**SentinelCore: Plataforma Modular de Seguridad como Servicio (SaaS)**

**1. Visión General**

**SentinelCore** es una plataforma SaaS innovadora diseñada para pequeñas y medianas empresas (PYMEs) y servicios técnicos, centralizando la gestión de vulnerabilidades y servicios de seguridad en entornos web y cloud. Desplegada completamente en la nube, elimina la necesidad de instalación local, ofreciendo accesibilidad a través de un dashboard web y la API de WhatsApp Business. Su arquitectura modular, basada en contenedores Docker, y el concepto único de “toquen de antesala” proporcionan una solución proactiva, escalable y educativa, diferenciándose de los antivirus tradicionales.

**Objetivos del Proyecto**

* **Centralización**: Integrar herramientas de seguridad open-source (OWASP ZAP, Wapiti, ClamAV) y servicios de partners en un solo panel.
* **Proactividad**: Implementar “toquens de antesala” para analizar archivos antes del almacenamiento, previniendo amenazas.
* **Accesibilidad**: Facilitar la gestión mediante un dashboard web y comandos vía WhatsApp, ideal para usuarios no técnicos.
* **Extensibilidad**: Mapear y gestionar servicios externos, permitiendo colaboración con fabricantes y partners.
* **Educación**: Incluir un módulo de aprendizaje con guías interactivas para PYMEs y equipos técnicos.

**Público Objetivo**

* **PYMEs**: Empresas con recursos limitados que necesitan soluciones de seguridad accesibles y fáciles de usar.
* **Servicios Técnicos**: Equipos de TI gestionando múltiples clientes, requiriendo control centralizado y comunicación ágil.
* **Entornos Educativos**: Instituciones que buscan herramientas prácticas para formación en ciberseguridad.

**2. Análisis de Mercado Mundial**

**Contexto Global**

El mercado global de ciberseguridad está en rápido crecimiento debido al aumento de ciberataques, la digitalización y la adopción de entornos cloud. Según un informe de **MarketsandMarkets** (2025), el mercado de ciberseguridad alcanzará los **USD 317 mil millones para 2027**, con una tasa de crecimiento anual compuesta (CAGR) del **10.9%** desde 2022. La demanda de soluciones SaaS y servicios gestionados está impulsada por PYMEs, que representan el **43% del mercado** en regiones como América del Norte, Europa y Asia-Pacífico.

**Mercados y Países con Alta Viabilidad**

1. **América del Norte (EE.UU., Canadá)**:
   * **Oportunidad**: Alta adopción de soluciones SaaS (68% de las empresas usan SaaS según **Gartner**). PYMEs buscan herramientas integradas para cumplir normativas como **CCPA** y **HIPAA**.
   * **Viabilidad**: Infraestructura cloud madura (AWS, Azure). Alta demanda de soluciones accesibles para servicios técnicos.
   * **Gráfico**:
   * pie title Mercado SaaS en América del Norte
   * "PYMEs": 43
   * "Grandes Empresas": 35
   * "Sector Público": 22
2. **Europa (Alemania, Reino Unido, España)**:
   * **Oportunidad**: Regulaciones estrictas como **GDPR** impulsan la demanda de herramientas de cumplimiento. España tiene un crecimiento del **12% anual** en ciberseguridad (INCIBE, 2025).
   * **Viabilidad**: PYMEs europeas prefieren soluciones cloud sin instalación local. Alta adopción de WhatsApp como canal de comunicación (80% de penetración en España).
   * **Gráfico**:
   * bar title Crecimiento Ciberseguridad en Europa
   * "Alemania": 15
   * "Reino Unido": 13
   * "España": 12
   * "Otros": 10
3. **Asia-Pacífico (India, Singapur, Australia)**:
   * **Oportunidad**: Crecimiento del **15% CAGR** en ciberseguridad (Frost & Sullivan, 2025). India tiene un auge de PYMEs digitalizadas.
   * **Viabilidad**: WhatsApp es ampliamente usado (India: 500M usuarios). Infraestructura cloud en expansión (AWS Asia, Google Cloud).
   * **Gráfico**:
   * pie title Adopción WhatsApp en Asia-Pacífico
   * "India": 50
   * "Singapur": 20
   * "Australia": 15
   * "Otros": 15
4. **Latinoamérica (Brasil, México, Colombia)**:
   * **Oportunidad**: Aumento de ciberataques (+20% en 2024, Fortinet). PYMEs buscan soluciones económicas y fáciles de implementar.
   * **Viabilidad**: Alta penetración de WhatsApp (98% en Brasil). Creciente adopción de soluciones SaaS (Mercado Libre Cloud, AWS).
   * **Gráfico**:
   * bar title Crecimiento SaaS en LATAM
   * "Brasil": 18
   * "México": 15
   * "Colombia": 12
   * "Otros": 10

