Universidad Distrital Francisco José de Caldas
Facultad de Ingeniería

Herramienta de Análisis de IP con Fuentes Abiertas

Proyecto de Investigación

Investigadores:

Juan Manuel Serrano Rodríguez
Código: 20211020091
jumserranor@udistrital.edu.co

Nicolás Guevara Herrán
Código: 20211180026
nguevarah@udistrital.edu.co

Área de Investigación:
Ciberseguridad y Análisis de Redes

Grupo de Investigación:
Modelamiento en Ingeniería de Sistemas

Director:
Lilian Astrid Bejarano Garzón

11 de noviembre de 2025

1. Selección y Definición del Tema de Investigación

1.1. Tema de Investigación

"Desarrollo de una Herramienta de Diagnóstico Unificado para el Análisis de Seguridad y Geolocalización de Direcciones IP mediante Integración de Fuentes Abiertas en el Contexto de Ciberseguridad Colombiana"

1.2. Título Provisional

"Diagnóstico de Seguridad IP con Fuentes Abiertas en Colombia"

1.3. Línea de Investigación

Este proyecto se enmarca dentro de la línea de investigación en **Inteligencia de Amenazas Cibernéticas**, específicamente en el área de **Análisis Automatizado de Infraestructuras de Red** y **Desarrollo de Herramientas de Ciberseguridad Basadas en Fuentes Abiertas**.

1.4. Área del Conocimiento

- **Área principal:** Ingeniería de Sistemas y Computación
- **Subárea:** Ciberseguridad y Redes de Computadores
- **Disciplina:** Inteligencia de Amenazas y Análisis de Vulnerabilidades

2. Planteamiento del Problema

2.1. Contexto del Problema

En el día a día de la ciberseguridad y el desarrollo de software, es común necesitar información detallada sobre una dirección IP. Un analista puede querer saber su ubicación geográfica, si está en una lista negra, qué puertos tiene abiertos o a qué organización pertenece.

Actualmente, esta información se encuentra dispersa en múltiples fuentes de datos, tanto gratuitas como de pago. Por ejemplo:

- **Geolocalización:** Se obtiene de bases de datos como GeoLite2 de MaxMind.

- **Información de Infraestructura:** Se consulta en servicios de escaneo de Internet como Censys o Shodan.
- **Listas de Reputación:** Se verifica en plataformas como AbuseIPDB o listas de bloqueo (blocklists).

2.2. Definición del Problema Principal

El problema central es la **fragmentación de la información**. Un desarrollador o analista que necesita un perfil completo de una dirección IP debe realizar las siguientes acciones manualmente:

1. Consultar la API o base de datos de geolocalización.
2. Consultar la API de un servicio como Censys para obtener datos de puertos y servicios.
3. Consultar una o más APIs de listas de reputación.
4. Consolidar y correlacionar toda esta información a mano para poder tomar una decisión.

Este proceso manual es **ineficiente, lento y propenso a errores**. Limita la capacidad de realizar análisis rápidos y automatizados, especialmente cuando se necesita evaluar un gran número de direcciones IP. Aunque existen herramientas comerciales que resuelven este problema, sus costos son a menudo prohibitivos para estudiantes, desarrolladores independientes o pequeños equipos.

2.3. Pregunta de Investigación

¿Es posible diseñar y desarrollar una herramienta de software que integre y unifique datos de fuentes abiertas y gratuitas (específicamente GeoLite2 y Censys a través de BigQuery) para presentar un diagnóstico de seguridad consolidado de una dirección IP, de manera eficiente y accesible?

2.4. Justificación de la Investigación

La creación de esta herramienta aborda directamente la ineficiencia del proceso actual. Al automatizar la recolección y presentación de datos, el proyecto ofrece una solución práctica que:

- **Ahorra tiempo:** Reduce drásticamente el tiempo necesario para investigar una IP.
- **Centraliza la información:** Ofrece un "panel únicoçon los datos más relevantes.
- **Es accesible:** Al basarse en fuentes gratuitas (GeoLite2 y las cuotas gratuitas de BigQuery), la solución es económicamente viable para un público amplio.
- **Tiene un propósito educativo:** Sirve como un excelente caso de estudio sobre cómo integrar diferentes APIs y fuentes de datos en una aplicación de ciberseguridad funcional.

Este proyecto, aunque de alcance limitado a dos fuentes principales, sienta las bases para una herramienta más completa y demuestra la viabilidad de construir soluciones de ciberseguridad efectivas sin depender exclusivamente de costosas licencias comerciales.

