ChatGPT

Modelo de Datos Neo4j para Evaluación de Riesgos de Infraestructura de Internet

1. Resumen Ejecutivo

Este modelo de datos en Neo4j está diseñado para mapear y evaluar riesgos en infraestructura de internet, enfocándose en dominios y sus cadenas de dependencias de terceros. El sistema permite identificar vulnerabilidades, calcular *scores* de riesgo y analizar el impacto de fallas en proveedores mediante análisis de grafos e integración temporal con Apache Iceberg.

2. Arquitectura del Modelo de Datos

2.1 Tipos de Nodos Principales

Organization (Organización)

```
(:Organization {
  id: String,                    // UUID único
  name: String,                  // Nombre de la organización
  type: String,                  // 'Enterprise', 'SMB', 'Startup'
  industry: String,              // Sector industrial
  country: String,               // País de registro
  created_at: DateTime,          // Fecha de creación en el sistema
  risk_score: Float,             // Score de riesgo agregado (0-100)
  risk_tier: String,             // 'Critical', 'High', 'Medium', 'Low'
  last_assessment: DateTime      // Última evaluación de riesgo
})
```

Domain (Dominio)

```
(:Domain {
  id: String,                    // UUID único
  fqdn: String,                  // Nombre de dominio completo
  tld: String,                   // Top-level domain
  registrar: String,             // Registrador del dominio
  registered_date: Date,         // Fecha de registro
  expiry_date: Date,             // Fecha de expiración
  status: String,                // 'Active', 'Suspended', 'Expired'
  dns_sec_enabled: Boolean,      // DNSSEC habilitado
  risk_score: Float,             // Score de riesgo del dominio
  business_criticality: String,  // 'Critical', 'High', 'Medium', 'Low'
  last_scan: DateTime            // Última evaluación
})
```

DNSServer (Servidor DNS)

```
(:DNSServer {
  id: String,                    // UUID único
  hostname: String,              // Nombre del servidor
  ip_address: String,            // Dirección IP
  ip_version: Integer,           // 4 o 6
  type: String,                  // 'Authoritative', 'Resolver', 'Root',
'TLD'
  provider: String,              // Proveedor del servicio
  geo_location: String,          // Ubicación geográfica
  anycast: Boolean,              // Usa anycast
  dnssec_validation: Boolean,    // Valida DNSSEC
  response_time_ms: Float,       // Tiempo de respuesta promedio
  availability_sla: Float,       // SLA de disponibilidad
  last_check: DateTime           // Última verificación
})
```

Certificate (Certificado)

```
(:Certificate {
  id: String,                    // UUID único
  serial_number: String,         // Número de serie
  subject_cn: String,            // Common Name
  subject_o: String,             // Organization
  issuer_cn: String,             // Issuer Common Name
  issuer_o: String,              // Issuer Organization
  type: String,                  // 'Root', 'Intermediate', 'EndEntity'
  signature_algorithm: String,   // Algoritmo de firma
  key_algorithm: String,         // Algoritmo de clave
  key_size: Integer,             // Tamaño de clave en bits
  valid_from: DateTime,          // Fecha de inicio de validez
  valid_to: DateTime,            // Fecha de expiración
  san_domains: [String],         // Subject Alternative Names
  is_wildcard: Boolean,          // Es certificado wildcard
  ct_logged: Boolean,            // En Certificate Transparency
  revocation_status: String,     // 'Valid', 'Revoked', 'Unknown'
  security_score: Float          // Score de seguridad (0-100)
})
```

