QAScan – Quality Assurance Scanner

NICOLAS ESTEBAN GARNICA HERRERA

JUAN SEBASTIAN HERNANDEZ AMAYA

JULIAN DAVID RODRIGUEZ MARTINEZ

Proyecto de Grado – Pregrado – Grupo 1- FIN – Primer Semestre - 2025

Universidad EAN

Facultad de Ingeniería

13 de junio de 2025, Bogotá D.C, Colombia

Tabla de contenido

RESUMEN

El presente proyecto de grado tiene como objetivo desarrollar un aplicativo web orientado al análisis estático de código y dependencias, apoyado en servicios de ciberseguridad, con el fin de identificar vulnerabilidades en el desarrollo de software utilizado por pequeñas empresas. Esta solución busca fortalecer la seguridad desde las etapas tempranas del ciclo de vida del software, promoviendo buenas prácticas en la escritura de código y la gestión de dependencias externas.

La creciente digitalización ha expuesto a las Pymes a riesgos cibernéticos que pueden comprometer su operación y la integridad de su información. En Colombia, estas organizaciones representan más del 90% del tejido empresarial y son fundamentales para la economía nacional, pero su exposición a ciberataques es significativa. En 2023, Kaspersky bloqueó 30 millones de ataques contra Pymes en el país, incluyendo 22 millones de intentos de phishing. Asimismo, el Estudio Anual de Ciberseguridad 2022-2023 de la CCIT (Cámara Colombiana de Informática y Telecomunicaciones) reportó un incremento del 26% en denuncias por ciberdelitos, siendo el hurto por medios informáticos el más frecuente.

Este panorama evidencia la necesidad de herramientas preventivas accesibles que permitan a las Pymes evaluar la seguridad de su software antes de su implementación. El aplicativo propuesto representa una alternativa sostenible, tanto técnica como económicamente, para fortalecer la ciberseguridad desde la base del desarrollo tecnológico.

ABSTRACT

The aim of this undergraduate project is to develop a web application focused on static code and dependency analysis, supported by cybersecurity services, in order to identify vulnerabilities in the software used by small businesses. This solution seeks to strengthen security from the early stages of the software development lifecycle, promoting best practices in code writing and the management of external dependencies.

The growing digitalization has exposed SMEs to cyber risks that can compromise their operations and the integrity of their information. In Colombia, these organizations account for more than 90% of the business fabric and are fundamental to the national economy, yet their exposure to cyberattacks is significant. In 2023, Kaspersky blocked 30 million attacks against SMEs in the country, including 22 million phishing attempts. Likewise, the 2022–2023 Annual Cybersecurity Study by the Colombian Chamber of Informatics and Telecommunications (CCIT) reported a 26% increase in cybercrime complaints, with theft through computer means being the most frequent.

This situation highlights the need for accessible preventive tools that allow SMEs to assess the security of their software before deployment. The proposed application represents a technically and economically sustainable alternative to strengthen cybersecurity from the foundation of technological development.

PALABRAS CLAVE

- Ciberseguridad
- Análisis estático de código
- Detección de vulnerabilidades
- Dependencias de software
- Desarrollo seguro de software
- Ciclo de vida del desarrollo seguro (SSDLC)
- Pequeñas y medianas empresas (Pymes)
- DevSecOps
- Aplicativo web
- Arquitectura de microservicios
- Prevención de amenazas

INTRODUCCIÓN

En la era digital, la ciberseguridad se ha consolidado como un pilar fundamental para la protección de la información y la continuidad operativa de las organizaciones. Las pequeñas empresas, a pesar de su importancia económica, enfrentan una alta vulnerabilidad frente a las amenazas informáticas, principalmente por la limitada disponibilidad de recursos para implementar soluciones de seguridad especializadas. Esta condición las convierte en un blanco recurrente de ataques que pueden comprometer seriamente su estabilidad y reputación.

Según el Informe de Amenazas Globales 2023 de Kaspersky, el 60% de las pequeñas empresas que fueron víctimas de ciberataques sufrieron interrupciones en sus operaciones que se extendieron por más de una semana. Asimismo, el Instituto Nacional de Ciberseguridad (INCIBE, 2022) destaca que el phishing y el ransomware son las amenazas más comunes que afectan a las Pymes, provocando importantes pérdidas financieras. En el contexto colombiano, la Cámara Colombiana de Informática y Telecomunicaciones (CCIT) reportó un aumento del 26% en las denuncias por ciberdelitos durante 2022, lo que refleja una creciente exposición a riesgos digitales en este sector.

Frente a este panorama, el presente proyecto propone el desarrollo de un aplicativo web para realizar análisis estático de código fuente y dependencias, utilizando herramientas de ciberseguridad que permitan detectar vulnerabilidades en fases tempranas del desarrollo de software. Esta solución se orienta a facilitar prácticas seguras en la programación y en la gestión de librerías externas, reduciendo la posibilidad de que se introduzcan fallos explotables en las aplicaciones.

El documento expone los aspectos técnicos y metodológicos que guían el diseño del aplicativo, así como su aplicabilidad en pequeñas empresas que deseen fortalecer su seguridad digital de forma accesible y eficiente. Además, se resalta la importancia de adoptar medidas preventivas en el ciclo de vida del software, como parte integral de una estrategia de ciberseguridad sostenible.

JUSTIFICACIÓN

Actualmente, la ciberseguridad es un componente esencial para garantizar la continuidad operativa y la sostenibilidad de las organizaciones, especialmente en un entorno donde la transformación digital avanza rápidamente. Las pequeñas y medianas empresas (Pymes), a pesar de su papel clave en la economía, son especialmente vulnerables a los ataques informáticos debido a la limitada disponibilidad de recursos financieros, infraestructura tecnológica insuficiente y escasa formación en ciberseguridad.

La Small Business Administration (SBA, 2023) advierte que las pequeñas empresas carecen, en su mayoría, de las herramientas y conocimientos necesarios para proteger adecuadamente sus sistemas. Esto las convierte en blancos atractivos para los ciberdelincuentes, quienes aprovechan errores comunes en el desarrollo de software, tales como código inseguro o dependencias desactualizadas con vulnerabilidades conocidas. De acuerdo con Rombaldo Junior, Becker y Johnson (2023), ataques como el ransomware y el phishing continúan siendo los más frecuentes, generando pérdidas económicas, afectaciones a la reputación y paralización de operaciones.

Frente a este panorama, se vuelve indispensable contar con herramientas preventivas que permitan identificar vulnerabilidades en las fases tempranas del desarrollo de software. El análisis estático de código y la revisión de dependencias externas emergen como estrategias clave para reducir la superficie de ataque de las aplicaciones antes de su despliegue. A diferencia de los sistemas de monitoreo en tiempo real, esta aproximación permite detectar fallos estructurales desde el origen, promoviendo buenas prácticas de programación y evitando riesgos desde la base.

El presente proyecto propone un aplicativo web accesible y adaptable, orientado a pequeñas empresas, que facilite el uso de herramientas de análisis estático y escaneo de dependencias con enfoque en ciberseguridad. Esto responde a la necesidad de soluciones con una adecuada relación costo-beneficio, fáciles de integrar en procesos de desarrollo, y que no requieran una alta especialización técnica. Según NIST (2023), este tipo de herramientas no solo contribuyen a reducir errores humanos y automatizar tareas repetitivas, sino que también permiten a las organizaciones crecer de manera segura, adaptando sus mecanismos de protección sin necesidad de reemplazar su arquitectura tecnológica.

En resumen, este proyecto no solo fortalecerá la protección digital de las Pymes desde una perspectiva preventiva, sino que también promoverá una cultura de desarrollo seguro, apoyando la construcción de un entorno empresarial más resiliente, competitivo y alineado con las exigencias del mercado digital actual.

OBJETIVOS DEL PROYECTO

OBJETIVO GENERAL

Desarrollar un prototipo de aplicativo web para desarrolladores de pequeñas empresas que permita la realización de análisis estático de código y dependencias mediante servicios de ciberseguridad.

OBJETIVOS ESPECIFICOS

- Analizar los principales riesgos de seguridad asociados al desarrollo de software y al uso de dependencias por los desarrolladores en pequeñas empresas.
- Diseñar la arquitectura de un aplicativo web que integre servicios de análisis estático de código y evaluación de vulnerabilidades en dependencias.
- Implementar y evaluar el aplicativo mediante pruebas funcionales y de seguridad bajo un modelo de aseguramiento de calidad, determinando su efectividad.
- Promover la adopción de buenas prácticas de ciberseguridad en el desarrollo de software, contribuyendo a la concienciación de los usuarios finales.

DEFINICIÓN DEL PROBLEMA

En la era digital, las pequeñas y medianas empresas (Pymes) han logrado un crecimiento significativo mediante la incorporación de tecnologías que mejoran la eficiencia operativa, la automatización de procesos y la comunicación con clientes y proveedores. Sin embargo, esta transformación digital también ha incrementado su exposición a riesgos cibernéticos, especialmente cuando se desarrollan o utilizan aplicaciones con vulnerabilidades de seguridad.

En Colombia, las Pymes representan más del 90% del tejido empresarial y son fundamentales para la economía nacional. A pesar de su importancia, muchas de estas organizaciones no cuentan con los recursos financieros ni con el conocimiento técnico necesario para aplicar controles de ciberseguridad efectivos en sus procesos de desarrollo de software. Esta situación las convierte en un blanco frecuente de ciberdelincuentes que explotan fallos en el código fuente o en dependencias externas utilizadas sin las debidas verificaciones de seguridad.

Según datos de Kaspersky, en 2023 se bloquearon 30 millones de intentos de ataque contra Pymes en Colombia, de los cuales más de 22 millones correspondieron a intentos de phishing. De igual manera, el Estudio Anual de Ciberseguridad 2022-2023 de la Cámara Colombiana de Informática y Telecomunicaciones (CCIT) reportó un aumento del 26% en las denuncias por ciberdelitos durante 2022, siendo el hurto por medios informáticos el más común.

