerisk.com/blog/owasp-risk-rating-methodology-and-simplerisk>
- [161] “OWASP Risk Rating Methodology | OWASP Foundation.” Accessed: Jul. 22, 2026. [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

ANEXOS

Anexo 1. Formato de entrevista a los colaboradores del proyecto “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas”.

Entrevista dirigida a: A los colaboradores del proyecto “Investigación sobre aplicaciones IoT en la adquisición de señales bioeléctricas”.	
Objetivo: Identificar el nivel de proceso sobre la situación actual de la seguridad del proyecto de Sistema de Monitoreo Biomédico.	
1	¿Qué controles de autenticación están actualmente implementados en el sistema para garantizar que solo personal autorizado acceda al sistema?
R1	Mediante tokens de autenticación
2	¿Existen mecanismos de autenticación diferenciada (roles de usuario: médicos, técnicos, administradores)? ¿Como se gestionan estos accesos?
R2	Si, los accesos se gestionan inicialmente con el rol, dentro del sistema está estipulado que rol tiene acceso a que rutas.
3	¿Se aplican políticas de contraseñas seguras o autenticación multifactor (MFA)?
R3	Existe la política de contraseña segura pero no autenticación multifactor
4	¿El sistema protege los datos biomédicos durante la transmisión? ¿se utiliza cifrado (TLS/SSL, VPN u otro)?
R4	Si, se usa el protocolo MQTT y TLS/SSL
5	¿Cómo se resguarda la información almacenada localmente o en servidores? ¿Se emplea cifrado en reposo?
R5	Se almacena en servidores, los datos sensibles dentro de la BD esta cifrados
6	¿Existen mecanismos para evitar fugas de datos sensibles (Data Leakage Prevention)?
R6	No

7	¿Se han considerado las vulnerabilidades críticas del OWASP Top 10 (inyecciones SQL, XSS, CSRF, fallos de autenticación, configuración insegura en el desarrollo de la aplicación)?
R7	Solo inyección SQL.
8	¿Se realiza validación de entradas y sanitización de datos en los formularios y APIs?
R8	Si, se aplican los procesos
9	¿Se utilizan librerías y dependencias actualizadas y verificadas frente a vulnerabilidades conocidas?
R9	No, hay muchas bibliotecas desactualizadas
10	¿Se realiza pruebas de penetración o análisis de vulnerabilidades de forma periódica? ¿Qué resultados se han obtenido?
R10	No
11	¿Qué mecanismos de respaldo o continuidad del servicio existen en caso de un fallo o ataques al sistema?
R11	Existen bloqueos por IP para tráfico maliciosos
12	¿Se han realizado pruebas de seguridad estática (SAST) al código fuente de la aplicación? ¿Qué resultados se han obtenido?
R12	No
13	¿Se han ejecutado pruebas dinámicas (DAST) mientras la aplicación está en funcionamiento? ¿Qué fallos se detectaron?
R13	No
14	¿Se cuenta con mecanismos de monitoreo de accesos y generación de registros (logs) para auditoría de seguridad?
R14	Si, hay un sistema de auditoría que registra en una base de datos MongoDB los logs de actividad del usuario.
15	¿Existen sistemas de detección y alerta temprana frente a intentos de intrusión o uso indebido?

R15	No
16	¿Qué limitaciones técnicas o de recursos han impedido implementar prácticas de seguridad más robustas?
R16	Falta de profesionalismo con experiencia en el área.
17	¿Qué funciones de seguridad previstas en la fase de diseño aún no han sido implementadas?
R17	Ninguna

Anexo 2. Registro de la técnica de observación aplicada en el proyecto de Sistema de monitoreo biomédico.

