Alma Mater Studiorum
Università di Bologna
Campus di Cesena

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

Progettazione e Sviluppo di un'Applicazione Web per l'Analisi delle Pubblicazioni Scientifiche

Relatore:
Prof. Giovanni Delnevo

Presentata da:
Andrea Monaco

Correlatori:
Dr. Andruccioli Manuel
Dr. Olaiya Kevin

Sessione II
Anno Accademico 2024-2025

Introduzione

La valutazione dell'importanza delle ricerche scientifiche è oggi un aspetto di grande importanza, tuttavia presenta alcune criticità. Ricercatori e istituzioni hanno spesso bisogno di misurare qualità e impatto delle pubblicazioni, ma per farlo in modo completo è necessario fare uso di numerose piattaforme specializzate con caratteristiche molto diverse.

Il panorama degli strumenti per la valutazione scientifica oggi soffre di una significativa frammentazione dei dati e delle informazioni. Scopus e Web of Science forniscono molte informazioni a pagamento sulle varie pubblicazioni, piattaforme specializzate come DBLP forniscono solo una parte dei dati, e infine strumenti di valutazione come CORE e SCImago Journal Ranking offrono molte informazioni su conferenze e riviste, ma nessuna sulle pubblicazioni. Un ricercatore, per ottenere tutte le informazioni, deve quindi navigare tra numerose interfacce.

Se ad esempio un ricercatore volesse ottenere tutte le informazioni su tutte le sue pubblicazioni dovrebbe andare su Scopus per ottenere tutte le metriche, poi spostarsi su SCImago e per ognuno degli articoli individuare la rivista corrispettiva nell'anno di pubblicazione dell'articolo per prelevare le metriche di SCImago legate alle riviste e andare su CORE per ottenere tutte le metriche legate alle conferenze in cui vengono pubblicati gli atti di conferenza, sempre nell'anno in cui si è tenuto l'atto di conferenza.

Nella seguente tesi viene descritto in dettaglio il problema, spiegando in modo approfondito lo scopo di ognuno dei portali e le metriche principali che mostrano agli utenti, infine viene descritto il progetto di DocRanks che ha

lo scopo di mostrare tutte le informazioni dei vari siti in un'unica interfaccia web.

Attraverso l'uso di DocRanks un utente può ottenere i dati completi sui vari ricercatori in pochissimo tempo, infatti si occupa di ottenere tutte le pubblicazioni dell'autore cercato e collegarle alle loro riviste e conferenze di appartenenza. Dopodiché il sito mostra i dati sull'origine della pubblicazione nell'anno corretto, evitando i possibili errori e velocizzando molto i tempi di ricerca, che altrimenti per la maggior parte degli autori richiederebbero svariati minuti e porterebbero frustrazione nell'utente in caso di ricerche infruttuose.

Inoltre DocRanks è molto semplice da utilizzare e mostra tutti i risultati in una semplice interfaccia coerente a se stessa evitando i problemi che un utente avrebbe nel navigare numerosi siti che adottano stili e filosofie diversi per mostrare l'informazione all'utente.

L'analisi e lo sviluppo di DocRanks viene presentato nella tesi attraverso tre capitoli.

Nel primo capitolo viene fornita una descrizione approfondita di: Scopus, Web of Science, DBLP, CORE e SCImago. Questi sono i portali da cui possono essere ottenute tutte le metriche più rilevanti sulle ricerche nell'ambito informatico. Ne segue una spiegazione approfondita sui loro sistemi di valutazione e sulle metriche che offrono. Infine vengono esposti i problemi che derivano dalla necessità di navigare numerosi siti con interfacce eterogenee.

Nel capitolo successivo vengono mostrate le varie tecnologie utilizzate per sviluppare il DocRanks stesso. Le tecnologie vengono suddivise in sezioni tra tecnologie legate al Frontend, che rappresenta la parte di un'applicazione con cui un utente interagisce, e il Backend che gestisce dati e funzionamento logico del server. Infine viene descritto l'ambiente di sviluppo che rende possibile il funzionamento del sistema.

Nell'ultimo capitolo si ha una descrizione dettagliata di DocRanks, di come è stato progettato e di come funziona. Nella prima sezione viene spiegata la logica dietro il Database, che ha il compito molto complesso di contenere

i dati provenienti da i vari portali in modo che possano essere collegati tra di loro. Viene descritto il suo processo di creazione composto da Progettazione Concettuale, Logica e Fisica e quindi ne viene descritta la struttura in dettaglio. Segue una spiegazione dei servizi usati per ottenere i dati dai vari portali e infine una spiegazione approfondita sul progetto in sé e sul suo funzionamento, con immagini estratte dal sito per mostrarne anche l'aspetto estetico.