Este contexto evidencia la necesidad urgente de que las Pymes adopten herramientas de seguridad que permitan detectar vulnerabilidades desde el inicio del desarrollo de sus aplicaciones. Sin embargo, existe una carencia de soluciones accesibles, especializadas y

orientadas a este segmento del mercado, lo que incrementa el riesgo de que errores comunes en el código o en las bibliotecas utilizadas se conviertan en puertas de entrada para ataques.

En este sentido, la problemática central se puede formular como la siguiente pregunta: ¿Cómo desarrollar una herramienta accesible para pequeñas y medianas empresas que permita detectar vulnerabilidades en el código fuente y sus dependencias, contribuyendo a la mitigación de riesgos cibernéticos desde las etapas iniciales del desarrollo de software?

MARCO TEÓRICO

Las pequeñas y medianas empresas (Pymes) en Colombia enfrentan crecientes desafíos en materia de ciberseguridad, especialmente en lo relacionado con el desarrollo seguro de software. Debido a sus recursos limitados y a la falta de personal especializado, estas organizaciones rara vez aplican prácticas de análisis preventivo que les permitan identificar vulnerabilidades en el código fuente o en las bibliotecas externas que utilizan. Esta debilidad estructural aumenta significativamente el riesgo de exposición a ciberataques.

La ciberseguridad ha pasado de ser una preocupación exclusiva de grandes corporaciones a convertirse en una prioridad para todo tipo de organizaciones. Las Pymes, que representan más del 90% del tejido empresarial colombiano, se han visto particularmente afectadas por la sofisticación y frecuencia de las amenazas digitales. Según la Cámara Colombiana de Informática y Telecomunicaciones (CCIT, 2023), el 60% de estas empresas ha sido víctima de al menos un ciberataque en los últimos dos años. Estos incidentes, en su mayoría relacionados con malas prácticas de desarrollo o uso de dependencias vulnerables, generan pérdidas financieras, interrupciones operativas y daño reputacional.

En este contexto, surge el análisis estático de código como una herramienta esencial para fortalecer la seguridad del software desde su fase de construcción. El análisis estático se refiere a la inspección automática del código fuente sin ejecutarlo, con el fin de detectar errores, vulnerabilidades de seguridad, malas prácticas y dependencias con fallos conocidos. Esta técnica es especialmente valiosa en entornos con escasa experiencia en ciberseguridad, ya que permite automatizar la identificación de problemas críticos sin requerir conocimientos avanzados en seguridad ofensiva o auditoría de software (OWASP, 2023).

Desde la perspectiva de la ingeniería de software, el desarrollo seguro se logra integrando principios de calidad y seguridad desde las etapas iniciales del ciclo de vida del software. En este contexto, modelos como el Secure Software Development Lifecycle (Secure SDLC) promueven la incorporación de controles de seguridad desde el análisis de requisitos hasta la implementación. Marcos de referencia como el Microsoft Security Development Lifecycle (SDL), el NIST SP 800-218 (SSDF) y el OWASP Software Assurance Maturity Model (SAMM) ofrecen lineamientos concretos para que organizaciones, incluso con recursos limitados, mejoren sus prácticas de desarrollo seguro.

En los últimos años, han emergido múltiples soluciones tecnológicas que permiten abordar la seguridad desde la fase de desarrollo. Este panorama constituye el estado del arte en herramientas y estrategias aplicadas para mitigar riesgos de seguridad en el desarrollo de software. Plataformas como SonarQube permiten realizar análisis estático del código para identificar malas prácticas, errores y posibles vulnerabilidades. Por su parte, herramientas como Snyk y OWASP Dependency-Check se enfocan en el análisis de dependencias de terceros, alertando sobre el uso de bibliotecas con vulnerabilidades conocidas, consultando bases de datos como el CVE (Common Vulnerabilities and Exposures).

Asimismo, GitHub Dependabot ha sido adoptado ampliamente para automatizar la gestión de dependencias inseguras en proyectos de código abierto y privado. Estas soluciones han sido clave en la implementación de prácticas DevSecOps, que integran la seguridad como parte del flujo continuo de desarrollo y entrega (CI/CD).

Adicionalmente, muchas de las vulnerabilidades en los sistemas actuales provienen de bibliotecas y dependencias externas. Según el informe de Synopsys (2023), más del 84% del código en aplicaciones modernas proviene de componentes de terceros, de los cuales más del 40% contienen vulnerabilidades conocidas. Por ello, el uso de herramientas que escanean dependencias y generan alertas sobre riesgos asociados a versiones específicas (como Snyk, Sonatype o OWASP Dependency-Check) es fundamental para mitigar riesgos de forma proactiva.

En cuanto a las vulnerabilidades más comunes en el desarrollo, OWASP (2023) identifica entre las más críticas: inyecciones (SQL, XML, etc.), exposición de datos sensibles, fallos en la autenticación, configuración insegura y uso de dependencias vulnerables. Estos errores pueden mitigarse mediante la aplicación sistemática de pruebas de seguridad, revisión de código y herramientas de análisis automatizado.

Existen diversas técnicas para el análisis de código que complementan la detección de errores. El análisis estático (SAST) permite escanear el código sin necesidad de ejecución, identificando errores estructurales y problemas de seguridad. Herramientas como SonarQube, Semgrep, Bandit y Flawfinder permiten realizar estos análisis de forma automatizada. Por otro lado, el análisis dinámico (DAST) prueba el comportamiento de la aplicación en ejecución para descubrir vulnerabilidades en tiempo real. Ambas técnicas son fundamentales en estrategias DevSecOps modernas.

Diversos casos de éxito demuestran la eficacia de estas prácticas. Organizaciones como Mozilla han reducido significativamente el número de vulnerabilidades en Firefox gracias a la

integración continua de herramientas de análisis y políticas de revisión continua del código fuente. Spotify, por su parte, implementó pipelines con validaciones automáticas de seguridad, lo que les permitió identificar errores antes de la etapa de despliegue. Estas experiencias refuerzan la idea de que el desarrollo seguro no requiere grandes presupuestos, sino una adecuada estrategia técnica y organizacional.

El uso de aplicaciones integradas que combinan análisis estático con escaneo de dependencias permite centralizar funciones clave de seguridad sin sobrecargar a los equipos técnicos. Estas soluciones ofrecen interfaces accesibles, generan informes detallados sobre los riesgos identificados y permiten establecer flujos de mejora continua. Según ENISA (2023), estas herramientas pueden reducir hasta en un 35% la exposición a vulnerabilidades explotables en entornos empresariales con bajos niveles de madurez en ciberseguridad.

A pesar de los beneficios, existen desafíos que dificultan la adopción de este tipo de soluciones, como el costo inicial de implementación, la falta de formación técnica del personal o la resistencia al cambio en los procesos internos de las Pymes (Exponencial Confirming, 2023). Sin embargo, la creciente disponibilidad de herramientas de código abierto y servicios en la nube ha permitido democratizar el acceso a estas tecnologías.

En este sentido, el desarrollo de un aplicativo web especializado y accesible para Pymes, que integre funciones de análisis estático de código y escaneo de dependencias, se presenta como una alternativa efectiva para reducir la exposición a amenazas, mejorar la calidad del software y fortalecer la postura de ciberseguridad desde una perspectiva preventiva.

ANALÍSIS DE REQUERIMIENTOS

Para dar cumplimiento a los objetivos del proyecto fue necesario establecer los siguientes requerimientos.

- R01: Análisis de código
    - **Tarea:** Analizar el código fuente de los proyectos para detectar vulnerabilidades.
    - **Objetivo:** Identificar posibles fallos de seguridad en el código antes de su implementación.
    - **Criterios de aceptación:** El prototipo debe escanear el código y detectar vulnerabilidades comunes y debe generar un reporte con los hallazgos y sugerencias de solución.
- R02: Detección de dependencias vulnerables
    - **Tarea:** Identificar librerías o paquetes con vulnerabilidades conocidas.
    - **Objetivo:** Evitar el uso de dependencias inseguras en los proyectos.
    - **Criterios de aceptación:** El prototipo debe comparar las dependencias con bases de datos de vulnerabilidades conocidas (CVE, NVD, Snyk, etc.). También debe alertar cuando una versión insegura esté en uso.
- R03: Generación de reportes
    - **Tarea:** Crear informes detallados con los hallazgos del análisis
    - **Objetivo:** Proveer información clara y útil para corregir vulnerabilidades.

- **Criterios de aceptación:** El reporte debe contener una lista de vulnerabilidades detectadas, su nivel de severidad y posibles soluciones. Este reporte debe ser exportable en formatos como PDF o HTML.

- R05: Análisis de configuración
  - **Tarea:** Revisar configuraciones de servidores, bases de datos y archivos de configuración en busca de errores de seguridad.
  - **Objetivo:** Prevenir configuraciones inseguras que puedan ser explotadas.
  - **Criterios de aceptación:** Debe verificar configuraciones como permisos excesivos, contraseñas por defecto o exposición de servicios críticos. Y debe generar alertas en caso de configuraciones inseguras.

ANÁLISIS DE RESTRICCIONES

Para el desarrollo del aplicativo de análisis de ciberseguridad para pequeñas empresas, es fundamental identificar las restricciones que pueden afectar su diseño, implementación y operación. A continuación, se detallan las principales restricciones del proyecto:

1. Restricciones Técnicas

- Capacidad de procesamiento y almacenamiento: El prototipo debe operar con recursos limitados, ya que muchas Pymes no cuentan con infraestructura tecnológica avanzada.
- Compatibilidad con diferentes sistemas operativos y plataformas: Debe integrarse con entornos Windows, Linux y MacOS, así como con plataformas en la nube.
- Conectividad y ancho de banda: El monitoreo en tiempo real requiere un uso eficiente de los recursos de red para no afectar la operación de la empresa.
- Seguridad y cifrado: La transmisión y almacenamiento de datos debe cumplir con estándares de seguridad como AES-256 y protocolos de autenticación segura.