**Posicionamiento de SentinelCore**

* **Diferenciador**: Solución SaaS con integración WhatsApp, toquens Dockerizados y enfoque en PYMEs.
* **Competencia**: Nessus, OpenVAS, Qualys ofrecen escaneos, pero carecen de comunicación multimodal y prevención proactiva como los toquens.
* **Ventaja**: Centralización, accesibilidad vía WhatsApp, y capacidad de integrar servicios de partners, ideal para mercados emergentes y regulaciones estrictas.

**3. Innovación y Características Clave**

**Toquens Dockerizados**

* **Definición**: Microservicios independientes que analizan archivos o servicios en contenedores Docker, con el “toquen de antesala” como filtro inicial.
* **Innovación**: Previene amenazas antes del almacenamiento, a diferencia de soluciones reactivas.
* **Ejemplo**: Un archivo PDF es interceptado por antesala.py, analizado por pdf\_sanitizer.py en un contenedor Docker, y sanitizado antes de guardarse.
* **Gráfico**:
* sequenceDiagram
* participant U as Usuario
* participant A as Toquen Antesala
* participant S as Toquen Especializado
* U->>A: Subir archivo (doc.pdf)
* A->>S: Delegar análisis (PDF)
* S-->>A: Resultado (Seguro/Inseguro)
* A-->>U: Notificación vía WhatsApp

**Escalabilidad y Modularidad**

* **Escalabilidad**: Docker permite añadir instancias de toquens bajo demanda. Redis asegura comunicación robusta.
* **Modularidad**: Nuevos toquens (e.g., image\_analyzer.py) o servicios de partners (e.g., AWS Security Hub) se integran sin modificar el núcleo.
* **Gráfico**:
* graph TD
* A[Core Module] --> B[Toquen Module]
* A --> C[Web Scanner Module]
* A --> D[Partner APIs]
* B --> E[New Toquen]
* C --> F[New Scanner]
* D --> G[New Partner]

**Módulo de Aprendizaje**

* **Propósito**: Proporcionar guías interactivas y simulaciones de vulnerabilidades en el dashboard.
* **Implementación**: Integrar tutoriales basados en OWASP Top 10 y MITRE ATT&CK, accesibles vía web y WhatsApp.
* **Beneficio**: Ideal para PYMEs con personal no técnico y entornos educativos.

**Integración con WhatsApp**

* **Implementación**: Usar WhatsApp Business API para notificaciones ("XSS detectado en http://example.com") y comandos (/start\_scan, /generate\_report).
* **Código Ejemplo**:
* from fastapi import APIRouter
* from twilio.rest import Client
* router = APIRouter()
* @router.post("/whatsapp/alert")
* async def send\_alert(to: str, message: str):
* client = Client("account\_sid", "auth\_token")
* client.messages.create(
* body=f"🔒 SentinelCore Alert: {message}",
* from\_="whatsapp:+14155238886",
* to=f"whatsapp:{to}"
* )
* **Beneficio**: Comunicación ágil para técnicos en campo y usuarios no técnicos.

