
This is the **published version** of the bachelor thesis:

Ramos Pomar, Gerard; Lladós Canet, Josep, tut. Desarrollo de una Aplicación Web para el Análisis de Redes de Citas Académicas mediante Técnicas de Grafos. 2025. (Enginyeria de Dades)

This version is available at <https://ddd.uab.cat/record/317358>

under the terms of the  license

This is the **published version** of the bachelor thesis:

Ramos Pomar, Gerard; Lladós Canet, Josep, tut. Desarrollo de una Aplicación Web para el Análisis de Redes de Citas Académicas mediante Técnicas de Grafos. 2025. (Enginyeria de Dades)

This version is available at <https://ddd.uab.cat/record/317358>

under the terms of the  license

This is the **published version** of the bachelor thesis:

Ramos Pomar, Gerard; Lladós Canet, Josep, tut. Desarrollo de una Aplicación Web para el Análisis de Redes de Citas Académicas mediante Técnicas de Grafos. 2025. (Enginyeria de Dades)

This version is available at <https://ddd.uab.cat/record/317358>

under the terms of the  license

Desenvolupament d'una Aplicació Web per a l'Anàlisi de Xarxes de Cites Acadèmiques amb Tècniques de Grafs

Gerard Ramos Pomar

Resum– Aquest treball desenvolupa una aplicació web basada en dades bibliogràfiques reals extretes del portal AMiner, amb l'objectiu de facilitar l'anàlisi i l'exploració de la producció científica mitjançant tècniques avançades de grafs. L'eina permet dur a terme cerques i obtenir visualitzacions que ajuden a descobrir informació rellevant sobre articles, autors i camps d'estudi, així com les relacions existents entre ells, amb la voluntat d'oferir valor afegit a la comunitat investigadora. El projecte cobreix tot el cicle de desenvolupament, des de l'adquisició i processament de les dades fins a la presentació en una interfície interactiva que facilita la consulta i interpretació dels resultats. Esdevé així un recurs valuós per a l'exploració i l'anàlisi del panorama científic.

Paraules clau– xarxa de cites, anàlisi de grafs, Neo4j, visualització de dades, aplicació web, dades bibliogràfiques, comunitats científiques

Abstract– This work develops a web application based on real bibliographic data extracted from the AMiner portal, aiming to facilitate the analysis and exploration of scientific production through advanced graph techniques. The tool enables searches and generates visualizations that help uncover relevant information about articles, authors, and fields of study, as well as the relationships among them, with the intention of providing added value to the research community. The project covers the entire development cycle, from data acquisition and processing to presentation in an interactive interface that facilitates the consultation and interpretation of results. Thus, it becomes a valuable resource for exploring and analyzing the scientific landscape.

Keywords– citation network, graph analysis, Neo4j, data visualization, web application, bibliographic data, scientific communities



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

UNA xarxa de cites és un graf dirigit que representa el domini de la literatura acadèmica, mostrant articles, autors i les connexions entre ells basades en citacions, coautories, versions i altres relacions. La comunitat científica genera un volum enorme d'articles cada dia. Per exemple, PubMed, un portal bibliogràfic públic de la National Library of Medicine, publica prop de tres mil articles diaris (Khalid et al., 2021) [1], i només és un dels múltiples portals existents. Aquesta gran quantitat de documents dificulta als investigadors la cerca d'articles i autors rellevants per a les seves necessitats. En aquest context, els grafs de cites esdevenen una eina molt valuosa, no només

per a la cerca d'informació, sinó també per a l'anàlisi de dades bibliogràfiques.

Actualment, existeixen diverses implementacions del graf de cites bibliogràfiques, com ara Google Scholar, the Microsoft Academic Graph, the Open Academic Graph, Semantic Scholar, AMiner, PubMed, Scopus and the Web of Science (Buneman et al., 2021) [2]. Cada plataforma presenta variacions en contingut, accés i esquema; però en tots ells l'estructura bàsica és un graf dirigit, on els nodes són publicacions i les arestes representen les cites d'una publicació a un altre (Price, 1965) [3].

Aquest projecte té com a objectiu la construcció d'un graf de cites, juntament amb les eines necessàries per dur a terme cerques i anàlisis sobre la xarxa, aportant valor a la comunitat acadèmica. Aquesta aplicació permetrà a investigadors realitzar una cerca més eficient en l'immens volum de documents disponibles i extreure conclusions a partir de les eines avançades d'anàlisi de grafs, facilitant així la identificació d'autors i articles influents, l'anàlisi de tendències en diferents àrees i la detecció de comunitats científiques,

• E-mail de contacte: Gerard.RamosP@autonoma.cat

• Treball tutoritzat per: Josep Lladós Canet (Departament de Ciències de la Computació)

• Curs 2024/25

entre altres aplicacions.

Més concretament el llistat d'objectius és el següent:

- Construir un graf de cites a partir de dades bibliogràfiques reals, que permeti realitzar cerques i anàlisis d'articles acadèmics, autors i les seves relacions.
- Dissenyar i implementar una base de dades orientada a grafs, estructurada per representar de manera eficient els diferents tipus de nodes i relacions presents a la xarxa de cites, i optimitzada per a consultes analítiques.
- Aplicar tècniques d'anàlisi de grafs amb l'objectiu d'expressar al màxim la informació continguda en l'estructura i les relacions de la xarxa, facilitant l'extracció de coneixement rellevant i aprofundint en la comprensió del conjunt de dades.
- Desenvolupar una interfície web interactiva i intuïtiva que permeti a la comunitat investigadora aprofitar el potencial del graf de cites mitjançant una presentació clara i estructurada de la informació, afavorint així una anàlisi comprensible i accessible del conjunt de dades.

2 METODOLOGIA

2.1 Metodologia de desenvolupament del projecte

Per assolir els objectius, el projecte s'ha organitzat en sis tasques, cadascuna composta per diverses tasques més específiques:

1. Recopilació de dades
2. Processament de dades
3. Decisions d'anàlisi i definició de funcionalitats
4. Disseny de la base de dades
5. Implementació del back-end
6. Implementació del front-end

Tal com es mostra al diagrama de Gantt de la figura 9 a l'annex, inicialment es va plantejar un desenvolupament seqüencial de les tasques. Aquest plantejament només defineix l'ordre lògic de les funcionalitats a implementar, però en cap cas s'ha seguit un mètode en cascada.

La metodologia utilitzada ha estat *incremental*, mantenint flexibilitat entre els diferents grups de tasques i realitzant sprints de dues setmanes, amb reunions a l'inici i al final per concretar objectius amb el tutor i avaluar els resultats obtinguts.

Al llarg del desenvolupament, s'han anat ajustant els objectius, les tasques i la planificació. És per això que, finalment, tasques com el desenvolupament d'un *crawler* per a l'obtenció de dades o l'ús d'una font externa no s'han implementat, ja que es va considerar que no s'alineaven amb la resta d'objectius del projecte i es va decidir prioritzar altres aspectes.