2. Restricciones Económicas

- Presupuesto limitado: Las Pymes tienen restricciones en la inversión en ciberseguridad, por lo que el prototipo debe ser accesible y rentable.
- Costos de mantenimiento y actualización: Se debe garantizar que los costos operativos sean bajos para que las empresas puedan mantener la solución a largo plazo.

3. Restricciones Organizacionales

- Capacitación del personal: El prototipo debe ser fácil de usar, ya que muchas Pymes no cuentan con expertos en ciberseguridad.

- Adopción y resistencia al cambio: La implementación de nuevas tecnologías puede enfrentar resistencia por parte del personal.

4. Restricciones Regulatorias y Normativas

- Cumplimiento con normativas locales e internacionales: Debe cumplir con leyes como la Ley 1581 de 2012 de Protección de Datos en Colombia y estándares internacionales como ISO 27001.
- Almacenamiento y tratamiento de datos: La solución debe garantizar la privacidad y protección de los datos de los clientes y usuarios.

METODOLOGIA PARA EL DESARROLLO DE LA SOLUCIÓN

Para garantizar el éxito del aplicativo web QAScan orientado al análisis preventivo de seguridad, se adoptará una metodología ágil basada en Scrum, complementada con prácticas de DevSecOps para integrar la seguridad desde las primeras etapas del desarrollo. A continuación, se describe la metodología:

1. Fase de Análisis y Requerimientos

- Identificación de necesidades específicas de las Pymes mediante encuestas y entrevistas.
- Análisis de riesgos de ciberseguridad y evaluación de amenazas más comunes.
- Definición de los requisitos funcionales del aplicativo web.

2. Fase de Diseño de la Solución

- Diseño de la arquitectura del aplicativo basada en microservicios para escalabilidad y flexibilidad.
- Planificación de la interfaz de usuario para facilitar su adopción en Pymes.

3. Fase de Desarrollo

- Implementación iterativa con ciclos de desarrollo cortos (sprints de 2-3 semanas).
- Uso de prácticas DevSecOps para garantizar la incorporación de seguridad en el ciclo de desarrollo.
- Desarrollo de los siguientes módulos clave del aplicativo web:
- Módulo de análisis estático de código fuente utilizando herramientas SAST y escaneo de dependencias con verificación en bases de datos CVE/NVD.
- Módulo de generación de reportes automáticos con hallazgos y sugerencias.

4. Fase de Pruebas y Validación

- Pruebas unitarias, de integración y de seguridad.
- Evaluación de rendimiento y escalabilidad.
- Pruebas con usuarios finales para asegurar la usabilidad

ANALISIS DE COSTOS

A continuación, se presentan los costos asociados al desarrollo del proyecto, abarcando las principales áreas que lo componen. Se detallarán los gastos estimados en investigación y diseño, desarrollo técnico, pruebas de calidad y gestión del proyecto. El objetivo es ofrecer una visión clara y realista de la inversión necesaria para llevar a cabo el aplicativo web, considerando tanto recursos humanos como tecnológicos.

COSTOS DE PRODUCCIÓN Y DESARROLLO

La ejecución del presente proyecto requirió la asignación de recursos humanos especializados en distintas fases clave: investigación, desarrollo de software, pruebas y mantenimiento. A continuación, se detallan los costos asociados a cada una de estas etapas, calculados en función del tiempo invertido por los integrantes del equipo y sus respectivas tarifas horarias. Estos valores reflejan el esfuerzo técnico y académico destinado al diseño, implementación y evaluación del aplicativo web de ciberseguridad orientado a pequeñas y medianas empresas.

Costos de Investigación y Desarrollo			
Nombre	**Horas**	**Valor X Hora**	**Total**
Sebastian Hernandez	20	$ 84.851	$ 1.697.020
Julian Rodriguez	18	$ 41.940	$ 754.920
Nicolas Garnica	8	$ 31.940	$ 255.520
Total, costos de Investigación y Desarrollo	46		$ 2.707.460

Costos de Desarrollo y Programación			
Nombre	**Horas**	**Valor X Hora**	**Total**
Sebastian Hernandez	10	$ 50.851	$ 508.510
Nicolas Garnica	10	$ 70.940	$ 609.400
Julian Rodriguez	8	$ 60.940	$ 487.520
Total, Costos de Desarrollo y Programación	40		$ 2.463.580

Pruebas, Mantenimiento e Iteraciones			
Nombre	**Horas**	**Valor X Hora**	**Total**
Sebastian Hernandez	4	$ 60.851	$ 243.404
Nicolas Garnica	2	$ 30.940	$ 61.880
Julian Rodriguez	0	$ 0	$ 0
Total, Pruebas, Mantenimiento e Iteraciones	6		$ 305.284

Cabe aclarar que los costos presentados corresponden únicamente al proceso de desarrollo del aplicativo en su fase de prototipo. Estos valores reflejan el esfuerzo invertido en investigación, diseño, implementación y pruebas funcionales del aplicativo, sin contemplar costos adicionales asociados a su despliegue comercial, escalabilidad o mantenimiento a largo plazo.

COSTOS DE IMPLEMENTACIÓN Y OPERACIÓN

Además del desarrollo del prototipo, se estimaron los costos necesarios para una posible implementación del aplicativo en un entorno real utilizando infraestructura local. Estos costos

corresponden a los servicios operativos requeridos para poner en funcionamiento el aplicativo, como el servidor de producción, licencias de uso de APIs y soporte técnico mensual, los cuales son fundamentales para garantizar la estabilidad, seguridad y operatividad continua del aplicativo en un contexto empresarial.

Infraestructura Operactiva			
Servicio	**Numero de Servicios**	**Valor X Hora**	**Total**
Servidor de Producción (local) Primer Mes	1	$ 10.000.000	$ 10.000.000
Licenciamiento de servicios API Mensual	3	$ 825.152	$ 2.475.456
Soporte técnico Mensual	1	$ 1.500.000	$ 1.500.000
Total, Infraestructura Operativa	5		$ 13.975.456

Es importante destacar que estos costos corresponden a una proyección de implementación en un entorno productivo local, y no forman parte del desarrollo del prototipo presentado. Su inclusión responde a la necesidad de ofrecer una visión preliminar del presupuesto requerido en caso de que una empresa decida poner en marcha el aplicativo como solución funcional en su infraestructura.

ANALISIS Y DISEÑO DEL APLICATIVO (PROTOTIPO)

El diagrama de casos de uso se elaboró con el propósito de representar de manera clara y concisa las principales interacciones entre el usuario final y el aplicativo QAScan para análisis de seguridad. Esta herramienta permite identificar los requerimientos funcionales desde la perspectiva del usuario, facilitando el análisis y diseño del aplicativo. Al tratarse de una plataforma gratuita y sin autenticación, el enfoque se centra en acciones directas como cargar archivos, ejecutar análisis y recibir reportes. El diagrama también ayuda a establecer los límites del aplicativo y a comunicar su funcionalidad de forma visual a los interesados en el proyecto.

ESPECIFICACIÓN DEL APLICACTIVO (PROTOTIPO)

PRODUCT BACKLOG

Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU001	Usuario	Analizar código estático y dependencias de los proyectos de desarrollo bajo cualquier lenguaje de programación	Buscar posibles errores de codificación en los desarrollos de sistemas	Entregar un informe en el que el usuario pueda validar las inconsistencias

Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU002	Servicio	Dar respuesta informado al usuario cuales son los errores encontrados	Entregar información de los errores encontrados	Entregar respuesta por medio del back-

				end de la aplicación

Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU003	Validar configuración segura	Verificar archivos de configuración del sistema y detectar configuraciones inseguras.	Para prevenir vulnerabilidades relacionadas con permisos, contraseñas por defecto o exposición de servicios.	El sistema debe analizar archivos como. env, config.yaml, settings.json y generar alertas si se detectan malas prácticas.

Codigo HU	Rol	Requerimiento	¿Para que?	Criterios de aceptación
HU004	Exportar informe de vulnerabilidades	Descargar el reporte generado del análisis.	Para conservar evidencia del estado del código y compartirlo con el equipo de desarrollo.	Permitir la descarga del informe en formato .html.

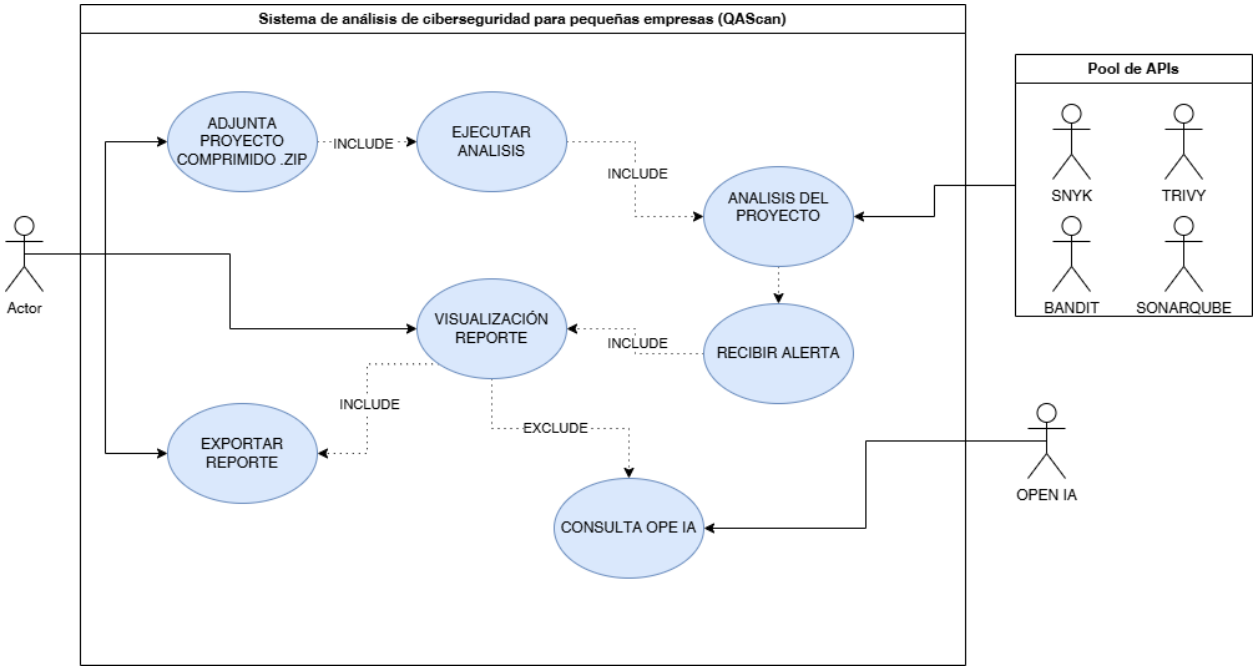
Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU005	Consultar ayuda sobre errores	Consultar explicaciones o soluciones sugeridas mediante IA.	Para entender mejor los errores reportados por el sistema y cómo corregirlos.	La interfaz debe mostrar una respuesta de IA contextualizada con base en el tipo de error detectado.