2.4.1. Análisis de Causas

Con el fin de comprender de forma estructurada las raíces del problema identificado —la *fragmentación de la información de inteligencia sobre IPs en el contexto colombiano*— se aplicó la metodología de Ishikawa (o diagrama de causa-efecto). Esta herramienta permitió clasificar las causas en cuatro grandes categorías: tecnológicas, económicas, humanas y regulatorias, lo cual facilita una visualización clara de los factores que convergen en el problema central.

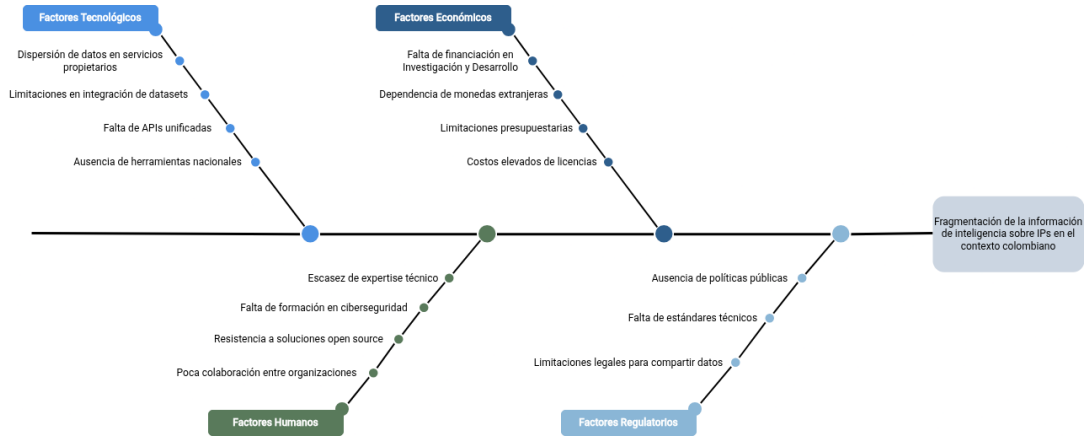

Figura 1: Diagrama de Ishikawa de la fragmentación de la información de inteligencia sobre IPs

A partir del análisis realizado, se identificaron las siguientes causas específicas:

Factores Tecnológicos:

- Dispersión de datos de inteligencia en servicios propietarios
- Falta de APIs unificadas para consulta de múltiples fuentes
- Limitaciones técnicas en la integración de datasets heterogéneos
- Ausencia de herramientas nacionales especializadas

Factores Económicos:

- Costos elevados de licencias para herramientas comerciales
- Limitaciones presupuestarias en organizaciones públicas y PYMES
- Dependencia de monedas extranjeras para servicios internacionales
- Falta de financiación para I+D en ciberseguridad nacional

Factores Humanos:

- Escasez de expertise técnico especializado
- Falta de programas de formación en desarrollo de herramientas de ciberseguridad
- Resistencia al cambio hacia soluciones open source
- Limitada cultura de colaboración entre organizaciones

Factores Regulatorios:

- Ausencia de políticas públicas para fomento de desarrollo tecnológico nacional
- Falta de estándares técnicos para herramientas de ciberseguridad
- Limitaciones en marcos legales para compartir información de amenazas

2.4.2. Pronóstico

Si persiste esta situación, se proyectan las siguientes consecuencias para el período 2025-2027:

Impacto Operacional:

- Incremento del 40 % en tiempos de respuesta ante incidentes de seguridad
- Reducción del 25 % en la efectividad de detección de amenazas

- Aumento del 60 % en costos operativos por múltiples licencias

Impacto Estratégico:

- Mayor dependencia de proveedores extranjeros
- Limitación en capacidades de soberanía digital nacional

Impacto Económico:

- Reducción de la competitividad de organizaciones colombianas
- Limitación en adopción de tecnologías emergentes por riesgos de seguridad

2.4.3. Control al Pronóstico

Para evitar este escenario negativo, es fundamental desarrollar una solución tecnológica que:

1. **Integre fuentes abiertas gratuitas** como Censys (disponible en Google BigQuery) y MaxMind GeoLite2 para proporcionar análisis comprehensivos sin costos de licenciamiento.
2. **Implemente algoritmos de correlación avanzados** que combinen reconocimiento pasivo con geolocalización para generar evaluaciones de riesgo contextualizadas.
3. **Proporcione interfaces intuitivas** que democraticen el acceso a capacidades avanzadas de análisis IP.
4. **Establezca metodologías reproducibles** que puedan ser adoptadas por organizaciones con diferentes niveles de madurez tecnológica.
5. **Genere capacidades técnicas locales** mediante documentación detallada y código abierto que facilite la transferencia de conocimiento.