Service (Servicio)

```
(:Service {
  id: String,                    // UUID único
  name: String,                  // Nombre del servicio
  type: String,                  // 'SaaS', 'PaaS', 'IaaS', 'CDN',
'Analytics'
  category: String,              // Categoría específica
```

```
  provider_name: String,          // Nombre del proveedor
  api_endpoint: String,           // Endpoint principal
  authentication_type: String,    // 'OAuth2', 'APIKey', 'SAML', etc.
  data_residency: [String],       // Países donde reside la data
  compliance_certs: [String],     // ['SOC2', 'ISO27001', 'GDPR']
  sla_availability: Float,        // SLA de disponibilidad
  has_data_export: Boolean,       // Permite exportación de datos
  vendor_lock_in_score: Integer, // Score de vendor lock-in (1-10)
  monthly_cost: Float,            // Costo mensual estimado
  risk_score: Float               // Score de riesgo del servicio
})
```

Provider (Proveedor)

```
(:Provider {
  id: String,                     // UUID único
  name: String,                   // Nombre del proveedor
  type: String,                   // 'DNS', 'Hosting', 'CDN', 'Certificate',
'Cloud'
  tier: Integer,                  // Nivel en la cadena (1=directo,
2=proveedor de proveedor)
  country: String,                // País de sede principal
  founded_year: Integer,          // Año de fundación
  employee_count: String,         // Rango de empleados
  revenue_range: String,          // Rango de ingresos
  market_share: Float,            // Cuota de mercado en su sector
  financial_health_score: Float, // Score de salud financiera
  security_rating: String,        // Rating de seguridad
  certifications: [String],       // Certificaciones de seguridad
  incident_history: Integer,      // Número de incidentes graves
  concentration_risk: Float,      // Riesgo de concentración (0-100)
  criticality_score: Float        // Score de criticidad
})
```

Technology (Tecnología)

```
(:Technology {
  id: String,                     // UUID único
  name: String,                   // Nombre de la tecnología
  version: String,                // Versión
  type: String,                   // 'Framework', 'Library', 'Platform',
'Tool'
  category: String,               // 'Frontend', 'Backend', 'Database', etc.
  vendor: String,                 // Vendedor/Mantenedor
  license: String,                // Tipo de licencia
  is_open_source: Boolean,        // Es open source
  latest_version: String,         // Última versión disponible
  eol_date: Date,                 // Fecha de end-of-life
  cve_count: Integer,             // Número de CVEs conocidos
```

```
  last_patch_date: Date,          // Fecha del último parche
  update_frequency: String,       // Frecuencia de actualizaciones
  security_score: Float           // Score de seguridad
})
```

2.2 Tipos de Relaciones

OWNS (Posee)

```
(:Organization)-[:OWNS {
  acquired_date: Date,            // Fecha de adquisición
  ownership_percentage: Float,    // Porcentaje de propiedad
  is_primary: Boolean            // Es dominio principal
}]->(:Domain)
```

DEPENDS_ON (Depende de)

```
(:Domain)-[:DEPENDS_ON {
  dependency_type: String,        // 'Critical', 'Important', 'Nice-to-have'
  service_level: String,          // Nivel de servicio contratado
  established_date: Date,         // Fecha de establecimiento
  monthly_requests: Integer,      // Requests mensuales
  data_volume_gb: Float,          // Volumen de datos
  failover_exists: Boolean        // Existe failover
}]->(:Service)
```

RESOLVES_TO (Resuelve a)

```
(:Domain)-[:RESOLVES_TO {
  record_type: String,            // 'A', 'AAAA', 'CNAME', 'MX'
  ttl: Integer,                   // Time to live
  priority: Integer,              // Prioridad (para MX)
  last_verified: DateTime         // Última verificación
}]->(:DNSServer)
```

SECURED_BY (Asegurado por)

```
(:Domain)-[:SECURED_BY {
  deployment_date: Date,          // Fecha de despliegue
  auto_renewal: Boolean,          // Renovación automática
  validation_type: String,        // 'DV', 'OV', 'EV'
  pin_sha256: String             // Pin del certificado
}]->(:Certificate)
```

PROVIDES (Provee)

```
(:Provider)-[:PROVIDES {
  service_type: String,          // Tipo de servicio
  contract_start: Date,          // Inicio del contrato
  contract_end: Date,            // Fin del contrato
  sla_terms: String,             // Términos del SLA
  support_tier: String           // Nivel de soporte
}]->(:Service)
```