Código HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación

HU006	Enviar alertas críticas	Enviar alertas cuando se detecten errores de alta severidad.	Para informar inmediatamente sobre fallos críticos en el código o dependencias.	Si se detecta una vulnerabilidad con severidad alta o crítica, se debe notificar al usuario de forma destacada en el frontend.

Codigo HU	Rol	Requerimiento	¿Para qué?	Criterios de aceptación
HU007	Elegir herramienta de análisis	Permitir al usuario seleccionar entre herramientas como SonarQube, Snyk, Semgrep, etc.	Para adaptar el análisis según sus necesidades.	El sistema debe ejecutar la herramienta seleccionada por el usuario al subir el archivo.

DIAGRAMAS CASOS DE USO



DESCRIPCIÓN DE CASOS DE USO

NOMBRE CASO DE USO	FUNCIONALIDAD	HISTORIA DE USUARIO VINCULADA	FLUJO PRINCIPAL	FLUJO ALTERNO
ADJUNTAR PROYECTO COMPRIMIDO .ZIP	El usuario podrá a través de un seleccionador de archivos elegir su proyecto desarrollado y comprimido en .zip	HU001	1. El usuario ingresa al sitio 2. El usuario selecciona la herramienta a utilizar 3. El usuario selecciona un archivo comprimido con su proyecto.	No aplica

NOMBRE CASO DE USO	FUNCIONALIDAD	HISTORIA DE USUARIO VINCULADA	FLUJO PRINCIPAL	FLUJO ALTERNO
EJECUTAR ANALISIS	El usuario podrá ejecutar un análisis de código estático y dependencias	HU001	1. El usuario ejecuta el análisis	No aplica

NOMBRE CASO DE USO	FUNCIONALIDAD	HISTORIA DE USUARIO VINCULADA	FLUJO PRINCIPAL	FLUJO ALTERNO
ANALISIS DEL PROYECTO	Por medio del back-end el servicio recibe el proyecto entregado por el usuario	HU002	1. Recibir proyecto entregado por el usuario 2. ejecuta análisis del proyecto	No aplica

NOMBRE CASO DE USO	FUNCIONALIDAD	HISTORIA DE USUARIO VINCULADA	FLUJO PRINCIPAL	FLUJO ALTERNO
RECIBIR ALERTA	Por medio del back-end el servicio entrega una respuesta para ser interpretada por el sistema	HU002	1. Entregar respuesta con los errores encontrados	No aplica

NOMBRE CASO DE USO	FUNCIONALIDAD	HISTORIA DE USUARIO VINCULADA	FLUJO PRINCIPAL	FLUJO ALTERNO
VISUALIZAR REPORTE	Se despliega una ventana donde el usuario puede validar la información recibida por el servicio	HU001	1. Se despliega ventana 2. Se da escritura a la respuesta entregada por el servicio	No aplica

ARQUITECTURA DE LA SOLUCIÓN

Este aplicativo está estructurado con una arquitectura de microservicios que permite su despliegue modular y escalable. El siguiente diagrama representa los principales componentes del aplicativo:

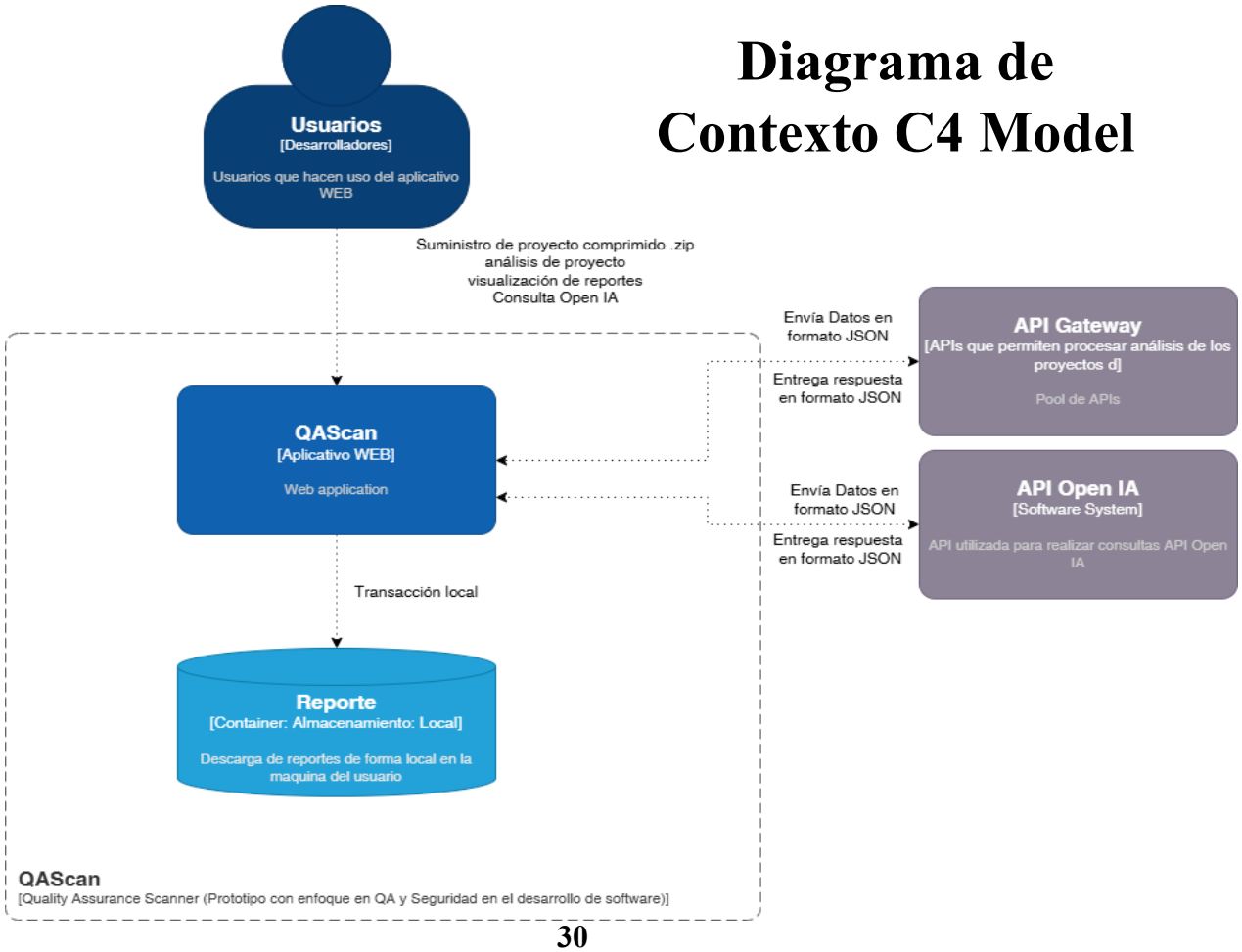


Diagrama de Contenedores C4 Model

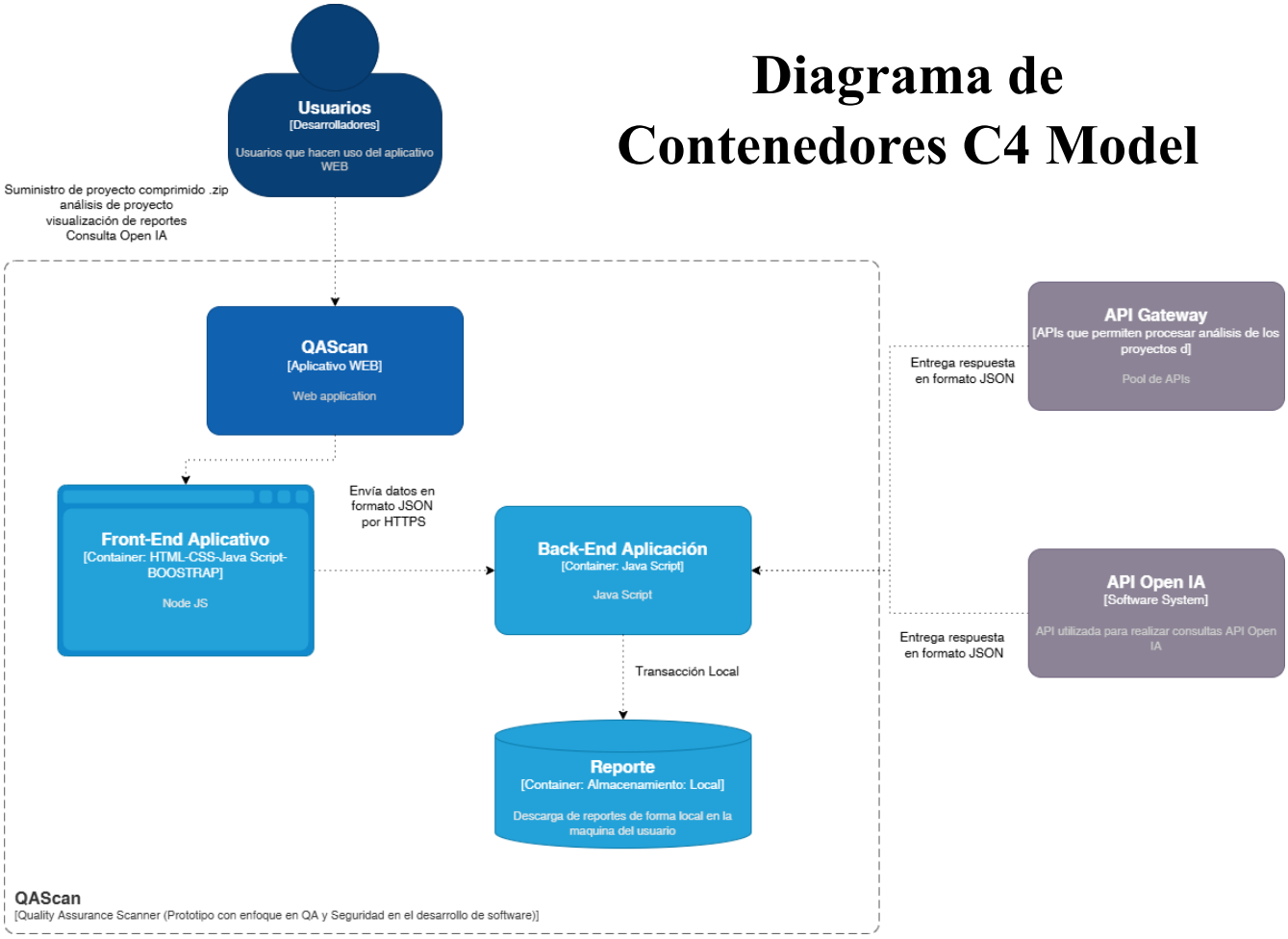
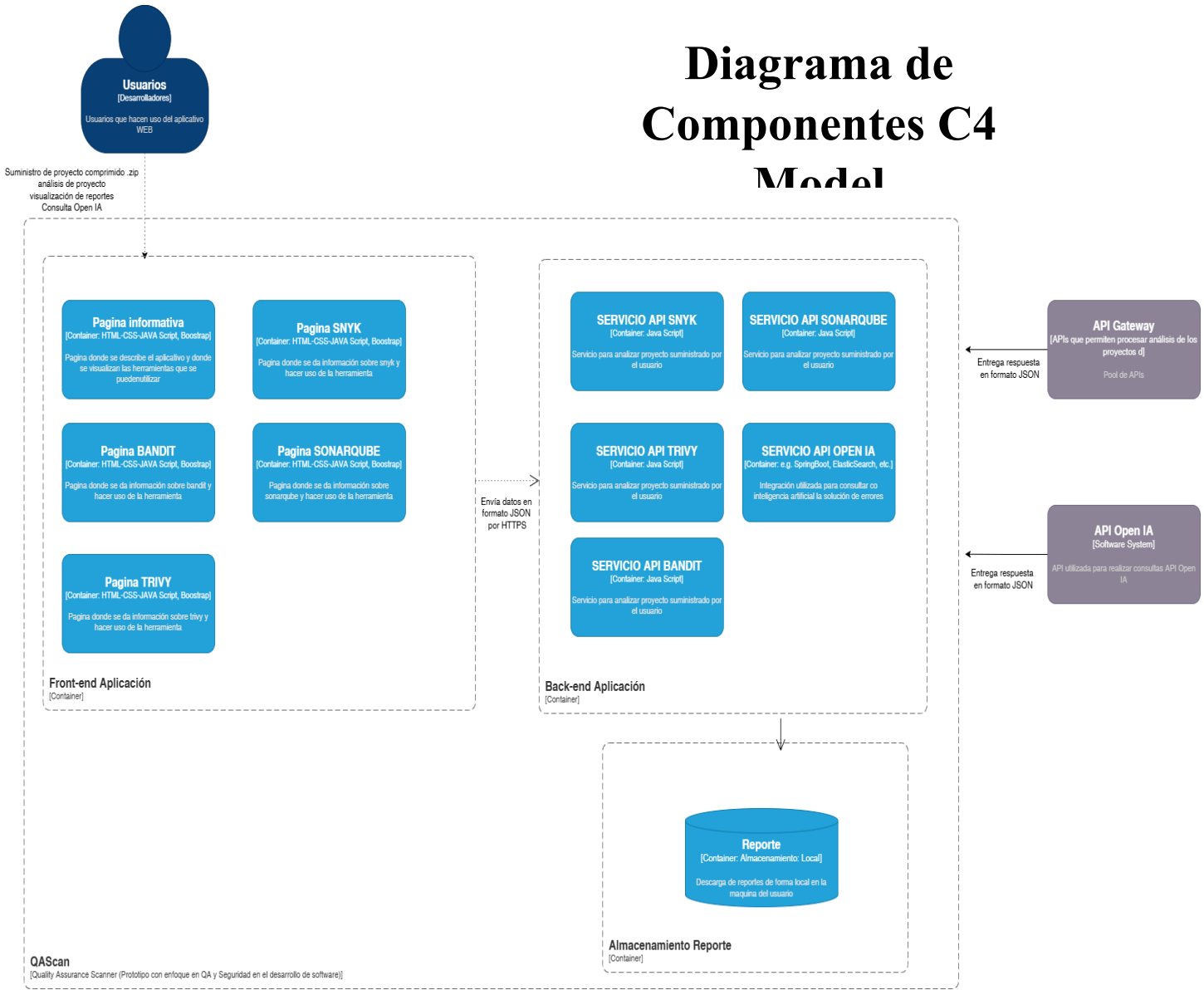


Diagrama de Componentes C4 Model



EXPLICACIÓN DE COMPONENTES

- **FRONTEND (interface WEB):** permite al usuario cargar su proyecto empaquetado en un archivo ZIP de manera sencilla y sin necesidad de registro previo. Esta interfaz es el punto de interacción inicial con el prototipo.
- **Api Gateway:** Actúa como el único punto de entrada para todas las solicitudes provenientes del Frontend. Su función principal es recibir el archivo ZIP cargado y las peticiones de análisis, para luego enrutarlas de manera inteligente a los microservicios correspondientes.
- **Servicio de análisis de código**: Este microservicio toma el código fuente del proyecto cargado y realiza un escaneo en profundidad para identificar posibles vulnerabilidades de seguridad conocidas, utilizando herramientas de análisis estático de código (SAST)
- **Servicio de análisis de dependencias:** Examina las bibliotecas y dependencias de terceros utilizadas en el proyecto del usuario, consultando bases de datos como Snyk para detectar versiones inseguras con vulnerabilidades reportadas.
- **Servicio de reportes:** Recopila los resultados del análisis de código y dependencias para generar un informe detallado en formato HTML, el cual el usuario puede visualizar y posteriormente descargar como un archivo PDF para su revisión y archivo.

COMPONENTES DE ASEGURAMIENTO DE CALIDAD DE SOFTWARE (QA)

Como parte integral del análisis y diseño del aplicativo, se incorpora un enfoque de aseguramiento de calidad (QA) que permitirá medir objetivamente el desempeño, la eficacia y la confiabilidad del aplicativo. Este enfoque es clave para garantizar que el prototipo cumpla con los estándares mínimos de calidad en entornos empresariales reales, y para ofrecer retroalimentación basada en datos durante las fases de prueba y validación.

El aseguramiento de calidad en QAScan considera los siguientes atributos clave del software:

1. Exactitud**:** Capacidad del aplicativo para entregar resultados precisos en el análisis de código y dependencias. Se mide por el número y tipo de vulnerabilidades detectadas en comparación con un análisis de referencia.

2. Fiabilidad: Estabilidad del aplicativo durante la ejecución, especialmente al procesar archivos grandes o proyectos complejos. Se medirá a través de pruebas de estrés y repetición de análisis sin errores inesperados.

3. Usabilidad: Facilidad de uso del aplicativo por parte de usuarios no expertos. Este atributo se evaluará mediante encuestas de satisfacción, pruebas con usuarios finales y análisis de interacción con la interfaz.

4. Rendimiento: Tiempo de respuesta del aplicativo para ejecutar el análisis y generar reportes. Este factor es crítico para la experiencia del usuario y se medirá en segundos o minutos dependiendo del tamaño del proyecto cargado.

5. Mantenibilidad: Facilidad con la que el sistema puede ser actualizado o adaptado. Gracias a su arquitectura basada en microservicios, cada módulo podrá ser ajustado sin comprometer la estabilidad del sistema completo.

6. Portabilidad: Capacidad del aplicativo para funcionar en distintos entornos (local o nube). Esto garantiza su aplicabilidad en empresas con diferentes niveles de infraestructura tecnológica.

Métricas de calidad propuestas:

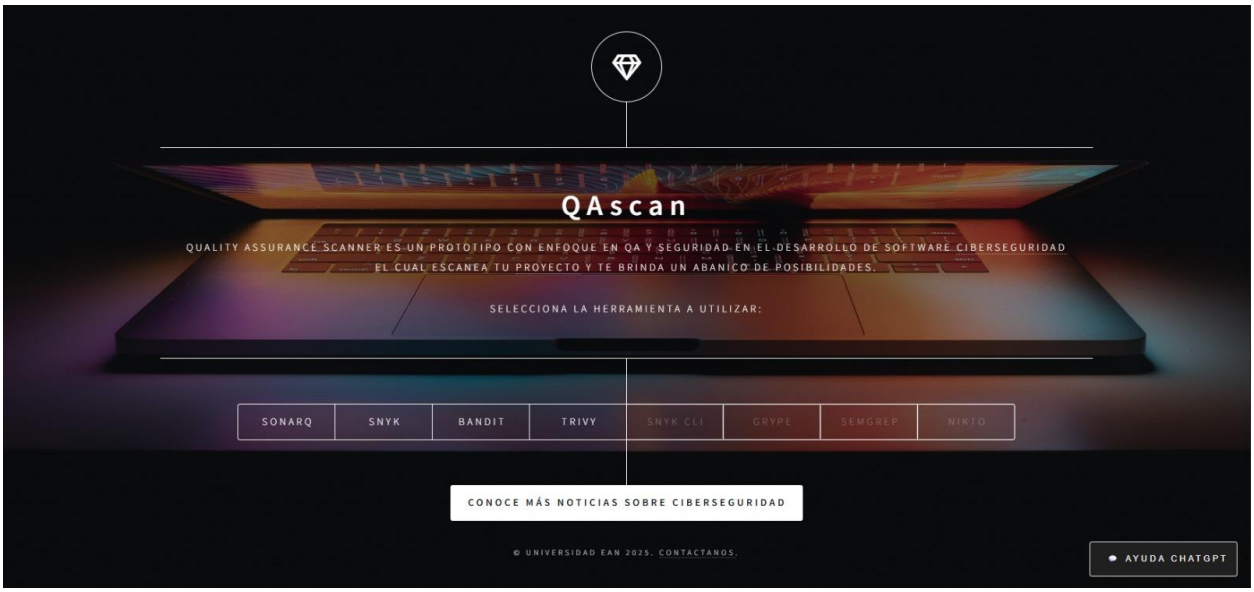
- **Cobertura de análisis:** Porcentaje de líneas de código o archivos del proyecto escaneados exitosamente.
- **Tasa de detección de vulnerabilidades:** Proporción de fallos encontrados frente a un conjunto conocido de pruebas (benchmark).
- **Falsos positivos/negativos:** Casos en los que el aplicativo reporta errores inexistentes o no detecta errores reales.
- **Tiempo medio de análisis:** Promedio de tiempo requerido para analizar proyectos de distintos tamaños.
- **Tasa de satisfacción del usuario:** Evaluada mediante formularios de retroalimentación y análisis heurístico de la interfaz.