3. Objetivos de la Investigación

3.1. Objetivo General

Diseñar, desarrollar y documentar una herramienta de software funcional que integre datos de geolocalización de la base de datos GeoLite2 y datos de infraestructura de Censys (vía BigQuery) para presentar un perfil unificado y de fácil consulta sobre una dirección IP.

3.2. Objetivos Específicos

1. **Diseñar una arquitectura de software simple** para la herramienta, basada en un backend (Quarkus) que centralice la lógica de negocio y un frontend (Vue.js) que consuma y presente los datos al usuario.

2. **Implementar la lógica en el backend** para que sea capaz de:

   - Recibir una dirección IP como entrada.
   - Consultar la base de datos local de GeoLite2 para obtener la información de geolocalización.
   - Ejecutar una consulta a la base de datos de Censys en Google BigQuery para obtener información de puertos y servicios.
   - Combinar los resultados de ambas fuentes en una única respuesta estructurada (JSON).

3. **Desarrollar una interfaz de usuario básica** en el frontend que permita:

   - Ingresar una dirección IP en un campo de texto.
   - Enviar la solicitud al backend.
   - Presentar de forma clara y ordenada la información consolidada recibida del backend.

4. **Documentar el proyecto**, incluyendo los pasos para la configuración del entorno de desarrollo, la instalación de la herramienta y una guía básica de uso de la API y la interfaz de usuario.

4. Justificación de la Investigación

4.1. Justificación Teórica

4.1.1. Contribución al Conocimiento Científico

Esta investigación contribuye al campo de la ciberseguridad mediante el desarrollo de una herramienta práctica que integra fuentes abiertas de datos para el análisis de direcciones IP. Como proyecto académico, busca explorar la viabilidad de combinar datos públicos disponibles (GeoLite2 y Censys a través de BigQuery) para crear una solución funcional de análisis de seguridad.

El trabajo se enfoca en la implementación práctica de técnicas ya establecidas en la literatura, adaptándolas a un contexto de recursos limitados y fuentes de datos

gratuitas. Se busca validar si es posible crear una herramienta útil para análisis básico de IP utilizando únicamente tecnologías y datos de acceso libre.

4.1.2. Objetivos Académicos

El proyecto tiene como objetivo académico principal desarrollar competencias prácticas en:

- **Integración de Datos:** Uso de APIs y bases de datos públicas para ciberseguridad
- **Desarrollo Web:** Implementación de aplicaciones usando tecnologías modernas
- **Análisis de Seguridad:** Aplicación práctica de conceptos de reconocimiento pasivo
- **Investigación Aplicada:** Documentación y evaluación sistemática de resultados

4.2. Justificación Metodológica

4.2.1. Enfoque Práctico

La metodología se basa en el desarrollo incremental de una aplicación web que integre datos de fuentes públicas confiables. Se utilizará un stack tecnológico moderno pero accesible:

Fuentes de Datos:

- **MaxMind GeoLite2:** Base de datos gratuita de geolocalización IP
- **Censys via BigQuery:** Datos de escaneo de Internet disponibles públicamente

Stack Tecnológico:

- **Backend:** Quarkus (Java) para APIs y lógica de negocio
- **Frontend:** Vue.js para interfaz de usuario
- **Datos:** Google BigQuery para consultas eficientes
- **Infraestructura:** Servicios cloud básicos para despliegue

4.2.2. Validación Práctica

La validación se realizará mediante:

- Pruebas funcionales de la aplicación desarrollada
- Comparación básica con herramientas existentes
- Evaluación de usabilidad con usuarios del entorno académico
- Documentación de lecciones aprendidas y limitaciones encontradas

4.3. Justificación Práctica

4.3.1. Aplicabilidad en el Entorno Académico

La herramienta desarrollada tendrá aplicación directa en:
Educación en Ciberseguridad:

- Material práctico para cursos de seguridad informática
- Ejemplo de integración de datos para análisis de amenazas
- Caso de estudio para desarrollo de aplicaciones de seguridad

Investigación Estudiantil:

- Base para futuros proyectos de grado relacionados
- Herramienta para análisis básico en investigaciones de seguridad
- Ejemplo de implementación con recursos limitados