2.2 Metodologia tècnica

Al llarg d'aquest projecte s'han utilitzat diverses eines i tecnologies que han facilitat tant el processament de dades com la construcció del sistema i la seva visualització. Per al tractament i neteja de les dades inicials, s'ha emprat la llibreria *pandas* de Python, treballant dins l'entorn de Jupyter Notebook, fet que ha permès una anàlisi exploratòria àgil i estructurada.

Pel que fa a l'emmagatzematge de la informació i la gestió de les relacions entre entitats, s'ha fet servir el sistema de base de dades orientat a grafs Neo4j. Aquesta base de dades, a diferència dels sistemes relacionals tradicionals com MySQL o PostgreSQL, representa les dades mitjançant una estructura de nodes i arestes, fet que resulta especialment adequat per a la representació de xarxes complexes com les de cites acadèmiques. A més, Neo4j disposa del llenguatge de consultes Cypher, específicament dissenyat per formular operacions sobre grafs.

La capa del backend ha estat desenvolupada en Python, utilitzant la llibreria oficial *neo4j* per a la comunicació amb la base de dades i *Cytoscape.js* per a la visualització interactiva dels grafs dins la interfície. Finalment, la interfície gràfica de l'aplicació s'ha implementat amb el framework *Dash*, que combina components HTML i CSS per construir una aplicació web interactiva, personalitzable i modular, adequada per a la visualització i exploració de dades relacionades en temps real.

3 DESENVOLUPAMENT

3.1 Adquisició i processament de dades

Una base de dades sòlida és essencial per dur a terme aquest projecte. Per aquest motiu, s'ha realitzat una cerca exhaustiva amb l'objectiu de trobar un conjunt de dades que s'ajusti a les necessitats del projecte.

Entre els portals considerats, la majoria no permeten descarregar la totalitat de les dades, sinó que ofereixen una API per fer consultes específiques. Tot i això, aquestes APIs solen tenir limitacions que fan poc eficient la construcció d'una base de dades completa. Alguns exemples d'aquest primer grup són Semantic Scholar [4] i Microsoft Academic Graph (MAG) [5].

També hi ha portals que sí posen a disposició totes les seves dades, però molts d'ells són incomplets si s'utilitzen de forma aïllada. Per aquest motiu, s'ha decidit fer ús de l'última versió del conjunt de dades que ofereix el portal Aminer [6], el qual combina dades de diversos portals com DBLP [7], ACM [8] i el mateix MAG. Aquest conjunt de dades ha estat desenvolupat en el marc de l'article *ArnetMiner: Extraction and Mining of Academic Social Networks* (Tang et al., 2008) [9], on es descriu el procés d'extracció i anàlisi de xarxes socials acadèmiques. El conjunt seleccionat conté més de 5,2 milions d'articles i 36,6 milions de relacions de citació. Està en format JSON, ocupa aproximadament 20 GB i cada article pot contenir fins a 24 camps d'informació, detallats a l'annex (Secció A.1).

Amb l'ús de Jupyter Notebook i Python, s'ha realitzat una exploració inicial de les dades descarregades. Com que el fitxer és massa gran per carregar-lo completament a memòria, s'ha utilitzat *ijson* [10], que permet llegir ar-

xiu JSON de manera iterativa i incremental. Així, es poden processar els articles per grups, juntament amb les seves dades associades.

Un cop processades, les dades s'emmagatzemen en un `DataFrame` de `pandas`, fet que facilita la seva anàlisi i posterior preparació per a ser introduïdes a la base de dades. A més, també s'emmagatzemen en disc en format CSV per a futures consultes.

El conjunt de dades està centrat en els articles. Aquests disposen de codis únics per identificar-los, juntament amb altra informació, com ara una llista d'autors i els camps d'estudi corresponents. Per ambdós casos es disposa d'informació addicional. Tant autors com camps d'estudi creixen ràpidament: per una mostra de 1000 articles, s'obtenen prop de 2900 autors i 2600 camps d'estudi.

Per tal de preparar les dades per importar-les a la base de dades, s'han de reestructurar i netejar. Primer, s'organitzen en diversos `DataFrames`: un per cada objecte que es vol crear (articles, autors i camps d'estudi), i un per cada relació existent entre aquests objectes. Es poden observar més detalladament aquestes relacions i els camps escollits a la Figura 2.

Seguidament, es realitza la neteja de dades. S'han identificat alguns aspectes rellevants. Per exemple, els noms dels autors no es presenten de forma uniforme: un mateix autor pot aparèixer com `Ricardo Jiménez-Peris`, `Ricardo Jimenez Peris` o `Jimenez-Peris`, `R...` Afortunadament, cada autor disposa d'un identificador únic, fet que evita confusions a l'hora d'identificar-los. El repte, en aquest cas, és triar la representació més adient del nom per mostrar-la de manera clara. L'estratègia adoptada consisteix a seleccionar la variant més freqüent i, en cas d'empat, el nom més llarg o amb més paraules.

Un altre inconvenient detectat és la presència de cadenes amb caràcters com `\` dins d'algunes cadenes de text. Aquests caràcters s'utilitzen en certs sistemes per representar cometes dobles dins d'una cadena (per exemple, en formats on les cometes també delimiten el text). Tanmateix, aquest tipus de codificació no és acceptat per Neo4j, el motor de dades utilitzat, en camps de tipus `string`, ja que pot provocar errors d'interpretació durant la importació. Per aquest motiu, s'han substituït per cometes sense caràcters d'escapament.

Més enllà dels aspectes esmentats, el conjunt de dades presenta una qualitat notable i no s'hi han identificat altres incidències rellevants que puguin comprometre la integració a la base de dades.

3.2 Arquitectura del sistema

L'arquitectura de l'aplicació s'ha dissenyat a partir de tres capes: la base de dades, el servidor backend i la interfície gràfica. L'esquema general del sistema es mostra a la figura 1.

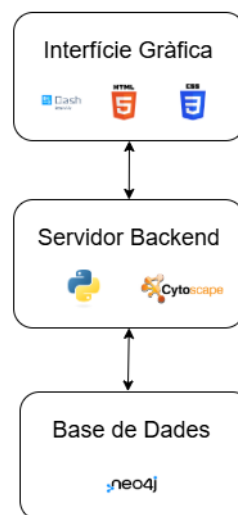


Fig. 1: Arquitectura general del sistema amb tres capes: interfície, backend i base de dades.

Tal com es pot observar, en l'extrem superior del flux hi ha la interfície gràfica, l'única capa amb la qual l'usuari final interacciona directament. Per transmetre dades entre la interfície i el servidor, es fa ús de les *callback functions*, funcions que són cridades automàticament sempre que canvia una propietat d'un component d'entrada, amb l'objectiu d'actualitzar alguna propietat en un altre component (la sortida) [11].

Aquestes funcions executen la tasca corresponent i, en cas que sigui necessari consultar la base de dades, es crea una connexió amb Neo4j mitjançant la llibreria `neo4j` [12]. La consulta s'executa, les dades s'extreuen i la sessió es tanca immediatament després. Un cop obtingudes, les dades es processen per adaptar-les al format requerit per Cytoscape i es retornen com a sortida del component corresponent. Finalment, es mostra l'estat actualitzat a la interfície perquè l'usuari pugui visualitzar i manipular.

