

## Especificación de Requisitos Funcionales y No Funcionales

### 1. Requisitos Funcionales

#### 1.1. Búsqueda en Tiempo Real

1.1.1. **Consultar** la API de OSCE para **obtener** datos básicos sobre entidades públicas y proveedores.

1.1.2. **Mostrar** resultados de búsqueda en una lista, permitiendo al usuario **seleccionar** una entidad específica.

#### 1.2. Análisis de Relaciones Dinámica

1.2.1. **Extraer** contratos y conexiones relacionadas **con la entidad seleccionada** desde la API.

1.2.2. **Crear** nodos y relaciones en **Neo4j** para las entidades y contratos relacionados.

1.2.3. **Eliminar** los datos anteriores en **Neo4j** antes de iniciar un nuevo análisis.

#### 1.3. Detección de Influencias y Relaciones Sospechosas

1.3.1. **Asignar** un “peso” a las conexiones según la frecuencia de contratación entre las mismas entidades.

1.3.2. **Identificar** nodos o conexiones con características inusuales utilizando **NetworkX**.

1.3.3. **Generar** alertas o notificaciones visuales de posibles relaciones sospechosas.

#### 1.4. Visualización de Redes de Influencias

1.4.1. **Usar** D3.js para mostrar una red de relaciones interactiva.

1.4.2. **Permitir** que el usuario pueda visualizar datos resumidos de entidades al posicionar el mouse encima del nodo.

1.4.3. **Permitir** que el usuario observe más datos al seleccionar un nodo en específico.

1.4.4. **Permitir** al usuario seleccionar a otro nodo como nodo principal para que se reinicie el proceso de análisis tomando como referencia el nodo actual seleccionado.

#### 1.5. Actualización de Datos

1.5.1. **Eliminar** datos de consulta anterior en **Neo4j** cuando el usuario realiza una nueva búsqueda.

1.5.2. **Asegurar** que no se acumulen datos innecesarios en la base de datos entre consultas.

## 2. Requisitos No Funcionales

- 2.1. Rendimiento: La visualización de relaciones en Neo4j y D3.js debe completarse en menos de 6 segundos.
- 2.2. Usabilidad: El diseño debe ser intuitivo y con opciones de búsqueda que no impliquen conocimientos técnicos muy avanzados.
- 2.3. Escalabilidad Temporal: Manejo de consultas secuenciales, limpiando y refrescando los datos en Neo4j para cada consulta.
- 2.4. Confiabilidad: Manejo de errores a la API.
- 2.5. Seguridad: Evitar la sobrecarga de consultas a la API.