Indice

Elenco delle figure

Capitolo 1

Contesto di Progetto

Il panorama degli strumenti per la valutazione bibliometrica è caratterizzato da una molteplicità di database specializzati e sistemi di ranking che sono suddivisi tra varie piattaforme indipendenti le une dalle altre. Questa separazione comporta una non necessaria perdita di tempo per ottenere una valutazione efficace e completa delle pubblicazioni degli autori: l'utente che legge un articolo di una rivista scientifica deve andare su Scopus per saperne l FWCI (Field Weight Citation Impact), poi deve passare su SCImago Journal Ranking per sapere il valore del quartile della rivista nell'anno di pubblicazione e deve tornare su Scopus per sapere altri dati sulla rivista quali CiteScore o SNIP (Source Normalized Impact per Paper).

In questo capitolo vengono presentati i principali strumenti atti alla valutazione e indicizzazione di testi scientifici con particolare attenzione sugli strumenti riguardanti l'area informatica. Vengono esaminate le caratteristiche dei principali strumenti bibliometrici disponibili, i loro sistemi di valutazione e le problematiche derivanti dalla loro frammentazione. Infine verranno valutati gli strumenti che provano a mettere in mostra i dati e verranno analizzate i loro algoritmi.

1.1 Database Bibliografici

I database bibliografici (DB) sono generalmente definiti come raccolte digitali di riferimenti a pubblicazioni, in particolare ad articoli di riviste e atti di conferenze, che sono etichettati con titoli specifici, nomi degli autori, affiliazioni e identificativi [1].

I DB rappresentano la fonte primaria di informazioni per la valutazione della ricerca scientifica, infatti questi sono i principali fornitori di metadati utilizzati per l'analisi degli studi. A oggi i DB non sono solo utilizzati per cercare gli articoli o le riviste più rilevanti, ma anche per tenere traccia della propria carriera personale e individuare le possibili collaborazioni [2].

Web of Science (WoS) e Scopus sono i due database bibliografici riconosciuti come le fonti di dati più complete [3]. WoS è stato il primo database bibliografico internazionale di grande diffusione. Pertanto, col tempo, è diventato la fonte di dati bibliografici più influente. È tradizionalmente utilizzato per la selezione delle riviste, la valutazione della ricerca, le analisi bibliometriche e altri compiti [4]. WoS è stato l'unica fonte di dati bibliografici per più di 40 anni, fino al 2004, quando Scopus è stato lanciato da Elsevier [5]. Nel corso degli anni, Scopus si è guadagnato un posto alla pari come fonte di dati bibliografici e si è dimostrato affidabile e, per certi versi, persino superiore a WoS [2, 3, 6].

Uno dei problemi che caratterizza sia Scopus che WoS è rappresentato dal costo che rende l'uso di entrambi una prerogativa tipica di ricercatori e studenti, in più molte istituzioni per limitare le spese si iscrivono solo a uno dei due colossi [2]. In questo si distinguono alcuni Database specializzati come Digital Bibliography & Library Project (DBLP) che è specialistico del settore informatico, e dunque meno completo dei due sopracitati, ma ad accesso libero e gratuito.

1.1.1 Scopus

Scopus è un database bibliografico sviluppato da Elsevier e lanciato nel 2004 come alternativa al Web of Science. Attualmente è uno dei più completi e autorevoli disponibili, indicizza oltre 28.900 riviste e circa 399.000 libri. Al momento del suo lancio conteneva circa 27 milioni di voci pubblicate (1966–2004). Da allora, il contenuto del database è cresciuto fino a oltre 78 milioni di voci, coprendo pubblicazioni dal 1788 al 2024, rendendolo uno dei più grandi database bibliografici oggi disponibili. Ogni anno vengono aggiunti circa 3 milioni di nuovi documenti [5]. Scopus ha più citazioni di WoS generalmente derivanti da fonti non di lingua inglese [7].

La Gestione dei Dati

L'architettura di Scopus è parzialmente ottenuta tramite algoritmi automatici e revisionata manualmente, ciò garantisce un'elevata qualità dei contenuti. Scopus solitamente sfrutta degli algoritmi per identificare gli autori e riconoscere le citazioni. Tuttavia parte del processo di selezione dei contenuti è gestito dal Content Selection and Advisory Board (CSAB), un comitato indipendente di esperti che valuta la qualità delle fonti indicizzate e ne garantisce l'affidabilità nel tempo [5].