Registro descriptivo de la Información
Objetivo: Identificar y comprender los posibles problemas o fallos que puedan estar ocurriendo en el Sistema de Monitoreo Biométrico.
Hechos Observados
• El acceso al sistema solo se realiza mediante usuario y contraseña, y aunque existen roles por tipo de usuario, no se aplica autenticación multifactor ni políticas de contraseñas seguras. • La plataforma web usa SSL para el acceso, pero los dispositivos aun envían información por puertos no seguros, lo que podría permitir interceptación de datos sensibles. • Los datos están alojados en un VPS y únicamente se cifra la información relacionada con contraseña. No se especifica protección para otros tipos de datos biomédicos. • No se han evaluado vulnerabilidades como XSS o fallos de autenticación. • Solo algunos formularios o APIs tienen mecanismos de validaciones, lo que deja parte del sistema expuesta a ataques por datos maliciosos. • Las librerías se mantienen actualizadas cuando es posible, pero no existe un proceso formal para verificar vulnerabilidades en cada actualización. • No se han ejecutado pruebas de penetración, análisis de vulnerabilidades ni pruebas dinámicas. • El sistema no cuenta con planes de recuperación en caso de fallos técnicos o ataques.

Registro descriptivo de la Información	
- No se generan registros logs ni cuenta con alertas de seguridad de detección de intruso. - Las mejoras de seguridad no se han implementado porque se han priorizado el funcionamiento del sistema antes que la protección de la información. - Aun no se han implementado validaciones completas de entradas de datos, identificadas como una necesidad desde la fase de diseño.	
Resumen:	El sistema de monitoreo biomédico presenta varios riesgos de seguridad debido a la falta de controles avanzados en autenticación, transmisión de datos y protección de información almacenada. Además, no se han aplicado estándares de seguridad como OWASP, ni se realizan pruebas de seguridad, respaldos o monitoreo de accesos. Por otro lado, la funcionalidad ha sido priorizada sobre la seguridad, lo que deja expuesto el sistema a ante posible ataques o pérdidas de información.
Responsable:	Angel Tomala Cinthia Vanessa.

Anexo 3. Plantillas de las 90 preguntas del modelo SAMM

Gobernancia							
Stream	Nivel	Estrategia y métricas	Respuesta			Notas de la entrevista	Clasificación
Crear y Promover	1	¿Entiende usted el apetito de riesgo que existe en toda la empresa para sus aplicaciones?	N	0	0,000		0,00
		Captura la tolerancia al riesgo de la dirección ejecutiva de su organización. La dirección de la organización examina y aprueba el conjunto de riesgos. Identifica las principales amenazas empresariales y técnicas para sus activos y datos. Documenta los riesgos y los almacena en un lugar accesible.					
	2	¿Tiene un plan estratégico para la seguridad de las aplicaciones y lo utiliza para tomar	0	0	0,000		
		El plan refleja las prioridades comerciales y la tolerancia al riesgo de la organización. El plan incluye hitos mensurables y un presupuesto. El plan es coherente con los impulsores y riesgos comerciales de la organización. El plan establece una hoja de ruta para iniciativas estratégicas y tácticas. Cuenta con la aceptación de las partes interesadas, incluidos los equipos de desarrollo.					
	3	¿Revisa y actualiza periódicamente el Plan Estratégico de Seguridad de Aplicaciones?	P	0	0,000		
		Revisa y actualiza el plan en respuesta a cambios significativos en el entorno empresarial, la organización o su tolerancia al riesgo. Los pasos para actualizar el plan incluyen revisarlo con todas las partes interesadas y actualizar los impulsores y estrategias del negocio. Ajusta el plan y la hoja de ruta en función de las lecciones aprendidas de las actividades completadas. Publica información sobre el progreso de las actividades de la hoja de ruta, asegurándose de que esté disponible para todas las partes interesadas.					
	1	¿Utiliza un conjunto de métricas para medir la eficacia y eficiencia del programa de seguridad de aplicaciones en todas las aplicaciones?	Q	0			
		Documenta cada métrica, incluyendo una descripción de las fuentes, la cobertura de la medición y una guía sobre cómo usarla para explicar las tendencias de seguridad de aplicaciones. Las métricas incluyen medidas de esfuerzo, resultados y las categorías de medición del entorno. La mayoría de las métricas se miden con frecuencia, son fáciles o económicas de recopilar y se expresan como un número cardinal o un porcentaje. Los equipos de seguridad y desarrollo de aplicaciones publican las métricas.					