Als apartats següents s'explica amb més detall el desenvolupament de cadascuna de les tres capes del sistema, més enllà de les connexions entre elles.

3.3 Disseny de base de dades

La base de dades és un graf implementat amb el motor Neo4j, en què els articles, els autors i els camps d'estudi es representen com a nodes, mentre que les arestes modelen les relacions entre ells. El diagrama de la base de dades es mostra a la figura 2.

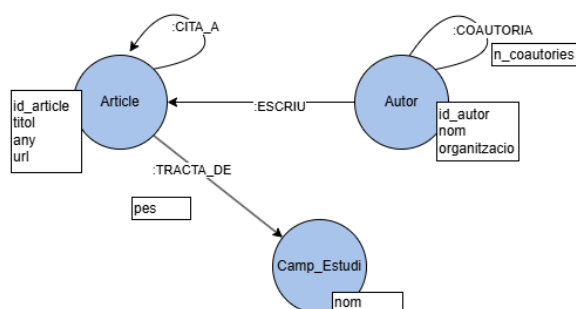


Fig. 2: Diagrama de la base de dades basada en grafs.

Els articles constitueixen l'eix central de la base de dades. Cada article pot haver estat escrit per entre 0 i n autors i pot tractar entre 0 i n camps d'estudi. Aquesta darrera relació incorpora un pes normalitzat entre 0 i 1 que indica la rellevància temàtica de l'article respecte a un camp concret. A més, els articles poden contenir relacions dirigides cap a altres articles, representant cites bibliogràfiques. Aquesta és l'única aresta del graf que pot existir en ambdues direccions entre un mateix parell de nodes.

Els nodes de tipus *Autor*, a banda de la relació amb els articles que han escrit, poden estar connectats entre si mitjançant una aresta de coautoria. Aquesta relació no és dirigida i incorpora com a pes el nombre de vegades que ambdós autors han col·laborat en la redacció d'un article.

Cal destacar dos aspectes rellevants. En primer lloc, Neo4j no admet arestes no dirigides com a tal: segons la seva documentació, *"totes les relacions tenen una direcció. Tot i això, pots tenir la noció de relacions no dirigides en el moment de la consulta"* [13]. Això implica que, tot i que les relacions s'han d'emmagatzemar amb una direcció determinada, les consultes poden ignorar aquesta direcció si es desitja. Tanmateix, aquest comportament configurable resulta especialment rellevant en algorismes com el Page-Rank, on la direcció de les arestes té un impacte directe en el resultat; interpretar una relació com a dirigida en un context on hauria de ser no dirigida pot conduir a resultats erronis.

En segon lloc, la relació de coautoria constitueix una redundància estructural, ja que dos autors són coautors si estan connectats al mateix article mitjançant una aresta de tipus *ESCRIU*. Tot i així, mantenir una aresta directa entre autors facilita significativament l'aplicació d'algorismes de processament de grafs, com ara els de camins mínims o detecció de comunitats, sense haver de reconstruir aquesta informació a cada consulta.

Finalment, els nodes de tipus *Camp_Estudi* només disposen de les arestes que els vinculen amb els articles; no contenen cap altra informació addicional dins del conjunt de dades.

Pel que fa al contingut dels nodes, els articles només conserven els camps que aporten valor significatiu a l'aplicació. Així mateix, quan està disponible, s'inclou l'enllaç URL per tal que es pugui accedir a informació addicional des de la interfície, si es considera necessari. En canvi, els nodes de tipus *Autor* i *Camp_Estudi* conserven tota la informació disponible dins del conjunt de dades original.

La importació de dades es realitza mitjançant el llenguatge Cypher, utilitzant els fitxers CSV específics per a cada tipus de node i aresta, generats a l'apartat d'Adquisició i processament de dades. El primer pas, consisteix en importar els nodes. Els tres tipus segueixen un esquema similar, al llistat 1 es mostra la consulta per a la importació dels articles.

```
LOAD CSV WITH HEADERS FROM 'file:///
articles.csv' AS row
WITH row
MERGE (a:Article {id: row.id})
SET a.titol = row.title, a.any = row.year,
a.url = row.url
RETURN COUNT(a);
```

Listing 1: Consulta en Cypher per a la importació d'articles.

En aquesta consulta, es carrega el fitxer corresponent, i mitjançant *MERGE*, es crea el node amb l'etiqueta *Article* i el paràmetre *id*. A continuació amb *SET*, se li assignen els atributs desitjats. Finalment, s'executa un *COUNT* per verificar que la consulta s'ha completat correctament i que s'ha importat el nombre de nodes esperats.

Una vegada importats tots els nodes, es poden importar les arestes que els uneixen. El llistat 2 mostra el codi necessari per crear les relacions entre autors i articles.

```
LOAD CSV WITH HEADERS FROM 'file:///
articles_autors.csv' AS row
WITH row
MATCH (ar:Article {id: row.id})
MATCH (au:Author {id: row.authorsId})
MERGE (au)-[rel:ESCRIU]->(ar)
RETURN count(rel);
```

Listing 2: Consulta en Cypher per a la importació de l'aresta *ESCRIU*.

Novament, l'esquema és similar per a tots els tipus d'arestes: s'importa l'arxiu CSV, es localitzen els dos nodes que es volen relacionar a partir del seu identificador mitjançant *MATCH*, i tot seguit es crea l'aresta amb *MERGE*. Si cal, se li assignen atributs, tot i que, en aquest cas la relació no en té cap. La consulta finalitza amb un *COUNT*, igual que en el cas de la importació dels nodes.

Finalment, es defineixen les *constraints*. Aquestes són regles que es poden establir dins de la base de dades per assegurar la qualitat i integritat. Un exemple són les *key constraints*, que garanteixen que un camp sigui únic i no nul. Aquest mecanisme és especialment útil per als identificadors dels nodes. Al llistat 3 es mostra un exemple per al cas dels articles.

```
CREATE CONSTRAINT id_article_unique IF NOT
EXISTS
FOR (a:Article) REQUIRE a.id IS NODE KEY;
```

Listing 3: Consulta en Cypher per establir una *key constraint* sobre l'identificador dels nodes *Article*.

A més de reforçar la coherència de les dades, aquesta restricció optimitza les cerques que es realitzen sobre aquest camp, millorant el rendiment global del sistema. Per implementar tot plegat, s'ha consultat la documentació oficial de Neo4j sobre *constraints* [14].

3.4 Descripció del backend

La capa del *backend* es pot dividir en tres components principals: les comandes a la base de dades, el processament de les dades i la lògica de control. Les dues primeres són clarament diferenciables, ja que corresponen a fitxers independents que poden ser reutilitzats per diferents parts del sistema. No obstant això, la lògica de control no es troba agrupada en un únic lloc, sinó que es distribueix en forma de *callbacks* independents associats a les diferents pàgines i components de la interfície.