**Gestión de Servicios Externos**

* **Service Mapping**: El service\_manager.py mapea servicios gestionados (web, cloud, endpoints) y activa/desactiva herramientas de partners.
* **Ejemplo**: Activar un escaneo de VirusTotal y mostrar resultados en el dashboard.
* **Gráfico**:
* graph TD
* A[Dashboard] --> B[Service Manager]
* B --> C[OWASP ZAP]
* B --> D[VirusTotal]
* B --> E[AWS Security Hub]
* E --> F[Resultados al Dashboard]

**4. Arquitectura Modular**

**Módulos Funcionales**

1. **Core Module**:
   * **Propósito**: Orquesta la comunicación y gestiona configuraciones.
   * **Componentes**: main.py, dispatcher.py, service\_manager.py, config/.
   * **Tecnologías**: Python 3.8+, asyncio, Redis, Docker.
2. **Toquen Module**:
   * **Propósito**: Analiza archivos antes del almacenamiento con toquens especializados.
   * **Componentes**: antesala.py, static\_scan.py, pdf\_sanitizer.py.
   * **Tecnologías**: libmagic, ClamAV, asyncio.Queue, Docker.
3. **Web Scanner Module**:
   * **Propósito**: Integra herramientas de escaneo web.
   * **Componentes**: zap\_wrapper.py, results\_parser.py.
   * **Tecnologías**: requests, OWASP ZAP API, Wapiti.
4. **Database Module**:
   * **Propósito**: Almacena resultados y recomendaciones.
   * **Componentes**: models.py.
   * **Tecnologías**: SQLite, sqlite3.
5. **GUI Module**:
   * **Propósito**: Proporciona dashboard web y soporte WhatsApp.
   * **Componentes**: dashboard.py, whatsapp\_api.py.
   * **Tecnologías**: FastAPI, Vue.js, Tailwind CSS, WhatsApp Business API.
6. **Mitigation Module**:
   * **Propósito**: Genera y aplica mitigaciones.
   * **Componentes**: mitigation\_engine.py.
   * **Tecnologías**: reportlab, OWASP/MITRE ATT&CK.

**Estructura de Carpetas (en evolución)**

sentinelcore/

├── api/

│ ├── main.py # FastAPI entry point

│ ├── routes.py # API endpoints

│ ├── auth.py # Authentication logic

│ └── Dockerfile

├── core/

│ ├── dispatcher.py # Task orchestration

│ ├── service\_manager.py # External service mapping

│ ├── config/

│ │ ├── settings.yaml

│ │ └── api\_keys.json

│ └── Dockerfile

├── toquens/

│ ├── antesala/

│ │ ├── scanner.py # Toquen de antesala logic

│ │ ├── utils.py

│ │ └── Dockerfile

│ ├── pdf/

│ │ ├── sanitizer.py # PDF sanitization

│ │ └── Dockerfile

│ ├── static/

│ │ ├── static\_scan.py # Static file analysis

│ │ └── Dockerfile

├── webscanner/

│ ├── zap\_wrapper.py # OWASP ZAP integration

│ ├── results\_parser.py # Scan result processing

│ └── Dockerfile

├── partners/

│ ├── aws\_security.py # AWS Security Hub integration

│ ├── clamav\_api.py # ClamAV integration

│ └── Dockerfile

├── frontend/

│ ├── public/

│ ├── src/

│ │ ├── views/ # Vue.js views

│ │ └── assets/

│ └── Dockerfile

├── reports/

│ ├── report\_generator.py # HTML/PDF report generation

│ ├── consultant\_integration.py # Consultant reports

│ └── Dockerfile

├── docs/

│ ├── architecture.md # Architecture documentation

│ ├── api\_specs/ # OpenAPI/Swagger

│ ├── user\_guide.md

│ └── manual\_usuario.md

├── tests/

│ ├── test\_api.py

│ ├── test\_toquens.py

│ ├── test\_webscanner.py

│ ├── test\_dashboard.py

│ └── test\_reports.py

├── docker-compose.yml

└── Dockerfile\_base

* **Nota**: La estructura es evolutiva, con planes para añadir nuevos toquens (e.g., toquens/image\_analyzer/) y conectores de partners (e.g., partners/azure\_defender.py).