Estas métricas no solo contribuirán a validar el correcto funcionamiento del aplicativo, sino que también fortalecerán las conclusiones del proyecto al ofrecer datos cuantificables sobre su efectividad y aplicabilidad en el contexto de las Pymes.

Además, el uso de estas métricas apoyará la mejora continua del prototipo mediante la retroalimentación objetiva y la identificación de oportunidades de optimización en futuras iteraciones. no solo contribuirán a validar el correcto funcionamiento del aplicativo, sino que también fortalecerán las conclusiones del proyecto al ofrecer datos cuantificables sobre su efectividad y aplicabilidad en el contexto de las Pymes.

PROTOTIPO NO FUNCIONAL MOCKUPS

1. PAGINA INCIAL



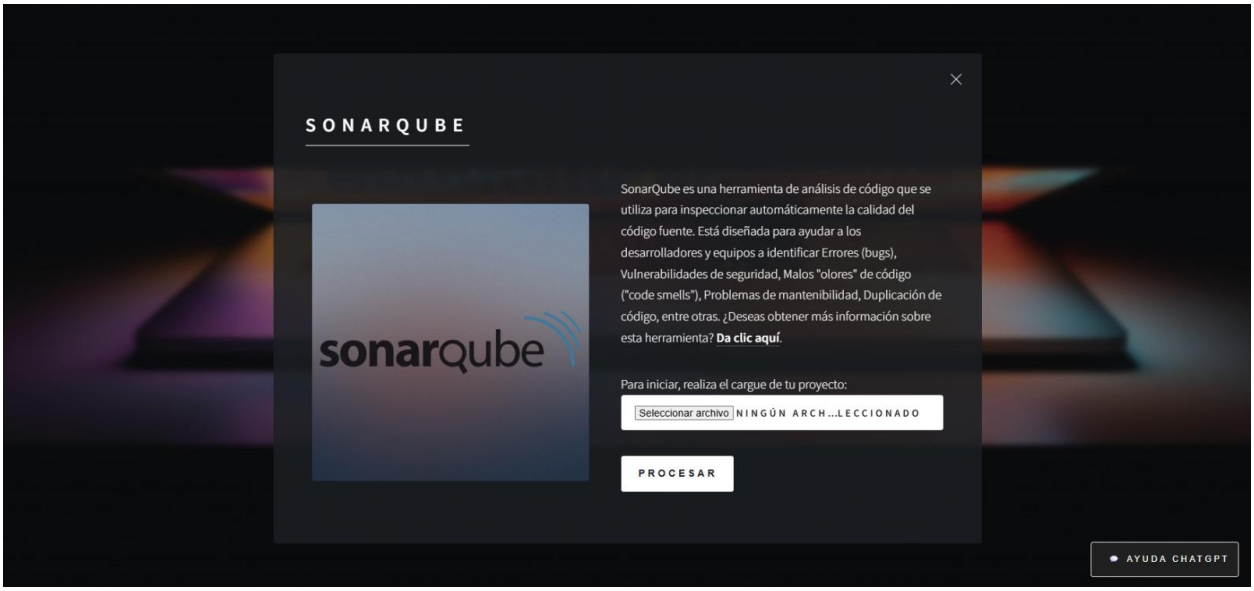
Descripción:

La imagen representa el diseño visual de la interfaz principal del aplicativo **QAScan** (Quality Assurance Scanner). En ella se muestra una propuesta estética moderna y funcional, orientada a ofrecer una experiencia de usuario intuitiva y profesional. El diseño destaca sobre un fondo oscuro con un portátil iluminado que simboliza la vigilancia digital activa y constante.

En el centro se encuentra el nombre del aplicativo, acompañado de una breve descripción que enfatiza su propósito: escanear proyectos mediante herramientas de ciberseguridad especializadas para brindar un diagnóstico integral. Debajo, se presenta un menú interactivo donde el usuario puede seleccionar herramientas como **SonarQ**, **Snyk**, **Bandit**, **Trivy**, **Grye**, **Semgrep** o **Nikto**, adaptadas a distintos tipos de análisis de vulnerabilidades y seguridad.

El diseño incorpora elementos visuales de alta calidad que transmiten tecnología y protección, reforzando el enfoque profesional del proyecto. Además, se integra una opción de asistencia con IA ("Ayuda ChatGPT") para brindar soporte en tiempo real, demostrando el valor añadido del aplicaativo en términos de accesibilidad y acompañamiento técnico.

MODULO HERRAMIENTA (4 HERRAMIENTAS GRATUITAS)



Descripción:

La imagen muestra un mockup de una interfaz gráfica de usuario diseñada para integrar y facilitar el uso de la herramienta SonarQube dentro de un entorno interactivo. Esta interfaz permite al usuario cargar un archivo de proyecto para su posterior análisis automático de calidad de código fuente mediante SonarQube.

El diseño se centra en una ventana emergente con un fondo oscuro y elegante, donde el nombre de la herramienta "SonarQube" aparece en la parte superior en letras blancas y espaciadas, destacando su identidad. A la izquierda, se presenta el logotipo oficial de SonarQube, lo que refuerza la identidad visual del sistema.

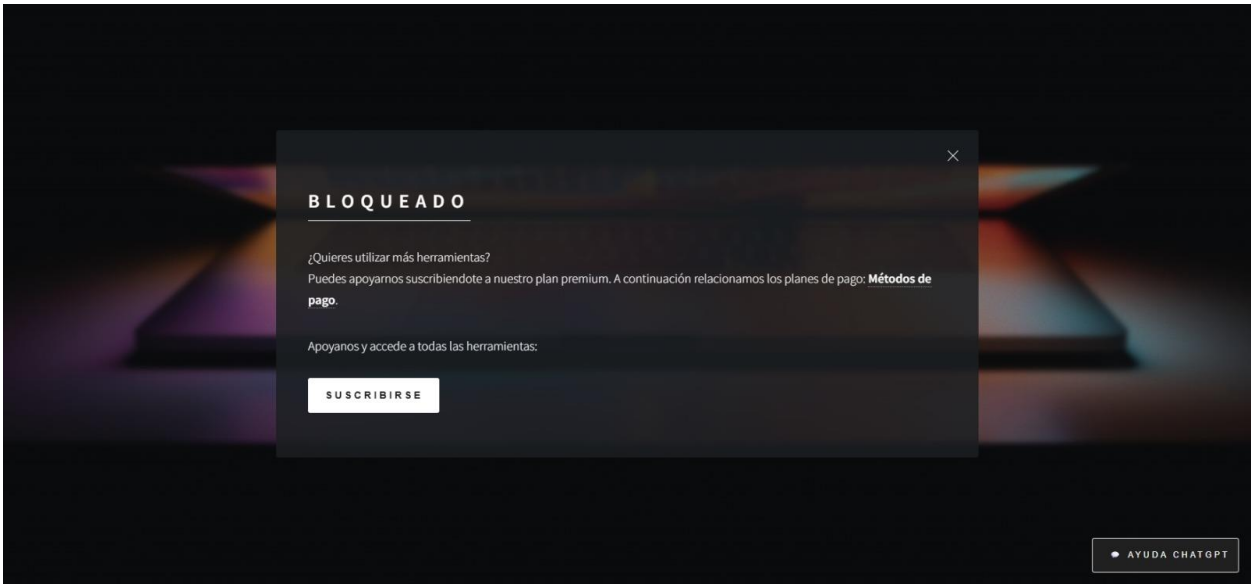
A la derecha de la imagen, se encuentra una breve pero clara descripción sobre SonarQube, explicando que es una herramienta de análisis de código utilizada para identificar errores (bugs), vulnerabilidades de seguridad, malos "olores" de código (code smells), problemas de mantenibilidad y duplicación de código, entre otros aspectos relevantes. Esta explicación se acompaña de un enlace resaltado en negrita para que el usuario pueda obtener más información si así lo desea.

En la parte inferior derecha del cuadro se dispone un botón para seleccionar y cargar un archivo del proyecto a analizar, seguido de un botón de acción etiquetado como "PROCESAR", que indica el inicio del análisis del código cargado.

Además, en la esquina inferior derecha del mockup se incluye un botón flotante con la etiqueta "AYUDA CHATGPT", que sugiere la posibilidad de recibir asistencia automatizada o contextual durante el proceso, integrando capacidades de inteligencia artificial para soporte al usuario.

En conjunto, este mockup representa una interfaz moderna, intuitiva y funcional que facilita el acceso a herramientas de análisis estático de código, promoviendo buenas prácticas de desarrollo seguro y mantenible en el marco del proyecto de grado.