4.3.2. Contribución al Programa Académico

El proyecto contribuye al programa de estudios mediante:

- **Aplicación Práctica:** Implementación real de conceptos teóricos estudiados
- **Tecnologías Actuales:** Experiencia con herramientas y frameworks modernos
- **Metodología de Desarrollo:** Aplicación de buenas prácticas de ingeniería de software
- **Documentación Técnica:** Desarrollo de habilidades de comunicación técnica

4.3.3. Viabilidad y Alcance Realista

El proyecto está diseñado para ser completado exitosamente en el tiempo disponible:

Recursos Disponibles:

- Acceso a datasets públicos sin costo
- Créditos académicos para servicios cloud
- Herramientas de desarrollo gratuitas o con licencias estudiantiles
- Supervisión académica y acceso a recursos universitarios

Limitaciones Reconocidas:

- Alcance limitado a fuentes de datos gratuitas
- Funcionalidades básicas en la versión inicial
- Evaluación limitada al entorno académico
- No pretende competir con soluciones comerciales especializadas

4.3.4. Impacto Esperado

Se espera que el proyecto genere:

- Una herramienta funcional para análisis básico de IP
- Documentación técnica que sirva como referencia para futuros estudiantes
- Experiencia práctica en desarrollo de aplicaciones de ciberseguridad
- Código fuente abierto que pueda ser mejorado por la comunidad académica

El proyecto se enfoca en demostrar que es posible crear herramientas útiles de ciberseguridad usando recursos disponibles públicamente, proporcionando una alternativa educativa a las costosas soluciones comerciales.

5. Marco de Referencia

5.1. Marco Teórico

5.1.1. Reconocimiento Pasivo en Ciberseguridad

Conceptos Fundamentales: El reconocimiento pasivo se define como el proceso de recolección de información sobre sistemas objetivo utilizando fuentes públicas disponibles, sin establecer comunicación directa que pueda ser detectada. Esta técnica presenta ventajas clave para el análisis de seguridad:

1. **No detectabilidad:** No genera tráfico hacia el objetivo
2. **Legalidad:** Utiliza información públicamente disponible
3. **Escalabilidad:** Permite análisis de múltiples direcciones IP

Herramientas y Plataformas Relevantes: Las plataformas modernas han democratizado el acceso a datos de reconocimiento masivo:

- **Censys:** Plataforma que proporciona datos de escaneo de Internet mediante BigQuery
- **Shodan:** Motor de búsqueda para dispositivos conectados a Internet
- **VirusTotal:** Agregador de análisis de amenazas y reputación de IP

Para este proyecto, nos enfocaremos en los datos de Censys disponibles a través de Google BigQuery, que ofrecen información histórica y actual sobre servicios expuestos en Internet.

5.1.2. Geolocalización de Direcciones IP

Principios Básicos: La geolocalización IP correlaciona direcciones de red con ubicaciones geográficas utilizando diversos métodos:

- **Registros WHOIS:** Información de asignación de bloques IP
- **Mediciones de latencia:** Correlación entre distancia y tiempo de respuesta
- **Bases de datos comerciales:** Agregación de múltiples fuentes de información

Limitaciones Conocidas:

- Precisión variable según la región geográfica

- Influencia de CDNs y tecnologías de distribución de contenido
- Diferencias en infraestructura de red entre países

MaxMind GeoLite2: Para este proyecto utilizaremos la base de datos GeoLite2 de MaxMind, que ofrece:

- Datos de geolocalización gratuitos actualizados mensualmente
- Cobertura global con precisión estimada del 95 % a nivel país
- Formato MMDB optimizado para consultas rápidas
- API simple para integración en aplicaciones

5.1.3. Inteligencia de Amenazas Cibernéticas

Marcos de Referencia: Los frameworks contemporáneos proporcionan estructura para el análisis de amenazas:

MITRE ATT&CK Framework:

- Taxonomía de técnicas de adversarios basada en observaciones reales
- 14 tácticas principales desde acceso inicial hasta exfiltración
- Base de conocimiento para correlacionar indicadores con técnicas

Pyramid of Pain:

- Jerarquía de indicadores según la dificultad de evasión para atacantes
- Direcciones IP en la base: fáciles de cambiar pero útiles para detección
- TTPs en la cima: difíciles de cambiar y más valiosos para defensa

5.2. Marco Conceptual

5.2.1. Definiciones Operacionales

Reconocimiento Pasivo: Recolección de información sobre direcciones IP utilizando fuentes públicas sin contacto directo con los sistemas objetivo.