Aquest component de control es reparteix entre les cinc seccions principals de l'aplicació: *Cercador*, *Camí mínim*, *Evolució del camp d'estudi*, *Top comunitats* i *Centralitat de nodes*. Totes aquestes seccions comparteixen una sèrie de funcions amb estructures similars, adaptades a les necessitats de cada cas concret.

Una de les funcions centrals és **load_and_prepare_data()**, l'única que no s'activa mitjançant un *callback*, sinó que és cridada per altres funcions de control. Aquesta funció consulta la base de dades mitjançant les funcions del component de comandes, rep les dades i les envia al component de processament per preparar-les en el format requerit per la llibreria *Cytoscape*.

La funció **display_graph()** s'activa en prémer el botó de cerca. Primer, comprova que les condicions siguin correctes: que el camp de text no estigui buit, que totes les opcions requerides estiguin seleccionades i que no hi hagi errors previsibles. Si tot és correcte, crida a la funció descrita al paràgraf anterior per obtenir les dades i mostra el graf. A més, es genera informació complementària com el nombre de nodes, arestes i la densitat del graf. Aquestes dades es mostren a la interfície per facilitar la comprensió i estudi del graf mostrat.

El càlcul de la densitat depèn de si el graf és dirigit o no. En el cas de grafs dirigits (*GD*) i no dirigits (*GND*), les fórmules són les següents:

$$Densitat_{GD} = \frac{m}{n \cdot (n - 1)}$$

$$Densitat_{GND} = \frac{2 \cdot m}{n \cdot (n - 1)}$$

on n és el nombre de nodes i m el nombre d'arestes.

La funció **update_layout()** permet canviar la disposició visual del graf. Les opcions disponibles són: *grid*, *random*, *circle*, *cose* i *concentric*. D'aquestes, les més destacables són:

- *cose*, que intenta representar la topologia del graf tenint en compte les relacions entre nodes. Tot i això, pot ser lenta amb grafs grans.
- *concentric*, ideal per mostrar un node i els seus veïns en forma circular. Funciona millor amb grafs densos, on altres opcions com *cose* poden superposar els nodes.

La funció **display_node_details()** s'activa en fer clic sobre un node i mostra la informació detallada d'aquest en un quadre lateral. Això permet a l'usuari explorar més profundament les característiques d'un node concret.

D'altra banda, la funció **update_graph_colors_and_legend()** actualitza els colors dels nodes i la llegenda cada vegada que es mostra un nou graf o es canvia el criteri de classificació. Agrupa els nodes segons el valor d'un camp escollit, els compta i els assigna un color diferenciat dins d'una escala predefinida. Tot plegat és útil per descobrir patrons que podrien passar desapercebuts. Per exemple, si a la pàgina de comunitats s'escull com a criteri de classificació l'organització a la qual pertany un autor, es podria observar si les comunitats creades estan influïdes per aquest factor.

Finalment, la funció **toggle_legend()** permet mostrar o amagar la llegenda fent clic sobre el seu títol. Això és especialment útil en casos on la llegenda pot ocupar massa espai i interferir en la interacció amb el panell de control.

A més, hi ha funcions específiques per a cada pàgina, dissenyades per dur a terme tasques particulars. Per exemple, les pàgines del *cercador* i de *l'evolució del camp d'estudi*

disposen d'un conjunt de funcions que gestionen el funcionament del cercador de text. Aquestes permeten cercar contingut, mostrar les opcions trobades, seleccionar una d'elles i fer que aquesta aparegui automàticament al quadre de text.

Les pàgines de *comunitats* i d'*evolució del camp d'estudi* incorporen, a més, una funció encarregada d'actualitzar el graf ja mostrat. En el primer cas, perquè es disposa d'un sistema de navegació a través d'un llistat de diferents comunitats; i en el segon, perquè es fa ús d'un *slider* temporal que determina quins articles s'han de visualitzar en el graf segons el període seleccionat.

El component de comandes a la base de dades disposa de vuit funcions. Totes elles, excepte una funció auxiliar, reben com a paràmetre l'objecte *driver*, que és l'encarregat de gestionar la connexió amb la base de dades. A partir d'aquest objecte s'executa la consulta, s'obtenen els resultats, i posteriorment es tracten al fitxer de processament de dades. Aquestes funcions segueixen, en general, l'estructura mostrada al llistat 3.

```
def example_query_function(driver,
    parameter1):
    cypher_query = """
        MATCH ...
        WHERE id = $parameter1
        ...
    """
    records, _, _ = driver.execute_query(
        cypher_query, parameter1=parameter1)
    return records
```

Listing 4: Estructura general de funció amb consulta Cypher

La funció rep els paràmetres necessaris, declara una variable de tipus *string* amb la consulta Cypher, executa la consulta utilitzant l'objecte *driver*, i hi incorpora els paràmetres requerits (si n'hi ha). A la consulta, aquests paràmetres s'escriuen amb el símbol $\$$ al davant.

La funció **text_search()**, té com a objectiu permetre cerques de text, és a dir, trobar els nodes d'un tipus determinat on el nom contingui la cadena proporcionada. Per facilitar la comparació entre cadenes, es transformen tots els caràcters a minúscules i s'eliminen els espais en blanc. A més, es limita la cerca als primers deu nodes trobats per tal de simplificar el procés de selecció per a l'usuari.

La funció **obtain_node_and_neighs()** s'encarrega de trobar els veïns d'un node concret que siguin d'un tipus determinat. Això permet, per exemple, obtenir ràpidament els autors o els camps d'estudi associats a un article que resulti d'interès per a l'usuari.

La funció **shortest_path()** rep un identificador inicial i un de destí, així com un paràmetre que indica si es tracta d'una comanda referida a autors o articles. Aquesta funció cerca el camí mínim entre els dos nodes. En cas que hi hagi més d'un camí amb el mateix nombre de salts, se'n selecciona un de manera no determinista; per tant, executar la funció dues vegades pot retornar resultats diferents. Aquesta funcionalitat permet conèixer a quants salts es troben dos autors o articles, i quins nodes es troben entre ells, la qual cosa facilita l'anàlisi de col·laboracions o influències.

La funció **obtain_min_and_max_year_evolution()** és l'única funció auxiliar del fitxer. No realitza cap consulta

a la base de dades, sinó que determina l'any mínim i màxim d'una llista de nodes passada com a paràmetre. Els resultats s'utilitzen per establir els valors extrems del *slider* de la pàgina d'estudi de l'evolució temporal d'un camp d'estudi.

Les funcions `check_is_graph_loaded()` i `load_graph_to_memory()` treballen conjuntament per gestionar la càrrega de subgrafs a memòria, amb l'objectiu d'executar-hi algorismes mitjançant la llibreria *Graph Data Science* (GDS) de Neo4j. Aquesta llibreria proporciona diversos algorismes ja implementats, que són utilitzats per a la detecció de comunitats i el càlcul de mesures de centralitat. La primera funció comprova si el subgraf ja ha estat carregat prèviament, mentre que la segona el carrega a memòria si cal.