ISSUED_BY (Emitido por)

```
(:Certificate)-[:ISSUED_BY {
  issue_date: DateTime,          // Fecha de emisión
  ca_type: String                // 'Root', 'Intermediate'
}]->(:Certificate)
```

USES_TECH (Usa tecnología)

```
(:Domain)-[:USES_TECH {
  detected_date: Date,           // Fecha de detección
  confidence: Float,             // Confianza en la detección
  is_vulnerable: Boolean,        // Versión vulnerable
  exposed_endpoints: Integer     // Endpoints expuestos
}]->(:Technology)
```

SUPPLIES_TO (Suministra a)

```
(:Provider)-[:SUPPLIES_TO {
  service_type: String,          // Tipo de servicio
  dependency_level: Integer,     // Nivel de dependencia
  can_replace: Boolean,          // Es reemplazable
  replacement_time_days: Integer // Tiempo de reemplazo
}]->(:Provider)
```

2.3 Propiedades de Metadatos para Evaluación de Riesgos

Metadatos de Seguridad TLS

```
// Propiedades adicionales para nodos Domain
{
  tls_version: String,           // 'TLS1.3', 'TLS1.2'
  cipher_suites: [String],       // Suites de cifrado soportadas
  perfect_forward_secrecy: Boolean,
  hsts_enabled: Boolean,         // HTTP Strict Transport Security
  hsts_max_age: Integer,         // Max age en segundos
```

```
  certificate_transparency: Boolean,
  ocsp_stapling: Boolean,
  ssl_labs_grade: String          // Grado de SSL Labs
}
```

Metadatos de Vulnerabilidades

```
(:Vulnerability {
  cve_id: String,                 // CVE-2023-XXXXX
  cvss_score: Float,              // Score CVSS
  cvss_vector: String,            // Vector CVSS
  cwe_id: String,                 // CWE-XXX
  description: String,
  published_date: Date,
  modified_date: Date,
  exploit_available: Boolean,
  exploit_maturity: String,       // 'Proof-of-concept', 'Functional', 'High'
  patch_available: Boolean,
  workaround_available: Boolean
})

// Relación
(:Technology)-[:HAS_VULNERABILITY {
  detected_date: Date,
  severity: String,               // 'Critical', 'High', 'Medium', 'Low'
  affected_versions: [String],
  fixed_versions: [String]
}]->(:Vulnerability)
```

3. Cálculo de Score de Riesgo

3.1 Fórmula Base

```
Risk Score = (Likelihood × Impact × Exposure) / Control Effectiveness
```

3.2 Componentes del Cálculo

Likelihood (Probabilidad)

- Historial de incidentes del proveedor
- Madurez de seguridad
- Complejidad de la infraestructura
- Exposición a internet

Impact (Impacto)

- Criticidad del negocio
- Número de usuarios afectados

- Pérdida financiera potencial
- Daño reputacional

Exposure (Exposición)

- Superficie de ataque
- Datos sensibles expuestos
- Integración con sistemas críticos
- Dependencias externas

Control Effectiveness (Efectividad de Controles)

- Configuración de seguridad
- Monitoreo y alertas
- Planes de contingencia
- Redundancia y failover

3.3 Propagación de Riesgo

```
// Consulta para calcular riesgo propagado
MATCH path = (org:Organization)-[:OWNS]->(d:Domain)-[:DEPENDS_ON*1..5]-
>(s:Service)
WITH org, d, s, length(path) as distance
WITH org, d, s, distance, s.risk_score * (0.8 ^ distance) as propagated_risk
WITH org, d, collect({service: s.name, risk: propagated_risk}) as
service_risks
RETURN org.name, d.fqdn,
       reduce(total = 0, r IN service_risks | total + r.risk) as
total_propagated_risk
```