Geolocalización IP: Proceso de determinar la ubicación geográfica aproximada de una dirección IP mediante consultas a bases de datos especializadas.

Inteligencia de Amenazas: Información procesada sobre indicadores de compromiso (IOCs) que incluye direcciones IP, dominios y patrones de comportamiento malicioso.

BigQuery: Servicio de almacén de datos de Google Cloud que permite consultas SQL sobre grandes volúmenes de datos, incluyendo datasets públicos de Censys.

Censys: Plataforma que realiza escaneos regulares de Internet y proporciona datos sobre servicios, certificados y configuraciones accesibles públicamente.

GeoLite2: Base de datos gratuita de geolocalización IP proporcionada por MaxMind, actualizada mensualmente.

Indicator of Compromise (IOC): Artefacto digital que indica con alta probabilidad una intrusión o actividad maliciosa, incluyendo direcciones IP sospechosas.

5.2.2. Categorización de Datos IP

Tipos de Información Disponible:

- **Geográfica:** País, región, ciudad, coordenadas aproximadas
- **Red:** ASN, ISP, tipo de organización
- **Servicios:** Puertos abiertos, protocolos, banners de servicio
- **Reputación:** Historial de actividad maliciosa, categorización de amenazas

5.3. Marco Tecnológico

5.3.1. Arquitectura de Datos

Fuentes de Datos Primarias:

1. **Censys via BigQuery:**
   - Datos históricos desde 2015
   - Actualizaciones semanales
   - Información de puertos, servicios y certificados
   - Acceso gratuito con limitaciones de quota
2. **MaxMind GeoLite2:**
   - Base de datos descargable

- Actualizaciones mensuales
- Formato MMDB para consultas eficientes
- Licencia gratuita para uso no comercial

5.3.2. Stack Tecnológico

Backend - Quarkus:

- Framework Java nativo en la nube
- Tiempo de inicio rápido y bajo consumo de memoria
- Integración nativa con APIs REST y bases de datos
- Soporte para contenedores y despliegue cloud

Frontend - Vue.js:

- Framework JavaScript progresivo
- Curva de aprendizaje suave
- Ecosistema rico de componentes
- Herramientas de desarrollo integradas

Infraestructura Cloud:

- Google Cloud Platform para BigQuery
- Servicios de hosting para la aplicación web
- CDN para distribución de contenido estático

5.4. Marco Espacial y Temporal

5.4.1. Contexto del Proyecto

El proyecto se desarrolla en el contexto académico universitario durante el período octubre-diciembre 2025, con las siguientes características:

Alcance Geográfico:

- Enfoque en análisis de IP sin restricción geográfica
- Especial atención a datos relevantes para el contexto colombiano

- Utilización de fuentes de datos globales

Limitaciones Temporales:

- Desarrollo incremental en 8 semanas
- Datos históricos disponibles desde 2015 (Censys)
- Actualizaciones de datos según disponibilidad de fuentes

5.4.2. Consideraciones Técnicas

Limitaciones de Recursos:

- Quotas gratuitas de servicios cloud
- Datos limitados a fuentes públicas gratuitas
- Capacidad de procesamiento según recursos académicos disponibles

Restricciones Éticas y Legales:

- Uso exclusivo de datos públicos
- Respeto a términos de servicio de proveedores de datos
- Implementación de medidas de privacidad por diseño

6. Hipótesis de la Investigación

6.1. Hipótesis Principal

Es factible desarrollar una herramienta de software que, utilizando exclusivamente las fuentes de datos gratuitas de GeoLite2 y Censys (a través de la cuota gratuita de BigQuery), puede proporcionar un perfil de información consolidado sobre una dirección IP que es útil y suficiente para un análisis de seguridad inicial.

6.2. Sub-hipótesis

1. **Viabilidad Técnica:** La integración de una base de datos local (GeoLite2) y una API remota (BigQuery para Censys) en una única aplicación (backend con Quarkus, frontend con Vue.js) es técnicamente realizable dentro del marco de un proyecto estudiantil de corta duración.

2. **Suficiencia de Datos:** La combinación de datos de geolocalización (país, ciudad) y datos de infraestructura (puertos abiertos, servicios) es suficiente para que un analista o desarrollador pueda tomar una decisión informada preliminar sobre la naturaleza de una dirección IP.

3. **Eficiencia del Proceso:** Una herramienta que automatiza la consulta y consolidación de estas dos fuentes de datos reducirá significativamente el tiempo y el esfuerzo manual en comparación con la consulta separada de cada fuente.