La funció `execute_community_algorithm()` executa l'algorisme de comunitats seleccionat i retorna les n comunitats amb més nodes, essent n un valor passat com a paràmetre. Les opcions disponibles són:

- **Weakly Connected Components:** retorna tots els subconjunts de nodes connectats entre si però aïllats dels altres.
- **Label Propagation:** assigna inicialment una etiqueta única a cada node i, en cada iteració, cada node adopta l'etiqueta més comuna entre els seus veïns. El procés continua fins que es converteix o s'arriba a un límit d'iteracions.
- **Louvain Modularity:** utilitza la modularitat per optimitzar la partició del graf. Els nodes s'agrupen inicialment en comunitats separades, i es van traslladant entre comunitats si això incrementa la modularitat global.

L'execució d'aquests algorismes facilita una millor comprensió del graf en qüestió: permet identificar si el graf està altament connectat o presenta diverses illes separades, així com descobrir comunitats científiques d'autors influents o grups d'articles altament correlacionats.

La funció `execute_centrality_algorithm()` executa un dels algorismes de centralitat disponibles i retorna els nodes amb els seus valors corresponents. Les opcions són:

- **PageRank:** mesura la influència d'un node dins del graf segons el nombre i la qualitat dels enllaços que rep.
- **Betweenness Centrality:** indica la importància d'un node en termes de connectivitat, mesurant quantes vegades un node es troba en els camins més curts entre altres parells de nodes.

Aquests algorismes s'executen tant sobre el graf de coautories com sobre el graf de cites. És important destacar que, en el cas del graf d'autors, cal passar un paràmetre indicant que no s'ha de tenir en compte la direcció de les arestes, ja que conceptualment són no dirigides, tot i que Neo4j obliga a emmagatzemar-les amb una direcció. Aquesta consideració es desenvolupa més detalladament a l'apartat 3.3.

Els algorismes de centralitat proporcionen informació molt valuosa. En el cas del *PageRank*, permeten identificar articles o autors que són especialment rellevants dins la

xarxa, la qual cosa pot ajudar a descobrir contingut potencialment més influent i de major interès per a l'usuari.

El component encarregat del processament de dades conté diverses funcions principals.

Les dues funcions més rellevants són les encarregades de convertir les dades obtingudes des de Neo4j al format requerit per Cytoscape: `process_nodes()` i `process_edges()`. Aquest format consisteix en una llista de diccionaris, on cada diccionari representa un node o una aresta del graf, i conté dos elements: la classe de l'element i un altre diccionari amb totes les propietats associades a aquest.

Pel que fa al processament dels nodes, la funció `process_nodes()` recorre la llista de nodes rebuda com a paràmetre, i en funció de l'etiqueta del node, es gestiona el processament segons les característiques específiques de cada tipus de node, tenint en compte que cada categoria disposa d'atributs diferents, tal com es pot observar a la figura 2. Per a tots els nodes, es genera un atribut anomenat *label*, que correspon al text que es mostrarà sobre el node en la visualització, i que es trunca en cas que la seva llargada excedeixi els catorze caràcters, mitjançant la funció `truncate_label()`. Això es fa per evitar sobrecarregar el graf amb informació i així obtenir una visualització més clara i entenedora.

A més, es defineix la grandària de cada node a través d'un atribut específic. També es gestionen dos casos especials: en primer lloc, si es proporciona la llista opcional de centralitats com a paràmetre, aquesta informació s'afegeix a les propietats de cada node, ja que es mostrarà posteriorment en els detalls dels nodes seleccionats. En segon lloc, si el paràmetre `isShortestPath` s'estableix com a cert, s'inclou un atribut que indica si el node correspon a un punt inicial, final o intermedi dins d'un camí mínim. Aquesta diferenciació permet personalitzar la visualització dels nodes als extrems del camí mitjançant un *label* amb un color distintiu, facilitant-ne la identificació visual dins del graf.

Pel que fa a les arestes, el processament dut a terme per la funció `process_edges()` és més senzill i no requereix tanta personalització com en el cas dels nodes. Per a cada aresta, s'emmagatzemen els identificadors dels nodes d'inici i de destí de la relació, així com el tipus de relació. A més, si l'aresta disposa d'un atribut de pes, aquest també s'inclou entre les propietats associades, ja que es mostrarà al graf.

3.5 Desenvolupament de la interfície

La interfície gràfica s'ha desenvolupat mitjançant la combinació d'elements HTML i estils CSS, tot integrat dins del framework *Dash*. L'organització visual es basa en una pàgina principal amb un menú i cinc subpàgines, cadascuna corresponent a una funcionalitat específica. La vista del menú principal és mostra la figura 3.

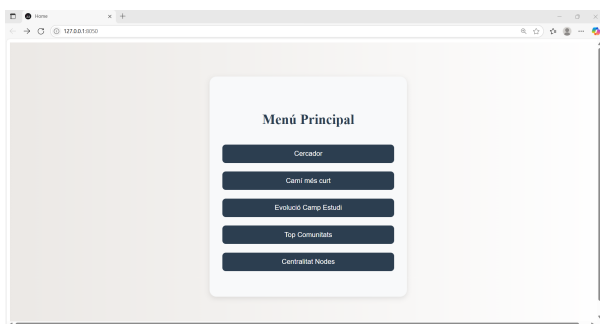


Fig. 3: Menú principal de l'aplicació amb accés a les funcionalitats.

Quan es prem qualsevol de les opcions del menú, es carrega la vista corresponent. Per exemple, en accedir a la secció d'*Evolució del camp d'estudi* i realitzar una consulta, es mostra una interfície com la que s'observa a la figura 4.

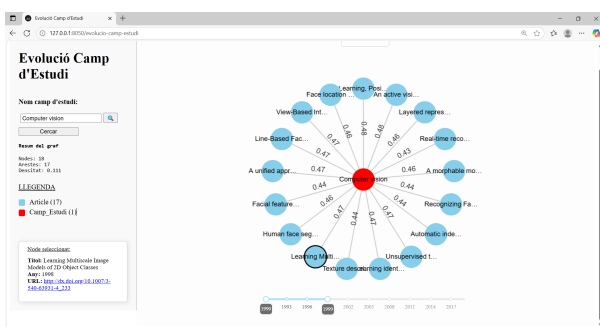


Fig. 4: Exemple d'execució per a la pàgina d'evolució de camps d'estudi.

Totes cinc pàgines comparteixen una estructura similar. A la dreta, ocupant la major part de la pantalla, es mostra el graf. A l'esquerra hi ha el panell de control, que inclou, a la cantonada superior, un botó per tornar al menú principal i, a continuació, el títol de la secció.

A sota del títol, s'hi troben els camps d'entrada (*inputs*) i el botó de cerca. Més avall s'hi mostra la informació general del graf, la llegenda de colors i els detalls del node seleccionat.

En el cas concret de la funcionalitat d'evolució d'un camp d'estudi, també s'hi incorpora un *slider* que permet determinar, de manera interactiva, quin període de publicació es vol visualitzar en el graf.