Diseño							
Evaluación de amenazas			Respuesta			Notas de la entrevista	Clasificación
Perfil de Riesgo de la Aplicación	1	¿Clasifica las aplicaciones según el riesgo empresarial basándose en un conjunto simple y predefinido de preguntas?	B		0	0,000	0,00
		Existe una clasificación de riesgos acordada. El equipo de aplicaciones comprende la clasificación de riesgos. La clasificación de riesgos abarca aspectos críticos de los riesgos empresariales que enfrenta la organización. La organización cuenta con un inventario de las aplicaciones dentro del alcance.					
	2	¿Utiliza perfiles de riesgo de aplicaciones centralizados y cuantificados para evaluar el riesgo empresarial?	A		0	0,000	
		El perfil de riesgo de la aplicación cumple con el estándar de riesgo organizacional. El perfil de riesgo de la aplicación cubre el impacto en la seguridad y la privacidad. La calidad del perfil de riesgo se valida de forma manual o automática. Los perfiles de riesgo de la aplicación se almacenan en un inventario central.					
	3	¿Revisa y actualiza periódicamente los perfiles de riesgo de sus aplicaciones?	R		0	0,000	
Modelado de amenazas		El estándar de riesgo organizacional considera la retroalimentación histórica para mejorar el método de evaluación. Los cambios significativos en la aplicación o el contexto empresarial dan lugar a una revisión de los perfiles de riesgo relevantes.					
	1	¿Identifica y gestiona las fallas de diseño arquitectónico mediante el modelado de amenazas?	B		0		
		Realiza modelado de amenazas para aplicaciones de alto riesgo. Utiliza listas de verificación de amenazas sencillas, como STRIDE. Conserva el resultado de un modelo de amenazas para su uso posterior.					
	2	¿Utiliza una metodología estándar, alineada con los niveles de riesgo de sus aplicaciones?	A		0		
		Capacita a sus arquitectos, líderes de seguridad y otras partes interesadas en cómo realizar un modelado práctico de amenazas. Tu metodología de modelado de amenazas incluye, como mínimo, la creación de diagramas, la identificación de amenazas, la mitigación de fallas de diseño y la validación de los artefactos de tu modelo de amenazas. Los cambios en la aplicación o el contexto empresarial desencadenan una revisión de los modelos de amenazas.					

Implementación							
Construcción segura			Respuesta			Notas de la entrevista	Clasificación
Proceso de Compilación	1	¿Está formalmente descrito todo su proceso de compilación?	A		0	0,000	0,00
		Tienes suficiente información para recrear los procesos de compilación. Tu documentación de compilación está actualizada. Tu documentación de compilación se almacena en una ubicación accesible. Las sumas de comprobación de los artefactos producidos se crean durante la compilación para facilitar la verificación posterior. Fortaleces las herramientas utilizadas en el proceso de compilación.					
	2	¿Está el proceso de compilación completamente automatizado?	A		0	0,000	
		El proceso de compilación en sí no requiere interacción humana. Sus herramientas de compilación están reforzadas según las mejores prácticas y las directrices del proveedor. Usted cifra los secretos que requieren las herramientas de compilación y controla el acceso según el principio del mínimo privilegio.					
	3	¿Implementa comprobaciones de seguridad automatizadas en sus procesos de compilación?	A		0	0,000	
Depende		Las compilaciones fallan si la aplicación no cumple con una línea base de seguridad predefinida. Tiene una severidad máxima aceptada para las vulnerabilidades. Registra las advertencias y los fallos en un sistema centralizado. Selecciona y configura herramientas para evaluar cada aplicación según sus requisitos de seguridad al menos una vez al año.					
	1	¿Tiene un conocimiento sólido de las dependencias en las que se basa?	A		0		
		Tiene una lista de materiales (BOM) actualizada para cada aplicación. Puede averiguar rápidamente qué aplicaciones se ven afectadas por un CVE específico. Ha analizado, abordado y documentado los hallazgos de las dependencias al menos una vez en los últimos tres meses.					
	2	¿Gestiona el riesgo de dependencias de terceros mediante un proceso formal?	A		0		
		Mantiene una lista de dependencias aprobadas que cumplen con los criterios predefinidos. Evalúa automáticamente las dependencias para detectar nuevos CVE y alerta al personal responsable.					