**Diagrama de Flujo (en evolución)**

graph TD

A[Usuario/Técnico] -->|Dashboard Web| B[GUI Module]

A -->|WhatsApp API| B

B -->|REST API| C[Core Module]

C -->|Dispatch Task| D[Toquen Module]

C -->|Dispatch Task| E[Web Scanner Module]

C -->|Service Mapping| F[Service Manager]

D -->|File Analysis| G[Toquen Antesala]

G -->|Delegate| H[Toquens Especializados]

E -->|Scan Results| I[Database Module]

I -->|Mitigation Data| J[Mitigation Module]

J -->|Reports| B

F -->|External Services| K[Partner APIs]

K -->|Results| B

* **Nota**: El diagrama es evolutivo, con futuras integraciones de conectores cloud (AWS, Azure) y nuevos toquens.

**5. Viabilidad del Proyecto**

**Viabilidad Técnica**

* **Modularidad**: Módulos independientes en contenedores Docker, asegurando aislamiento y escalabilidad.
* **Integración**: Uso de herramientas open-source (OWASP ZAP, Wapiti, ClamAV) con APIs bien documentadas.
* **Cloud-Hosted**: Desplegada en Heroku/AWS, accesible vía API REST y dashboard web.
* **WhatsApp Integration**: Uso de WhatsApp Business API para notificaciones y comandos.
* **Gráfico**:
* pie title Tecnologías Utilizadas
* "Python/FastAPI": 40
* "Docker": 30
* "WhatsApp API": 15
* "Vue.js/Tailwind": 10
* "SQLite": 5

**Viabilidad Organizativa**

* **Equipo**: 4 personas con roles definidos:
  + **Líder Técnico**: Core Module, Toquen Antesala, Service Manager.
  + **Desarrollador de Escáneres**: Web Scanner, Toquens Especializados.
  + **Desarrollador de Interfaz**: GUI Module (dashboard, WhatsApp).
  + **Especialista en Base de Datos/Mitigación**: Database, Mitigation.
* **Metodología**: Scrum con sprints de 2 semanas, usando GitHub, Trello, y Discord.
* **Capacitación**: 2 semanas iniciales para Docker, FastAPI, y WhatsApp API.

**Viabilidad Económica**

* **Costo Bajo**: Uso de herramientas open-source y servicios cloud gratuitos (Heroku, AWS Free Tier).
* **Limitaciones**: Evitar APIs comerciales (Nessus, Qualys) en el MVP.
* **Gráfico**:
* bar title Presupuesto Estimado
* "Desarrollo": 60
* "Infraestructura Cloud": 20
* "Capacitación": 15
* "Documentación": 5

**Riesgos y Mitigaciones**

* **Riesgo**: Complejidad en integración de APIs.
  + **Mitigación**: Usar herramientas con documentación clara, pruebas unitarias con pytest.
* **Riesgo**: Curva de aprendizaje para Docker/WhatsApp API.
  + **Mitigación**: Capacitación inicial, tutoriales prácticos.
* **Riesgo**: Seguridad de la plataforma.
  + **Mitigación**: Autenticación robusta (API keys, OAuth), validación de entradas, auditorías.

**6. Plan de Ejecución**

**Cronograma (6 meses)**

* **Mes 1**: Planificación, capacitación (Docker, FastAPI, WhatsApp API), diseño de arquitectura.
* **Mes 2-3**: Desarrollo de Core, Toquens, Web Scanner (Dockerized).
* **Mes 4**: Implementación de Database, GUI, WhatsApp integration.
* **Mes 5**: Mitigation Module, service mapping, pruebas.
* **Mes 6**: Pruebas finales, documentación, presentación.
* **Gráfico**:
* gantt
* title Cronograma SentinelCore
* dateFormat YYYY-MM
* section Planificación
* Capacitación y Diseño :done, 2025-07, 1mo
* section Desarrollo
* Core y Toquens :active, 2025-08, 2mo
* Database y GUI :2025-10, 1mo
* Mitigation y Mapping :2025-11, 1mo
* section Pruebas
* Pruebas y Documentación :2025-12, 1mo