L'estil dels components del graf també ha estat completament personalitzat. Mitjançant diccionaris d'estils CSS, s'ha definit l'aspecte visual dels nodes: amplada, alçada, text visible i condicions per mostrar-lo. També s'ha creat un estil específic per al node seleccionat, ressaltant-ne la vora amb color negre per destacar-lo.

Les arestes també s'han personalitzat, incloent-ne el gruix, direcció, color i estil. A més, s'han definit estils específics per a les arestes dels tipus COAUTORIA i TRACTA_DE, de manera que mostrin el pes de l'enllaç com a text sobre l'aresta.

4 PROVES I RESULTATS

Amb l'objectiu de validar el correcte funcionament de l'aplicació i assegurar-ne la robustesa, s'han realitzat proves

funcionals, així com proves de rendiment enfocades a determinar els límits de càrrega pel que fa a la quantitat de dades gestionables pel sistema.

4.1 Proves funcionals

Pel que fa a la funcionalitat, durant el desenvolupament s'han aplicat proves unitàries a les funcions a mesura que es desenvolupaven, amb l'objectiu de garantir que produïssin els resultats esperats. Paral·lelament, s'han realitzat proves d'integració per verificar la correcta interacció entre els diferents components del sistema.

A mesura que es completaven les diferents parts de l'aplicació, també s'han dut a terme proves manuals mitjançant la interacció directa amb la interfície. Aquestes proves han tingut com a objectiu detectar errors i comportaments inesperats. Alguns exemples inclouen la introducció del mateix identificador com a node d'origen i de destí en el càlcul del camí mínim, o la introducció de caràcters no numèrics en camps destinats a valors numèrics. Aquestes i altres proves han contribuït a incrementar la robustesa de l'aplicació davant situacions excepcionals i errors d'entrada.

4.2 Proves de rendiment

Per determinar els límits de capacitat de l'aplicació, s'han carregat diferents volums de dades. Amb la incorporació de 750.000 articles, juntament amb els seus respectius autors i camps d'estudi, s'ha observat que l'aplicació era capaç d'executar els algorismes sense grans inconvenients. Tot i que en alguns casos el temps de resposta era elevat, el coll d'ampolla no es trobava en l'execució de les consultes o algorismes, sinó en la visualització de les dades.

En seccions que impliquen una representació gràfica amb un nombre elevat de nodes, com ara l'apartat de centralitat de nodes, no s'ha pogut aplicar cap mètode de distribució que permeti una visualització clara i llegible de tot el graf. A més, l'execució del mètode de distribució COSE (descriu a l'apartat 3.4) pot provocar que la pàgina deixi de respondre a causa del consum elevat de recursos.

En realitzar proves amb conjunts de 250.000 i 100.000 articles, l'aplicació continua sense ser capaç de mostrar la totalitat dels nodes de manera funcional. Finalment, es du a terme una prova amb un conjunt reduït de 1.000 articles. En aquest cas, l'aplicació és completament funcional i estable.

Tanmateix, carregar el graf de coautories amb aproximadament 3.000 autors suposa un cost considerable. En mesurar els temps d'execució, s'observa que el procés de càrrega de les dades al graf dura aproximadament tres segons, però la visualització i estabilització dels nodes (fins assolir un punt de convergència) pot requerir fins a un minut, dependent de l'execució.

4.3 Resultats generals

Tot i les limitacions descrites, els resultats obtinguts són força òptims, ja que la representació gràfica d'un graf d'aquestes dimensions no és en cap cas una tasca trivial. L'aplicació resultant pot esdevenir una eina útil per a l'estudi de xarxes de cites acadèmiques.

Una de les funcionalitats destacades és la pàgina de **camí mínim**, que permet estudiar la relació entre dos autors o ar-

tibles dins la xarxa, sempre que existeixi una connexió entre ells. Aquesta funcionalitat s'inspira en conceptes com el número d'Erdős, una mesura de proximitat acadèmica que indica el nombre mínim de passos de coautoría que separen un autor del matemàtic Paul Erdős. Per exemple, si un investigador ha escrit un article conjuntament amb Erdős, té un número d'Erdős igual a 1; si ha col·laborat amb algú que alhora ha col·laborat amb Erdős, el seu número és 2, i així successivament.

Tot i que aquest projecte no calcula aquest número explícitament, la capacitat de determinar el camí més curt entre dues entitats dins el graf permet identificar relacions de col·laboració o influència entre investigadors, i explorar com s'articula el flux de coneixement dins la comunitat científica. A la figura 5 es mostra un exemple pràctic d'aquesta funcionalitat.

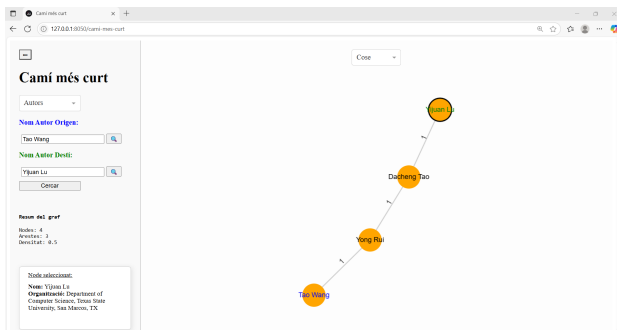


Fig. 5: Exemple d'execució per a la pagina de camins mínims.

La secció **d'evolució del camp d'estudi** possibilita identificar articles rellevants dins d'un àmbit concret i observar-ne l'evolució al llarg dels anys. Mitjançant el control de l'eix temporal (slider), és possible recuperar, per exemple, articles dels anys noranta sobre intel·ligència artificial. Aquesta funcionalitat pot resultar especialment valuosa, ja que facilita la descoberta de treballs que podrien ser difícils de localitzar per altres vies. La visualització d'aquesta secció es mostra a la figura 4.

La pàgina **d'estudi de comunitats** ofereix un gran potencial analític. Permet identificar conjunts d'articles que es citen repetidament entre ells, cosa que indica una possible similitud temàtica i que aquests articles podrien ser recomanats conjuntament en un sistema de recomanació, per exemple. També facilita la detecció de grups d'autors que col·laboren habitualment, així com la identificació de comunitats científiques fins ara desconegudes.

A més, aplicant criteris de classificació basats en atributs com l'afiliació institucional dels autors, es pot analitzar de manera visual i ràpida la possible correlació entre les comunitats detectades i les organitzacions o universitats a les quals pertanyen. Aquesta funcionalitat també ajuda a detectar col·laboracions interinstitucionals, com es pot observar a la visualització de la figura 6.