MODAL SERVICIO BLOQUEDO



Descripción:

El mockup muestra una ventana de diálogo centrada y oscura con esquinas ligeramente redondeadas, que se activa cuando el usuario intenta acceder a funcionalidades "bloqueadas". Su propósito principal es invitar al usuario a suscribirse a un plan premium para desbloquear el acceso completo a las herramientas.

Elementos de la Interfaz:

1. Fondo:

- Un fondo oscuro y borroso (bokeh effect) con destellos de luz y formas geométricas sugiere un entorno digital o una interfaz de usuario subyacente. Los colores parecen ser tonos cálidos como naranjas y morados, contrastando con la ventana emergente oscura.

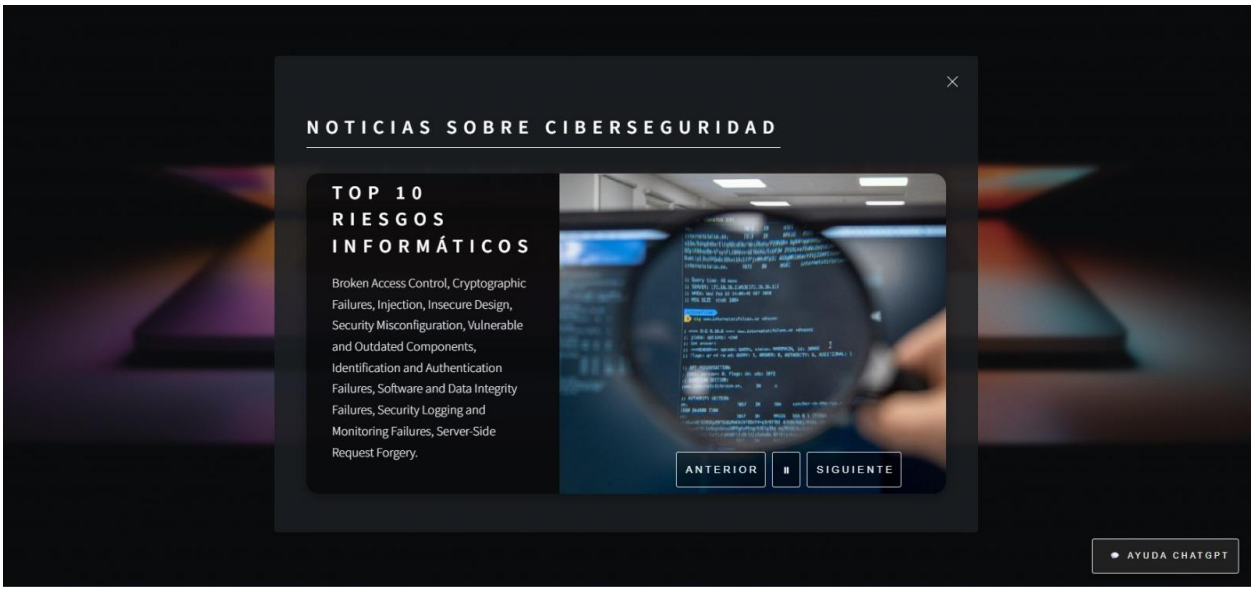
2. Ventana Emergente (Modal):
    - Color y Estilo: Predominantemente negro o gris muy oscuro, con un diseño minimalista y moderno.
    - Botón de Cierre: En la esquina superior derecha, hay un ícono de "X" en color claro (blanco o gris claro), indicando la posibilidad de cerrar la ventana emergente.
    - Título Principal: "BLOQUEADO" en mayúsculas, con un estilo de fuente limpio y moderno. Este título es prominente y comunica de inmediato el estado actual del acceso del usuario.
    - Contenido del Mensaje (Texto):
        - Pregunta Inicial: "¿Quieres utilizar más herramientas?" Esta pregunta busca generar interés y necesidad en el usuario.
        - Propuesta de Solución: "Puedes apoyarnos suscribiéndote a nuestro plan premium. A continuación relacionamos los planes de pago:" Esta frase guía al usuario hacia la solución de la suscripción y menciona la disponibilidad de información sobre los planes.

- Enlace/Mención de Métodos de Pago: "Métodos de pago" aparece resaltado, probablemente indicando un enlace o una sección donde el usuario puede encontrar esta información.
- Invitación a la Acción Final: "Apoyamos y accede a todas las herramientas:" Esta línea refuerza la propuesta de valor y el beneficio de la suscripción.

- Botón de Acción (Call to Action - CTA):
  - Texto: "SUSCRIBIRSE" en mayúsculas.
  - Estilo: Un botón rectangular con bordes redondeados y un fondo blanco o muy claro, que contrasta fuertemente con el fondo oscuro de la ventana. Esto lo hace muy visible y clickable.

3. Elemento Adicional (Inferior Derecha):
   - "● AYUDA CHATGPT" en la esquina inferior derecha del fondo, es un elemento pequeño que sugiere una opción de soporte o asistencia al usuario, posiblemente con un chatbot. El círculo pequeño y el texto en color claro indican que es un elemento secundario pero útil.

MODULO NOTICIAS CIBERSEGURIDAD



Descripción:

Este mockup representa una ventana emergente (modal) diseñada para mostrar noticias y contenido informativo sobre ciberseguridad. La ventana, de estilo oscuro y moderno, permite al usuario navegar a través de diferentes artículos o elementos de noticias.

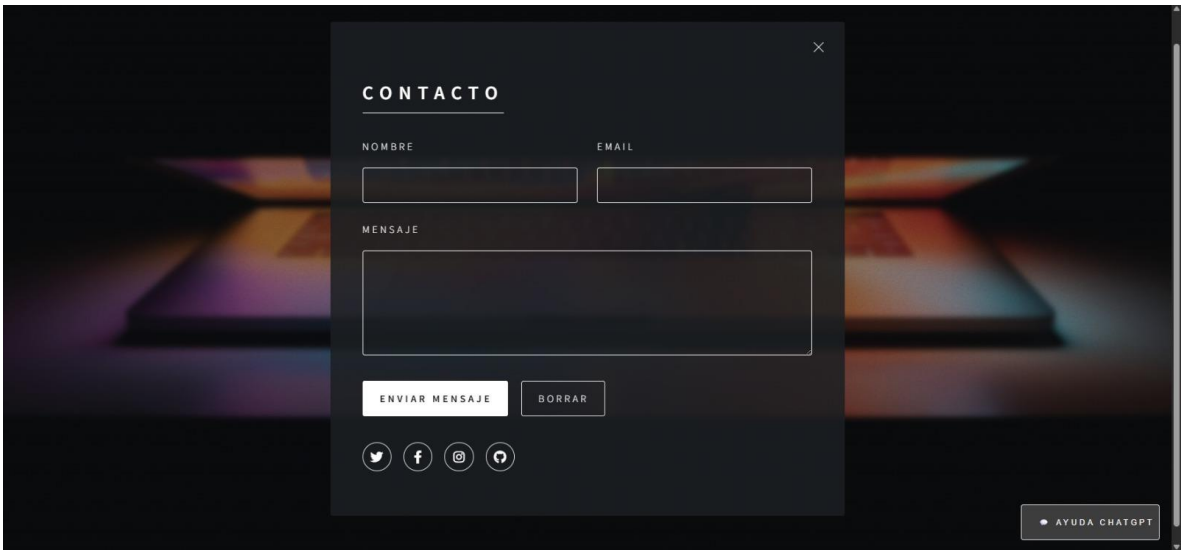
Características Clave:

- Contenido Principal: La ventana exhibe una sección destacada con un "TOP 10 RIESGOS INFORMÁTICOS", acompañada de una lista de estos riesgos y una imagen ilustrativa (una lupa sobre una pantalla con código).
- Navegación Desplazable: Los botones de "ANTERIOR" y "SIGUIENTE" en la parte inferior sugieren claramente que el usuario puede desplazarse horizontalmente para ver

distintas noticias o artículos dentro de esta misma ventana, sin necesidad de cerrarla y abrir otra.

- Diseño Consistente: Mantiene el estilo general de la aplicación con un fondo difuminado y la opción de ayuda "AYUDA CHATGPT" en la esquina inferior.
- Propósito: Sirve como un centro de información o un carrusel de noticias, permitiendo al usuario explorar contenido relevante de ciberseguridad de manera interactiva.

MODULO CONTACTO



Descripción:

Este mockup presenta una **ventana emergente (modal)** que funciona como un **formulario de contacto**, diseñada para que los usuarios puedan enviar mensajes o consultas directamente a la

plataforma o empresa. Mantiene el estilo visual de las ventanas anteriores, lo que indica consistencia en el diseño de la aplicación.

Características Clave:

- **Propósito:** Es una interfaz dedicada a la comunicación directa, permitiendo a los usuarios enviar su nombre, correo electrónico y un mensaje.
- **Campos del Formulario:**
    - **"NOMBRE":** Un campo de texto de una sola línea para que el usuario ingrese su nombre.
    - **"EMAIL":** Otro campo de texto de una sola línea para la dirección de correo electrónico del usuario.
    - **"MENSAJE":** Un área de texto más grande (textarea) que permite al usuario escribir un mensaje más extenso.
    - Todos los campos tienen un borde claro que los define sobre el fondo oscuro.
- **Botones de Acción:**
    - **"ENVIAR MENSAJE":** Un botón principal con fondo blanco y texto oscuro, que contrasta para destacar como la acción principal.
    - **"BORRAR":** Un botón secundario con borde claro y fondo oscuro, para resetear los campos del formulario.

- **Enlaces a Redes Sociales:** En la parte inferior del formulario, hay íconos de redes sociales (Twitter, Facebook, Instagram, GitHub), lo que ofrece métodos de contacto alternativos o enlaces a la presencia social de la entidad.
- **Diseño Consistente:** Al igual que los mockups anteriores, utiliza un fondo difuminado y oscuro, y mantiene el botón de cierre "X" en la esquina superior derecha y la opción de "AYUDA CHATGPT" en la parte inferior de la pantalla general.