Verificación							
Evaluación de Arquitectura			Respuesta			Notas de la entrevista	Clasificación
Validación de la arquitectura	1	¿Revisa la arquitectura de la aplicación para determinar los objetivos de seguridad clave según las necesidades?	A		0	0,000	0,00
		Tiene un modelo acordado de la arquitectura general del software. Incluye componentes, interfaces e integraciones en el modelo de arquitectura. Verifica la correcta provisión de mecanismos generales de seguridad. Registra los controles de seguridad faltantes como defectos.					
	2	¿Revisa periódicamente los mecanismos de seguridad de su arquitectura?	A		0	0,000	
		Revisa el cumplimiento de los requisitos internos y externos. Revisa sistemáticamente cada interfaz del sistema. Utiliza un método de revisión formalizado y una validación estructurada. Registra los mecanismos de seguridad faltantes como defectos.					
	3	¿Revisa periódicamente la eficacia de los controles de seguridad?	A		0	0,000	
Mitigación de la arquitectura		Evaluará las capacidades preventivas, de detección y de respuesta de los controles de seguridad. Evaluará la alineación estratégica, el reporte adecuado y la escalabilidad de los controles de seguridad. Evaluará su eficacia al menos una vez al año. Registrará las deficiencias identificadas como defectos.					
	1	¿Revisa la arquitectura de la aplicación para mitigar las amenazas típicas según las necesidades?	A		0		
		Tiene un modelo acordado de la arquitectura general del software. Personal experto en seguridad realiza la revisión. Considera diferentes tipos de amenazas, incluidas las internas y las relacionadas con los datos.					
	2	¿Evalúa periódicamente las amenazas a su arquitectura?	A		0		
		Revisa sistemáticamente cada amenaza identificada en la Evaluación de Amenazas. Personal capacitado o con experiencia lidera el ejercicio de revisión. Identifica las características de mitigación a nivel de diseño para cada amenaza identificada. Registra las amenazas no gestionadas como defectos.					
	3	¿Actualiza periódicamente sus arquitecturas de referencia según los resultados de la evaluación	A		0		

Operación						
Gestión de Incidentes		Respuesta		Notas de la entrevista		Clasificación
Detección de incidentes	1	¿Analiza periódicamente los datos de registro para detectar incidentes de seguridad?	A	0	0,000	0,00
		Dispone de un punto de contacto para la creación de incidentes de seguridad. Analiza los datos según los periodos de retención de datos de registro. La frecuencia de este análisis se ajusta a la criticidad de sus aplicaciones.				
	2	¿Sigue un proceso documentado para la detección de incidentes?	A	0	0,000	
		El proceso tiene un responsable específico. Almacena la documentación del proceso en un lugar accesible. El proceso considera una ruta de escalamiento para un análisis más detallado. Capacita a los empleados responsables de la detección de incidentes en este proceso. Cuenta con una lista de verificación de posibles ataques para simplificar la detección de incidentes.				
	3	¿Revisa y actualiza periódicamente el proceso de detección de incidentes?	A	0	0,000	
		Realizas revisiones al menos una vez al año. Actualizas la lista de posibles ataques con datos externos e internos.				
Respuesta a incidentes	1	¿Responde a los incidentes detectados?	H	0		
		Tiene una persona o rol definido para la gestión de incidentes. Documenta los incidentes de seguridad.				
	2	¿Utiliza un proceso repetible para la gestión de incidentes?	I	0		
		Se ha acordado una clasificación de incidentes. El proceso considera el análisis de causa raíz para incidentes de alta gravedad. Los empleados responsables de la respuesta a incidentes están capacitados en este proceso. Disponemos de herramientas de análisis forense.				
	3	¿Dispone de un equipo dedicado a la respuesta a incidentes?	E	0		
		El equipo realiza un análisis de causa raíz para todos los incidentes de seguridad, a menos que exista una razón específica para no hacerlo. Usted revisa y actualiza el proceso de respuesta al menos una vez al año.				