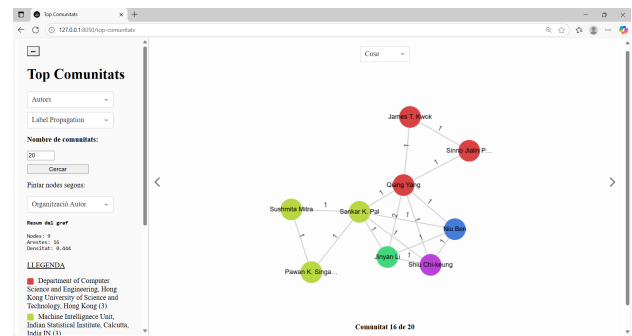


Fig. 6: Exemple d'execució per a la pàgina de comunitats.

En la visualització es pot observar un grup d'autors identificats en vermell, pertanyents al Departament de Ciències de la Computació i Enginyeria de la Universitat de Ciència i Tecnologia de Hong Kong. També es detecta una altra comunitat diferenciada en verd clar corresponent a autors de la *Machine Intelligence Unit* de l'Institut Indi d'Estadística a Calcuta.

Resulta especialment interessant que un membre de cadascuna d'aquestes comunitats col·labora amb tres autors més, cadascun d'una organització diferent, creant així ponts entre comunitats i institucions. Aquest tipus d'anàlisi revela patrons de col·laboració interdisciplinària o internacional que podrien passar desapercebuts amb altres eines.

Carregant un volum més gran de dades, les possibilitats d'aquesta secció de l'aplicació poden esdevenir especialment prometedores per a l'estudi i visualització de xarxes científiques.

La pàgina de **centralitat de nodes** permet visualitzar el graf complet, oferint una perspectiva global especialment útil per entendre l'estructura general de la xarxa. Aquesta visió de conjunt facilita la identificació immediata dels nodes més rellevants, gràcies al fet que la seva mida es troba directament relacionada amb la seva centralitat.

A més, el graf és interactiu: l'usuari pot fer zoom, desplaçar-se i reorganitzar manualment els nodes per tal d'analitzar amb més detall una determinada regió del graf.

A la figura 7, es mostra una visualització resultant de l'aplicació de l'algoritme *PageRank* sobre el graf de cites.

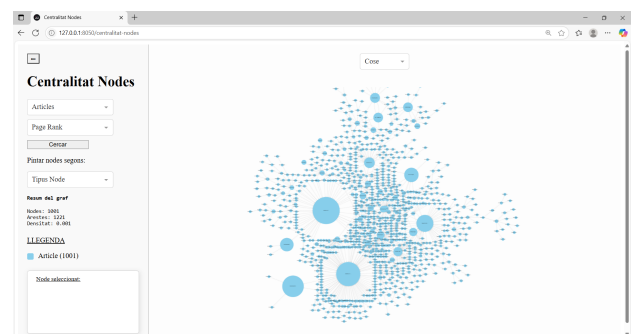


Fig. 7: Resultat de l'algoritme PageRank sobre el graf de cites.

Aquesta vista permet identificar de forma ràpida els nodes més influents dins la xarxa i estudiar les seves connexions en detall. A més, es pot accedir directament als documents originals mitjançant els enllaços associats a cada node, fet que facilita una anàlisi més profunda dels continguts.

Finalment, la secció del **Cercador** actua com a element integrador de tota l'aplicació, ja que permet aprofundir en qualsevol node que hagi resultat d'interès en altres seccions. Aquesta funcionalitat possibilita analitzar les relacions directes d'un node amb la resta de la xarxa de forma eficient i centralitzada.

En fer clic sobre el node més gran i central del graf a la figura 7, l'usuari descobreix que es tracta de l'article titulat "Induction of Decision Trees". A partir d'aquest punt, pot copiar el títol i dirigir-se a la pàgina del cercador per continuar-ne l'exploració.

Un cop introduït el títol i seleccionat que es mostrin els veïns de tipus autor, el sistema genera la visualització mostrada a la figura 8.

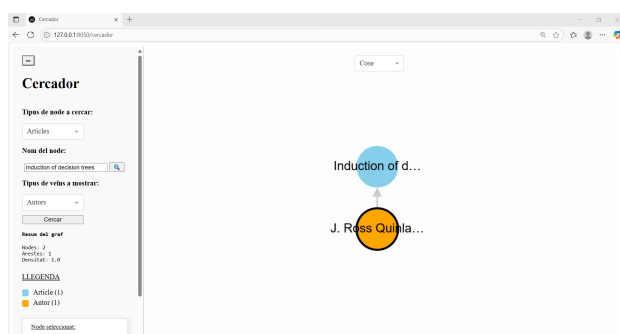


Fig. 8: Exemple d'execució de la pàgina del Cercador.

En aquest cas, s'observa que l'article té un únic autor. A partir d'aquí, es poden fer cerques addicionals per conèixer si aquest autor ha publicat altres treballs, o amb quins altres investigadors ha col·laborat al llarg del temps. Aquest procés exemplifica el potencial de la funcionalitat de cerca com a porta d'entrada per a múltiples línies d'exploració dins la xarxa de dades.

5 CONCLUSIONS

Per concloure, cal destacar que el treball realitzat ha assolit l'objectiu general del projecte, ja que s'ha construït un graf de cites amb les eines necessàries per a realitzar cerques i anàlisis sobre la xarxa.

Pel que fa als objectius més específics:

El primer objectiu consistia a construir un graf de cites a partir de dades bibliogràfiques reals que permetés realitzar cerques i anàlisis sobre els seus components. En primer lloc, s'ha obtingut un conjunt de dades autèntic amb més de cinc milions d'articles, cadascun dels quals conté diversos camps informatius. En segon lloc, l'eina resultant disposa d'una àmplia varietat de funcionalitats per analitzar la xarxa de manera exhaustiva i, a més, incorpora un cercador que permet visualitzar articles, autors o camps d'estudi específics i els seus veïns immediats. Per tant, es pot afirmar que aquest objectiu ha estat plenament assolit.

El segon objectiu consistia en implementar una base de dades orientada a grafs, optimitzada per a l'execució de consultes analítiques. El disseny adoptat és simple, evitant una excessiva varietat de tipus de relacions i nodes, i s'ha optimitzat, per exemple, creant una aresta de coautoría per representar la relació entre autors que, encara que implícita al graf, resultaria menys eficient de cercar directament. A

més, s'han utilitzat *constraints* per convertir en índexs els camps dels nodes sobre els quals es realitzen cerques de text. Finalment, els nodes només contenen la informació essencial i utilitzada, evitant dades innecessàries. Per tant, aquest objectiu també s'ha complert amb èxit.

El tercer objectiu es referia a extreure el màxim valor de les dades disponibles mitjançant tècniques avançades d'anàlisi de grafs. Les dades escollides s'han explotat de manera significativa, trobant nombrosos casos d'ús útils per a la recerca acadèmica. Hauria estat interessant aprofundir en camps com paraules clau o els resums dels articles, ja que aquests podrien haver donat peu a la detecció de similituds entre articles, obrir la porta a sistemes de recomanació o a la creació de nous tipus d'arestes en el graf. En conclusió, aquest objectiu ha estat assolit, però s'hi podria treure encara més potencial.