En resumen, este mockup es un **formulario de contacto intuitivo y estilizado**, integrado coherentemente en la interfaz de usuario de la aplicación, que ofrece múltiples vías para que el usuario se comunique.

MODULO REPORTE

Este mockup representa una pantalla de confirmación o de "Reporte de Resultados", funcionando como una página de transición o de presentación de información clave al usuario. A diferencia de los anteriores, que eran ventanas emergentes, este parece ser una sección principal o una página completa dentro de la aplicación.

Características Clave:

- Título Principal: "REPORTE DE RESULTADOS" en mayúsculas y en una fuente destacada, indica claramente el propósito de la pantalla.
- Mensaje Informativo: Debajo del título, hay un texto que informa al usuario: "A CONTINUACIÓN SE MUESTRAN LOS RESULTADOS DE TU SOLICITUD, RECUERDA QUE PUEDES HACER USO DE LA OPCIÓN "AYUDA CHATGPT" SI TIENES ALGUNA DUDA CON RESPECTO A LOS RESULTADOS". Este mensaje es crucial porque:
    - Confirma que la solicitud del usuario ha sido procesada y los resultados están listos.
    - Dirige al usuario a la funcionalidad de "Ayuda ChatGPT" en caso de preguntas sobre los resultados, promoviendo el soporte interactivo.
- Elemento Visual Central: Una imagen de un ordenador portátil (laptop) abierto, con la pantalla iluminada en tonos azules y naranjas/rosas, se encuentra en el centro de la pantalla. Esta imagen es relevante ya que sugiere un entorno de trabajo digital y la visualización de información.

- Botón de Acción:
    - "VER REPORTE": Un botón rectangular con bordes redondeados y un contorno delgado, ubicado estratégicamente debajo de la laptop. Este es el principal llamado a la acción, que invita al usuario a proceder y visualizar los resultados completos.
- Elementos Separadores: Dos líneas horizontales delgadas en la parte superior e inferior del área de contenido principal añaden un toque de diseño y ayudan a enmarcar la información.
- Información del Pie de Página: En la parte inferior central, se lee "© UNIVERSIDAD EAN 2025. CONTÁCTANOS.". Esto proporciona información de derechos de autor y una invitación general a contactar, lo que es común en aplicaciones y sitios web. La mención de "UNIVERSIDAD EAN 2025" sugiere que este proyecto podría estar relacionado con un trabajo universitario.

MODULO VENTANA MODAL REPORTE

Descripción:

Este mockup muestra una **ventana emergente (modal)** que presenta los **resultados de un análisis o solicitud previa del usuario**. Es la interfaz donde el usuario finalmente ve la información que solicitó, como un informe detallado.

Características Clave:

- **Título:** "RESULTADOS", indicando claramente lo que contiene la ventana.
- **Contenido Principal:** En su interior, se visualiza un "REPORTE" con un nombre específico ("TRIVY") y un identificador de archivo ("PACKAGE-LOCK.JSON"). Lo más importante es que el cuerpo del reporte contiene **información detallada organizada en una lista**, con elementos que parecen ser hallazgos o datos técnicos (por ejemplo, identificadores, descripciones, y niveles de importancia).

- **Capacidad de Desplazamiento:** El reporte es lo suficientemente extenso como para requerir una **barra de desplazamiento vertical**, lo que permite al usuario ver todo el contenido.
- **Acción de Descarga:** Incluye un botón prominentemente ubicado con el texto "DESCARGAR REPORTE", que ofrece la posibilidad de guardar una copia de esta información.
- **Diseño Consistente:** Mantiene el estilo oscuro y minimalista de las otras ventanas de la aplicación, con un botón de cierre y la opción de "AYUDA CHATGPT" visibles, lo que refuerza la unidad visual de la interfaz.

MODULO CHATGPT

Descripción:

La imagen muestra un chat general donde el usuario podrá hacer uso de la IA (Intelegencia Artificial) para tener una respuesta o ayuda de como poder dar solución a los errores encontrados en el desarrollo por las herramientas o servicios utilizados.

El chat será una ventana emergente situada en la parte izquierda abajo de la pantalla.

COMPONENTES DE CALIDAD DE SOFTWARE

A continuación, se describen los componentes de calidad de software que definen el desempeño y las características del prototipo, destacando su diseño y funcionalidad. Esta lista abarca aspectos esenciales como la experiencia del usuario, la protección de datos, la facilidad de mantenimiento, la adaptabilidad a diferentes entornos y la optimización del rendimiento, proporcionando una visión integral de las capacidades del aplicativo en un contexto de arquitectura de microservicios escalable y segura.

- **Usabilidad:** El aplicativo tiene una interfaz limpia y directa, accesible desde cualquier navegador, sin necesidad de registro.
- **Seguridad:** No se almacena información sensible del usuario. Se emplean protocolos de cifrado.
- **Mantenibilidad:** Arquitectura modular basada en microservicios que permite reemplazar o actualizar componentes.
- **Portabilidad:** Compatible con múltiples entornos (nube y local).
- **Eficiencia del desempeño:** Optimización del procesamiento de análisis incluso en escenarios de alta concurrencia.

CONCLUSIÓN

El desarrollo del prototipo del aplicativo web QAScan para el análisis de prácticas correctas en la codificación de desarrollados enfocado en ciberseguridad para pequeñas empresas ha demostrado ser una solución efectiva, pertinente y viable frente a los desafíos actuales del entorno digital en las empresas colombianas. A lo largo del proyecto se abordaron con éxito cada una de las etapas del ciclo de vida del software, desde la identificación de requerimientos específicos del sector Pyme en Colombia hasta la implementación de una herramienta funcional, intuitiva para el usuario y técnicamente robusta.

El aplicativo desarrollado permite detectar vulnerabilidades en etapas tempranas del desarrollo, analizar vulnerabilidades críticas y generar reportes claros que ayuden a los usuarios, todo ello a través de una plataforma de libre acceso y escalable. Esto no solo responde a una necesidad urgente, sino que también representa una oportunidad para fortalecer el conocimiento de desarrolladoras y empresas frente a los crecientes riesgos cibernéticos.

Gracias a la adopción de las metodologías ágiles, la integración de herramientas reconocidas en el ámbito de la ciberseguridad y una arquitectura de microservicios adaptable, se garantizó una solución sostenible en el tiempo. Las pruebas realizadas evidenciaron un desempeño eficaz del aplicativo en escenarios locales, reafirmando su aplicabilidad en entornos empresariales con recursos limitados.

En definitiva, el proyecto no solo cumplió con los objetivos planteados, sino que aportó una contribución significativa al fortalecimiento de la seguridad digital de las Pymes, promoviendo una cultura de prevención y eficiencia tecnológica. Este logro refleja el

compromiso, el rigor técnico y la visión social de sus autores, consolidando el proyecto como una propuesta de alto impacto y valor estratégico para el ecosistema empresarial colombiano.

REFERENCIAS

Instituto Nacional de Ciberseguridad (INCIBE). (2022). *Las principales vulnerabilidades de una pyme en materia de ciberseguridad*. Recuperado de https://www.incibe.es/empresas/blog/las-principales-vulnerabilidades-de-una-pyme-en-materia-de-ciberseguridad

ArXiv. (2023). *Integración de soluciones de ciberseguridad para pymes*. Recuperado de https://arxiv.org/abs/2309.17186

Cámara Colombiana de Informática y Telecomunicaciones (CCIT). (2023). *Estudio anual de ciberseguridad 2022-2023.* Recuperado de https://www.ccit.org.co/estudios/estudio-anual-de-ciberseguridad-2022-2023/

CCIT. (2023). *Estudio anual de ciberseguridad 2022-2023*. Recuperado de https://www.ccit.org.co/estudios/estudio-anual-de-ciberseguridad-2022-2023/

ENISA. (2023). *Cybersecurity Guide for SMEs.* Recuperado de https://www.enisa.europa.eu/publications/cybersecurity-guide-for-smes

Impacto TIC. (2023). *Ciberseguridad para Pymes en Colombia: desafíos y soluciones.* Recuperado de https://impactotic.co/ciber-seguridad/ciberseguridad-para-pymes-de-colombia/

ImpactoTIC. (2023). *Ciberseguridad para pymes de Colombia.* Recuperado de https://impactotic.co/ciber-seguridad/ciberseguridad-para-pymes-de-colombia/

Instituto Nacional de Ciberseguridad (INCIBE). (2022). *Informe de ciberseguridad: Amenazas y recomendaciones para PYMEs*. Recuperado de https://www.incibe.es/informes/informe-ciberseguridad-amenazas-recomendaciones-pymes

Instituto Nacional de Ciberseguridad (INCIBE). (2022). *Las principales vulnerabilidades de una pyme en materia de ciberseguridad.* Recuperado de https://www.incibe.es/empresas/blog/las-principales-vulnerabilidades-de-una-pyme-en-materia-de-ciberseguridad

Kaspersky. (2023). *Boletín de seguridad de Kaspersky, estadísticas de 2023.* Recuperado de https://securelist.lat/ksb-2023-statistics/98257/

Misión Pyme. (2023). *Pymes, las más afectadas por los ciberataques.* Recuperado de https://misionpyme.com/pymes-4-0/pymes-las-mas-afectadas-por-los-ciberataques/

NIST. (2023). *Small Business Cybersecurity.* Recuperado de https://www.nist.gov/itl/smallbusinesscyber

Rombaldo Junior, C., Becker, I., & Johnson, S. (2023). *Unaware, Unfunded and Uneducated: A Systematic Review of SME Cybersecurity.* Recuperado de https://arxiv.org/abs/2309.17186

SafePymes. (2023). *Plataforma de ciberseguridad para pymes*. Recuperado de https://www.ccit.org.co/safepymes/

Small Business Administration (SBA). (2023). *Cybersecurity guidance for small businesses.* Recuperado de https://www.sba.gov/business-guide/manage-your-business/cybersecurity-threats

Small Business Administration (SBA). (2023). *Refuerce su ciberseguridad.* Recuperado de https://www.sba.gov/es/guia-de-negocios/administre-su-empresa/refuerce-su-ciberseguridad