6.3. Variables

- **Variable Independiente:** La herramienta de software desarrollada que integra GeoLite2 y Censys.
- **Variable Dependiente:** La utilidad y eficiencia del perfil de IP generado, medida en términos de:
  - **Completitud de la información:** ¿Contiene los datos esperados de ambas fuentes?
  - **Tiempo de respuesta:** ¿Cuánto tarda en generar un informe?
  - **Facilidad de uso:** ¿La información se presenta de manera clara y comprensible?

7. Aspectos Metodológicos

7.1. Tipo de Estudio

7.1.1. Diseño de Investigación

El estudio implementa un **diseño de investigación aplicada** con enfoque práctico-experimental. Se centra en el desarrollo e implementación de una herramienta funcional que integre fuentes de datos públicas para análisis de direcciones IP.

Componentes del Estudio:

- **Desarrollo tecnológico:** Construcción de aplicación web funcional
- **Evaluación funcional:** Pruebas de rendimiento y usabilidad
- **Análisis comparativo:** Validación contra herramientas existentes
- **Documentación:** Registro sistemático del proceso y resultados

7.2. Método de Investigación

7.2.1. Metodología de Desarrollo

Se utilizará una metodología ágil adaptada al contexto académico:
Desarrollo Iterativo:

- Sprints de 1 semana para desarrollo rápido
- Entregas incrementales con funcionalidades básicas
- Pruebas continuas durante el desarrollo
- Documentación paralela al código

Arquitectura Orientada a Servicios:

- Separación clara entre frontend y backend
- APIs REST para comunicación entre componentes
- Integración modular con fuentes de datos externas

7.3. Fuentes y Técnicas para Recolección de Información

7.3.1. Fuentes de Datos Primarias

Datasets Técnicos:

1. **Censys via BigQuery:**
   - Datos de escaneo de Internet actualizados semanalmente
   - Información de puertos, servicios y certificados
   - Acceso a través de consultas SQL estándar
   - Quota gratuita de 1TB mensual para consultas
2. **MaxMind GeoLite2:**
   - Base de datos descargable en formato MMDB
   - Actualizaciones mensuales automáticas
   - Cobertura global con precisión variable por región
   - Licencia gratuita para uso académico

7.3.2. Técnicas de Recolección

Integración Automatizada:

- **APIs REST:** Consultas programáticas a servicios externos
- **SQL Queries:** Extracción eficiente de datos desde BigQuery
- **File Processing:** Lectura local de bases de datos MMDB
- **Caching:** Almacenamiento temporal para optimizar rendimiento

Validación de Datos:

- Verificación de formato y consistencia de datos
- Filtrado de información irrelevante o duplicada
- Manejo de errores y datos faltantes
- Logging de actividades para auditoría

7.4. Arquitectura del Sistema

7.4.1. Componentes Principales

Backend (Quarkus):

- **API Gateway:** Punto de entrada único para solicitudes
- **Servicio de Geolocalización:** Integración con GeoLite2
- **Servicio de Threat Intelligence:** Consultas a BigQuery/Censys
- **Servicio de Correlación:** Combinación de datos de múltiples fuentes
- **Cache Service:** Almacenamiento temporal de resultados frecuentes

Frontend (Vue.js):

- **Interfaz de Consulta:** Formularios para ingreso de direcciones IP
- **Dashboard de Resultados:** Visualización de datos integrados
- **Componentes de Mapas:** Representación geográfica de resultados
- **Exportación de Datos:** Funcionalidades para guardar resultados

7.4.2. Flujo de Datos

1. **Entrada:** Usuario ingresa dirección IP a analizar
2. **Validación:** Sistema verifica formato y validez de la IP
3. **Consulta Paralela:**
   - Geolocalización via GeoLite2
   - Threat intelligence via Censys/BigQuery
4. **Correlación:** Integración de resultados de múltiples fuentes
5. **Presentación:** Visualización unificada en interfaz web

7.5. Herramientas y Tecnologías

7.5.1. Stack de Desarrollo

Backend - Quarkus Framework:

- Lenguaje: Java 17 LTS
- Framework: Quarkus 3.x
- Base de datos: PostgreSQL para cache y logs
- APIs: REST con Jackson para JSON

Frontend - Vue.js Ecosystem:

- Framework: Vue.js 3 con Composition API
- Routing: Vue Router
- State Management: Pinia
- UI Components: Vuetify o Bootstrap Vue
- Mapas: Leaflet.js para visualización geográfica