El quart objectiu consistia en dissenyar una interfície interactiva i intuïtiva que presentés la informació de manera clara i estructurada, facilitant així a la comunitat científica l'aprofitament òptim del conjunt de dades. La versió final de l'aplicació ofereix un alt grau d'interactivitat, permetent a l'usuari establir els paràmetres d'entrada per a les cerques, moure els nodes del graf, fer zoom i veure els detalls dels nodes seleccionats, entre altres funcionalitats. Tot i això, la intuïció i facilitat d'ús del sistema no constitueixen el seu punt més fort, ja que aquesta part ha rebut una dedicació limitada i no s'han realitzat proves d'usabilitat amb usuaris variats que permetessin identificar i corregir possibles dificultats d'interacció. Pel que fa a la presentació visual, l'organització de les vistes és neta i funcional, amb una disposició que prioritza l'espai destinat al graf i una columna lateral per als paràmetres de configuració. En resum, la interfície disposa d'un nivell elevat d'interactivitat, però la facilitat d'ús podria ser notablement millorada abans d'una eventual publicació o posada en producció.

Tot i que el projecte ha assolit els objectius plantejats, cal destacar que hi ha aspectes que podrien ser millorats o ampliat per potenciar-ne el valor i la qualitat. En aquest sentit, un dels factors més importants, que modificaria si tornés a realitzar el projecte, seria trobar una solució eficient per mostrar grans quantitats de nodes, o bé dissenyar una forma alternativa per visualitzar les centralitats sense necessitat de representar tot el graf, ja que treballar amb un volum més elevat de dades és fonamental.

També seria interessant considerar la incorporació de les paraules clau i els resums dels articles per implementar un sistema de recomanació basat en la similitud entre continguts. A més, el cercador de text podria millorar-se per oferir resultats més rellevants, mostrant cerques properes al text introduït, fet que augmentaria l'eficiència de la cerca.

Finalment, un altre aspecte clau seria augmentar la interconnexió entre les funcionalitats de l'aplicació. Per exemple, permetre que fent clic en un node des de qualsevol pàgina es pogués executar de manera directa una acció concreta, com ara mostrar els veïns o calcular el camí mínim. Actualment, aquestes funcionalitats són accessibles, però no tan ràpides ni còmodes com ho serien si estiguessin més integrades i accessibles des de qualsevol context.

Per acabar, aquest projecte ha demostrat la viabilitat i utilitat de l'anàlisi de xarxes de cites acadèmiques mitjançant tècniques de graf, tot i que hi ha marge per a futures millores que en potenciarien encara més l'impacte i l'eficiència.

AGRAÏMENTS

Vull agrair al meu tutor, Josep Lladós Canet, pel suport i l'acompanyament durant aquest treball. La seva ajuda ha estat clau per enfocar i desenvolupar el projecte.

REFERÈNCIES

- [1] S. Khalid, S. Wu, A. Wahid, A. Alam, and I. Ullah. An effective scholarly search by combining inverted indices and structured search with citation networks analysis. *IEEE Access*, 9:120210, 2021. Disponible en línia.
- [2] P. Buneman, D. Dosso, M. Lissandrini, and G. Silvello. Data citation and the citation graph. *Quantitative Science Studies*, pages 3–4, 2021. Advance Publication.
- [3] D. J. D. S. Price. Networks of scientific papers. *Science*, pages 510–515, 1965.
- [4] Semantic Scholar. Semantic scholar. <https://www.semanticscholar.org/>. [Accedit: 12-mar-2025].
- [5] Microsoft Research. Microsoft academic graph. <https://www.microsoft.com/en-us/research/project/microsoft-academic-graph/>. [Accedit: 12-mar-2025].
- [6] AMiner. Aminer citation network. <https://www.aminer.cn/citation>. [Accedit: 16-mar-2025].
- [7] M. Ley. Dblp: Digital bibliography & library project. <https://dblp.uni-trier.de/>. [Accedit: 14-mar-2025].
- [8] Association for Computing Machinery (ACM). Acm digital library. <https://dl.acm.org/>. [Accedit: 14-mar-2025].
- [9] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD'08)*, pages 990–998, 2008.
- [10] R. Tobar and I. Sagalaev. ijson: Iterative json parser with standard python iterator interfaces. <https://pypi.org/project/ijson/>. [Accedit: 18-mar-2025].
- [11] Plotly. Basic dash callbacks. <https://dash.plotly.com/basic-callbacks>.
- [12] Neo4j. Neo4j python manual. <https://neo4j.com/docs/python-manual/current/>. [Accedit: 24-abr-2025].
- [13] Neo4j. Cypher manual – property, structural, and constructed values. <https://neo4j.com/docs/cypher-manual/current/values-and-types/property-structural-constructed/>.

[14] Neo4j. Cypher manual – constraints. <https://neo4j.com/docs/cypher-manual/current/constraints/>. [Accedit: 8-abr-2025].

APÈNDIX

A.1 Camps disponibles al conjunt de dades inicial

Atributs d'un article		
Nom del camp	Tipus	Descripció
id	string	Identificador de l'article
title	string	Títol de l'article
authorsName	list of strings	Nom de l'autor o autors
authorsOrganization	list of strings	Organització a la que pertany l'autor o autors
authorsId	list of strings	Identificador de l'autor o autors
venue	string	Mitjà on s'ha publicat l'article
year	int	Any de publicació
keywords	list of string	Paraules clau de l'article
fieldOfStudy	string	Camps d'estudi al que pertany l'article
weightOfFieldOfStudy	float	Pes dins del camp d'estudi
references	list of strings	Identificadors dels articles referenciats per l'article en qüestió
nCitation	int	Nombre de vegades que l'article ha estat citat
pageStart	string	Pàgina inicial de l'article dins de la publicació
pageEnd	string	Pàgina final de l'article dins de la publicació
docType	string	Tipus de document
lang	string	Llenguatge utilitzat
volume	string	Volum de la publicació on apareix l'article
issue	string	Edició dins d'un volum
issn	string	Identificador de publicació periòdica
isbn	string	Identificador de llibres
doi	string	Codi per identificar i enllaçar l'article digitalment
url	list	Enllaços externs on trobar contingut de l'article
abstract	string	Resum breu de l'article
indexedAbstract	dict	Resum estructurat de l'article

A.2 Diagrama de Gantt amb la planificació inicial del projecte

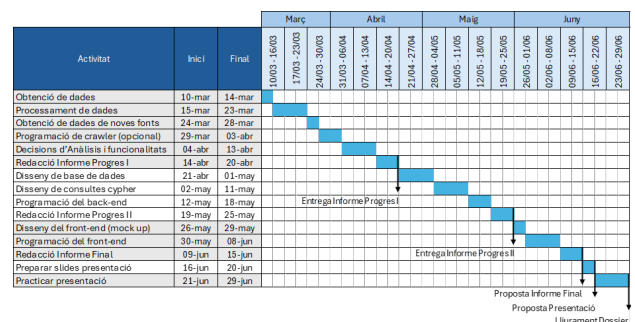


Fig. 9: Diagrama de Gantt amb la planificació inicial del projecte.