4. Consultas de Análisis

4.1 Identificar Puntos Únicos de Falla

```
MATCH (d:Domain)-[:DEPENDS_ON]->(s:Service)<-[:PROVIDES]-(p:Provider)
WHERE NOT (d)-[:DEPENDS_ON]->(:Service)<-[:PROVIDES]-(p2:Provider)
WHERE p2 <> p AND s.dependency_type = 'Critical'
RETURN d.fqdn as domain, s.name as critical_service, p.name as
single_provider
```

4.2 Análisis de Concentración de Proveedores

```
MATCH (org:Organization)-[:OWNS]->(d:Domain)-[:DEPENDS_ON]->(s:Service)<-
[:PROVIDES]-(p:Provider)
WITH org, p, count(distinct s) as service_count, collect(distinct s.name) as
services
WHERE service_count > 3
```

```
RETURN org.name, p.name as provider, service_count, services,
       (toFloat(service_count) / 10) * 100 as concentration_risk_score
```

4.3 Cadenas de Dependencia Recursivas

```
MATCH path = (d:Domain)-[:DEPENDS_ON|PROVIDES|SUPPLIES_TO*1..10]->(final)
WHERE NOT (final)-[:DEPENDS_ON|PROVIDES|SUPPLIES_TO]->()
RETURN d.fqdn, [node in nodes(path) | labels(node)[0] + ':' + node.name] as
dependency_chain,
       length(path) as chain_depth
ORDER BY chain_depth DESC
```

4.4 Evaluación de Riesgo de Certificados

```
MATCH (d:Domain)-[:SECURED_BY]->(cert:Certificate)
WHERE cert.valid_to < datetime() + duration({days: 30})
   OR cert.key_size < 2048
   OR cert.signature_algorithm IN ['SHA1', 'MD5']
RETURN d.fqdn, cert.subject_cn,
       CASE
         WHEN cert.valid_to < datetime() THEN 'EXPIRED'
         WHEN cert.valid_to < datetime() + duration({days: 30}) THEN
'EXPIRING_SOON'
         WHEN cert.key_size < 2048 THEN 'WEAK_KEY'
         WHEN cert.signature_algorithm IN ['SHA1', 'MD5'] THEN
'WEAK_SIGNATURE'
       END as risk_type,
       cert.valid_to as expiry_date
```

5. Integración con Apache Iceberg

5.1 Arquitectura de Snapshots Temporales

```
-- Tabla Iceberg para snapshots diarios
CREATE TABLE infrastructure_snapshots (
  snapshot_date DATE,
  node_type STRING,
  node_id STRING,
  properties MAP<STRING, STRING>,
  risk_score DOUBLE,
  relationships ARRAY<STRUCT<
    rel_type: STRING,
    target_id: STRING,
    properties: MAP<STRING, STRING>
  >>
) USING iceberg
PARTITIONED BY (snapshot_date, node_type)
```

5.2 Consultas Temporales

```
-- Análisis de evolución de riesgo
WITH risk_evolution AS (
  SELECT
    node_id,
    snapshot_date,
    risk_score,
    LAG(risk_score) OVER (PARTITION BY node_id ORDER BY snapshot_date) as
prev_risk_score
  FROM infrastructure_snapshots
  WHERE node_type = 'Domain'
    AND snapshot_date BETWEEN current_date - 30 AND current_date
)
SELECT
  node_id,
  snapshot_date,
  risk_score,
  risk_score - prev_risk_score as risk_change,
  (risk_score - prev_risk_score) / prev_risk_score * 100 as risk_change_pct
FROM risk_evolution
WHERE abs(risk_score - prev_risk_score) > 5
ORDER BY abs(risk_change) DESC
```