Infraestructura y Despliegue:

- Cloud Platform: Google Cloud Platform
- Contenedores: Docker para empaquetado
- CI/CD: GitHub Actions
- Monitoreo: Logs básicos y métricas de rendimiento

7.5.2. Herramientas de Desarrollo

IDEs y Editores:

- IntelliJ IDEA Community para Java/Quarkus
- Visual Studio Code para Vue.js
- Postman para testing de APIs

Control de Versiones:

- Git con GitHub para repositorio
- Branching strategy simplificado
- Issues tracking para gestión de tareas

7.6. Validación y Testing

7.6.1. Estrategias de Prueba

Testing Funcional:

- Unit tests para componentes individuales
- Integration tests para APIs
- End-to-end tests para flujos completos
- Manual testing de interfaz de usuario

Testing de Rendimiento:

- Pruebas de carga con múltiples consultas simultáneas
- Medición de tiempos de respuesta
- Evaluación de uso de memoria y CPU
- Testing de límites de quota de APIs externas

7.6.2. Validación de Resultados

Comparación con Herramientas Existentes:

- Verificación de geolocalización contra servicios conocidos
- Comparación de threat intelligence con fuentes públicas
- Evaluación de precisión y cobertura de datos

Testing de Usuario:

- Pruebas de usabilidad con estudiantes y profesores
- Evaluación de intuitividad de la interfaz
- Recolección de feedback para mejoras

7.7. Consideraciones Éticas y Técnicas

7.7.1. Uso Responsable de Datos

- Uso exclusivo de datos públicos y gratuitos
- Respeto a términos de servicio de proveedores
- No almacenamiento persistente de consultas de usuarios
- Implementación de rate limiting para evitar abuso

7.7.2. Limitaciones Técnicas Reconocidas

- Dependencia de disponibilidad de servicios externos
- Precisión limitada por calidad de fuentes de datos
- Capacidad de procesamiento limitada a recursos académicos
- Cobertura geográfica variable según la fuente

8. Cronograma de Trabajo

8.1. Duración y Alcance del Proyecto

Período Total: 8 semanas (del 10 de octubre al 1 de diciembre de 2025)
Equipo: 2 estudiantes universitarios
Modalidad: Desarrollo ágil con entregas incrementales

8.2. Metodología de Planificación

El cronograma se estructura utilizando sprints cortos de 1 semana para permitir entregas rápidas y ajustes continuos. Cada sprint incluye desarrollo, testing básico y documentación del progreso. Se consideran las limitaciones de tiempo académico y la curva de aprendizaje para las tecnologías seleccionadas.

8.3. Estructura de Desglose del Trabajo

8.3.1. Semana 1: Configuración e Investigación Inicial (10-17 Octubre)

Objetivos: Establecer ambiente de desarrollo y comprender fuentes de datos
Actividades principales:

- Configuración de entorno de desarrollo (Git, IDEs, cuentas cloud)
- Investigación de APIs de Censys y formato de datos BigQuery
- Descarga e instalación de GeoLite2
- Creación de prototipos de consulta básicos
- Definición de arquitectura inicial del sistema

8.3.2. Semana 2: Backend Base (17-24 Octubre)

Objetivos: Implementar estructura básica del backend con Quarkus
Actividades principales:

- Configuración inicial del proyecto Quarkus
- Implementación de API REST básica
- Integración con base de datos GeoLite2
- Desarrollo de servicios de geolocalización
- Testing unitario de componentes básicos

8.3.3. Semana 3: Integración BigQuery (24-31 Octubre)

Objetivos: Conectar con datos de Censys a través de BigQuery
Actividades principales:

- Configuración de acceso a Google Cloud Platform

- Implementación de cliente BigQuery
- Desarrollo de consultas SQL para datos de Censys
- Integración con servicios existentes de geolocalización
- Implementación de cache básico para optimizar consultas

8.3.4. Semana 4: Frontend Inicial (31 Octubre - 7 Noviembre)

Objetivos: Desarrollar interfaz básica con Vue.js
Actividades principales:

- Configuración del proyecto Vue.js
- Desarrollo de componentes básicos (formularios, tablas)
- Implementación de cliente HTTP para APIs
- Desarrollo de vistas principales (búsqueda, resultados)
- Integración con mapas usando Leaflet

8.3.5. Semana 5: Funcionalidades Avanzadas (7-14 Noviembre)

Objetivos: Mejorar funcionalidades y experiencia de usuario
Actividades principales:

- Implementación de búsqueda por lotes (múltiples IPs)
- Desarrollo de funcionalidad de exportación (JSON, CSV)
- Mejora de visualizaciones (gráficos, mapas interactivos)
- Implementación de historial de búsquedas (local storage)
- Optimización de rendimiento frontend

8.3.6. Semana 6: Testing y Optimización (14-21 Noviembre)

Objetivos: Validar funcionalidad y optimizar rendimiento
Actividades principales:

- Testing funcional completo de la aplicación
- Pruebas de rendimiento con múltiples consultas

- Testing de usabilidad con usuarios del entorno académico
- Optimización de consultas y cache
- Corrección de bugs identificados

8.3.7. Semana 7: Comparación y Validación (21-28 Noviembre)

Objetivos: Validar resultados contra herramientas existentes
Actividades principales:

- Comparación de resultados con herramientas públicas disponibles
- Análisis de precisión de geolocalización
- Evaluación de cobertura de threat intelligence
- Documentación de limitaciones encontradas
- Preparación de datos para presentación final

8.3.8. Semana 8: Documentación y Entrega (28 Noviembre - 1 Diciembre)

Objetivos: Completar documentación y preparar entrega final
Actividades principales:

- Completar documentación técnica del código
- Elaborar manual de usuario
- Preparar guía de instalación y configuración
- Creación de presentación final
- Deployment en ambiente de producción (si es posible)

8.4. Cronograma Visual

Actividad	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**
Configuración inicial	X							
Backend desarrollo		X	X					
Frontend desarrollo				X	X			
Testing y optimización						X		
Validación y comparación							X	
Documentación final								X

Cuadro 1: Cronograma Semanal del Proyecto

Riesgo	**Probabilidad**	**Impacto**	**Mitigación**
Dificultades con BigQuery	Media	Alto	Tutoriales, soporte académico
Problemas de rendimiento	Alta	Medio	Testing temprano, optimización
Complejidad de Vue.js	Media	Medio	Documentación, ejemplos simples
Limitaciones de datos gratuitos	Baja	Alto	Fuentes alternativas, scope adjustment
Retrasos por carga académica	Alta	Medio	Planificación flexible, buffer time

Cuadro 2: Matriz de Riesgos del Proyecto

8.5. Recursos y Herramientas

8.5.1. Equipo de Trabajo

- **Estudiante 1:** Enfoque en backend y integración de datos
- **Estudiante 2:** Enfoque en frontend y experiencia de usuario
- **Colaboración:** Testing, documentación y validación conjunta

8.5.2. Infraestructura Tecnológica

- **Desarrollo:** Computadoras personales con IDEs gratuitos
- **Cloud Services:** Google Cloud Platform (créditos académicos)
- **Repositorio:** GitHub (cuenta gratuita)
- **Comunicación:** Discord, WhatsApp, reuniones presenciales
- **Documentación:** Google Docs, LaTeX para documentos formales

8.5.3. Herramientas de Software

- **Backend:** Quarkus, Java 21, PostgreSQL
- **Frontend:** Vue.js 3, Vuetify, Leaflet.js
- **Testing:** JUnit, Jest, Postman
- **Deployment:** Docker, Google Cloud Run

Referencias

[1] MaxMind, Inc. (2025). *GeoLite2 Free Geolocation Data*. [Online]. Disponible en: https://www.maxmind.com/en/geolite2-developer-portal

[2] Censys, Inc. (2025). *Censys Universal Internet Data Set*. [Online]. Disponible en: https://censys.io/data

[3] Google Cloud. (2025). *BigQuery: Cloud Data Warehouse*. [Online]. Disponible en: https://cloud.google.com/bigquery

[4] Red Hat, Inc. (2025). *Quarkus: Supersonic Subatomic Java*. [Online]. Disponible en: https://quarkus.io/

[5] Evan You and The Vue Team. (2025). *Vue.js: The Progressive JavaScript Framework*. [Online]. Disponible en: https://vuejs.org/

[6] Durumeric, Z., Wustrow, E., and Halderman, J. A. (2015). "ZMap: Fast Internet-wide Scanning and Its Security Applications," in *Proceedings of the 22nd USENIX Security Symposium*, Washington, DC, USA.

[7] Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J. A. (2015). "A search engine backed by Internet-wide scanning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, Denver, CO, USA.

[8] Padmanabhan, V. N. and Subramanian, L. (2001). "An investigation of geographic mapping techniques for internet hosts," in *Proceedings of the 2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, San Diego, CA, USA.