**Metodología**

* **Scrum**: Sprints de 2 semanas, reuniones diarias de 15 minutos.
* **Herramientas**:
  + **GitHub**: Control de versiones.
  + **Trello**: Gestión de tareas.
  + **Discord**: Comunicación.
  + **pytest**: Pruebas unitarias e integración.

**7. Checklist de Tareas**

**Planificación**

* Definir arquitectura modular y servicios gestionados.
* Investigar WhatsApp Business API y FastAPI.
* Configurar repositorio GitHub y docker-compose.yml.

**Desarrollo**

* Implementar Core Module (main.py, dispatcher.py, service\_manager.py).
* Desarrollar Toquen Module (antesala/scanner.py, pdf/sanitizer.py) en Docker.
* Integrar Web Scanner Module con OWASP ZAP y Wapiti.
* Configurar Database Module con SQLite.
* Crear GUI Module (dashboard web con Vue.js, whatsapp\_api.py).
* Implementar Mitigation Module con reportes HTML/PDF.

**Pruebas**

* Pruebas unitarias (pytest) para cada módulo.
* Pruebas de integración (Core-Toquen-Web Scanner).
* Simulación de ataques (XSS, SQLi) para validar detección.
* Pruebas de usabilidad (dashboard, WhatsApp).

**Documentación**

* Manual de usuario con guías educativas.
* Documentación técnica (arquitectura, APIs, OpenAPI/Swagger).
* Registro de mejoras y problemas.

**8. Ventajas Competitivas**

| **Característica** | **Competencia (Nessus, OpenVAS)** | **SentinelCore** |
| --- | --- | --- |
| **Implementación** | Local/SaaS | 100% Cloud (API remota) |
| **Comunicación** | Email/Web | WhatsApp + Dashboard |
| **Toquens** | No existe | Dockerizados y especializados |
| **Integración Partners** | Limitada | API abierta para fabricantes |
| **Target** | Grandes empresas | PYMEs y servicios técnicos |
| **Educación** | Genérica | Módulo de aprendizaje |

* **Gráfico**:
* pie title Ventajas Competitivas
* "Centralización": 25
* "Proactividad": 20
* "Accesibilidad": 20
* "Extensibilidad": 20
* "Educación": 15

**9. Conclusión y Próximos Pasos**

**SentinelCore** redefine la ciberseguridad para PYMEs y servicios técnicos con una solución SaaS modular, proactiva y accesible. Los toquens Dockerizados, la integración con WhatsApp, y la gestión de servicios externos posicionan a SentinelCore como una herramienta innovadora y escalable. Su enfoque educativo y facilidad de uso la hacen ideal para mercados emergentes y entornos regulados.

**Próximos Pasos**

* **Fase Alpha (2 meses)**: Dockerizar toquens básicos, integrar OWASP ZAP y WhatsApp API.
* **Fase Beta (3 meses)**: Desarrollar dashboard web, incorporar partners (ClamAV, AWS Security Hub).
* **Lanzamiento (1 mes)**: Pruebas finales, documentación multilingüe, presentación a clientes.

**Preguntas Clave para la Empresa**

1. ¿Qué integraciones con partners son prioritarias para el mercado objetivo?
2. ¿Es viable incluir Telegram como canal adicional a WhatsApp?
3. ¿Qué normativas específicas (e.g., GDPR, CCPA) deben priorizarse en el MVP?

**10. Anexos**

* **Prototipo Figma**: Diseño preliminar del dashboard web.
* **Repositorio GitHub**: Incluye Dockerfiles y documentación inicial.
* **Documentación API**: Especificaciones OpenAPI/Swagger en docs/api\_specs/.