5.3 Detección de Anomalías

```
-- Detectar cambios anómalos en dependencias
SELECT
  a.node_id,
  a.snapshot_date,
  a.dependency_count,
  b.avg_dependencies,
  b.stddev_dependencies,
  (a.dependency_count - b.avg_dependencies) / b.stddev_dependencies as
z_score
FROM (
  SELECT
    node_id,
    snapshot_date,
    SIZE(relationships) as dependency_count
  FROM infrastructure_snapshots
  WHERE node_type = 'Domain'
) a
JOIN (
  SELECT
    node_id,
    AVG(SIZE(relationships)) as avg_dependencies,
    STDDEV(SIZE(relationships)) as stddev_dependencies
  FROM infrastructure_snapshots
```

```
  WHERE node_type = 'Domain'
    AND snapshot_date BETWEEN current_date - 90 AND current_date
  GROUP BY node_id
) b ON a.node_id = b.node_id
WHERE abs((a.dependency_count - b.avg_dependencies) / b.stddev_dependencies)
> 3
```

6. Dashboard y Métricas Clave

6.1 KPIs de Riesgo

- **Risk Score Promedio por Organización**
- **Número de Puntos Únicos de Falla**
- **Concentración de Proveedores (Índice Herfindahl)**
- **Tiempo Medio hasta Expiración de Certificados**
- **Porcentaje de Servicios con Vulnerabilidades Críticas**

6.2 Alertas Automatizadas

- Certificados próximos a expirar (< 30 días)
- Nuevas vulnerabilidades críticas en tecnologías usadas
- Cambios significativos en *scores* de riesgo (>20%)
- Nuevas dependencias de alto riesgo detectadas
- Proveedores con incidentes de seguridad recientes

6.3 Reportes Ejecutivos

- Mapa de calor de riesgos por dominio
- Tendencias de riesgo en el tiempo
- Análisis comparativo con industria
- Proyecciones de riesgo basadas en tendencias
- Recomendaciones priorizadas de mitigación

7. Roadmap de Implementación

Fase 1: Infraestructura Base (Mes 1-2)

- Configuración de Neo4j cluster
- Modelo de datos *core* (Organization, Domain, DNS)
- Importación inicial de datos DNS
- APIs básicas de consulta

Fase 2: Servicios y Proveedores (Mes 3-4)

- Integración con APIs de proveedores
- Descubrimiento de servicios SaaS
- Mapeo de dependencias básicas
- Cálculo inicial de *scores* de riesgo

Fase 3: Seguridad y Certificados (Mes 5-6)

- Escaneo de certificados SSL/TLS
- Integración con bases de vulnerabilidades
- Análisis de configuraciones de seguridad
- Sistema de alertas

Fase 4: Análisis Temporal (Mes 7-8)

- Integración con Apache Iceberg
- *Snapshots* diarios automatizados
- Análisis de tendencias
- Detección de anomalías

Fase 5: Optimización y Escala (Mes 9-12)

- Optimización de consultas
- *Machine learning* para predicción de riesgos
- Integración con herramientas empresariales
- Expansión a nuevos tipos de servicios

8. Consideraciones de Seguridad y Privacidad

8.1 Control de Acceso

- Autenticación multi-factor para acceso al sistema
- RBAC (control de acceso basado en roles) para diferentes niveles
- Auditoría completa de accesos y cambios
- Encriptación de datos en reposo y tránsito

8.2 Privacidad de Datos

- Anonimización de datos sensibles
- Cumplimiento con GDPR/CCPA
- Políticas de retención de datos
- Derecho al olvido implementado

8.3 Seguridad Operacional

- Endurecimiento de la infraestructura Neo4j
- Monitoreo continuo de seguridad
- Respuesta a incidentes automatizada
- *Backups* encriptados y pruebas de recuperación

9. Conclusiones

Este modelo de datos proporciona una base sólida para:

- Visibilidad completa de dependencias de infraestructura.
- Evaluación cuantitativa de riesgos.
- Identificación proactiva de vulnerabilidades.
- Toma de decisiones basada en datos.

- Cumplimiento regulatorio mejorado.

La combinación de Neo4j para análisis en tiempo real y Apache Iceberg para análisis histórico permite tanto la respuesta inmediata a riesgos como el análisis estratégico de tendencias a largo plazo.