Per migliorare ulteriormente l'accuratezza dei profili autoriali Scopus impiega un "gold set" di circa 12.000 autori che verificano le informazioni sui vari profili. Scopus stima una precisione media del 98,1% e richiamo del 94,4% nella corretta attribuzione delle pubblicazioni. Un ulteriore strumento a disposizione degli autori per migliorare l'accuratezza è l'Author Feedback Wizard (AFW), disponibile sul portale di Scopus, che permette ai ricercatori di richiedere direttamente modifiche ai propri profili per correggere eventuali errori [5].

Scopus include articoli pubblicati a partire dal 1788, ma ha copertura completa sulle citazioni solamente a partire dal 1996.

Infine il portale fornisce delle API che permettono di accedere alle informazioni bibliometriche presenti nel database senza utilizzare interfacce web.

Queste API sono basate su un'architettura RESTful e permettono di restituire i dati in diversi formati, tra cui json o xml, semplificando l'integrazione con siti web o in software non sviluppati direttamente da Elsevier.

Metriche e Indicatori

Scopus fornisce una serie di metriche collegate all'autore tra le quali le più importanti sono: Numero di pubblicazioni, Numero di Citazioni, Citazioni per documento e H-index. In più ci sono metriche legate a articoli e riviste, oltre al numero di citazioni ci sono metriche proprietarie quali:

- Field-Weighted Citation Impact (FWCI): rapporto tra citazioni ricevute e citazioni attese nel settore.
- CiteScore: media delle citazioni ricevute per documento negli ultimi 4 anni, è l'alternativa proprietaria di Scopus all'Impact Factor di WoS.
- Source Normalized Impact per Paper (SNIP): impatto delle citazioni normalizzato per il campo di ricerca.

Limitazioni

Nonostante la sua completezza, Scopus presenta alcune limitazioni significative quali:

- ha una ridotta copertura delle conferenze, in particolare nel settore informatico, e non permette di valutare la qualità di un atto di convegno basandosi sul prestigio della conferenza di cui è parte poichè non ha metriche associate ad essa;
- non fornisce informazioni riguardo i quartili delle riviste, in quanto calcolati da SCImago, quindi non si può valutare un articolo sulla base del quartile della rivista su cui è stato pubblicato, a meno che non lo si ricerchi altrove.

Anche le api scopus hanno delle limitazioni che possono creare problemi:

- il rate limit delle api è molto restrittivo, 5000 richieste settimanali per utenti base;
- la maggior parte delle informazioni sono precluse all'api base.

1.1.2 Web of Science

Web of Science (WoS), sviluppato da Clarivate Analytics è un database bibliometrico multidisciplinare fondato nel 1997, con il nome di Web of Knowledge, dall'Institute for Scientific Information (ISI). WoS ha una copertura che parte dal 1900, contiene 79 milioni di articoli selezionati direttamente dall'Editorial Board di Calrivate, "Core Collection", e ospita un totale di 171 milioni di articoli.

Metodo di Selezione

Secondo la filosofia di WoS vengono selezionate le riviste, i libri e gli atti di conferenze attraverso degli editor di Clarivate, le riviste vengono valutate secondo vari criteri di qualità. Se una rivista soddisfa i suddetti criteri allora viene inserita nella sezione Emerging Sources Citation Index (ESCI) della collezione Core di WoS, se aumenta ulteriormente di importanza viene spostata in un altra sezione, viceversa può essere espulsa.

WoS è strutturato in diverse collezioni specializzate:

- Science Citation Index Expanded (SCIE) per le scienze naturali,
- Social Sciences Citation Index (SSCI) per le scienze sociali,
- Arts & Humanities Citation Index (AHCI) per le discipline umanistiche,
- Emerging Sources Citation Index (ESCI) per le riviste emergenti,
- Book Citation Index (BCI) per i libri,

- Conference Proceeding Citation Index (CPCI) per gli atti delle conferenze.

Questa suddivisione serve a favorire i ricercatori nelle analisi e aiuta a inserire nuove riviste al database utilizzando la sezione ESCI per mettere alcune pubblicazioni in osservazione.

Metriche e Indicatori

Web of Science (WoS) fornisce un insieme di metriche bibliometriche ampiamente utilizzate nella valutazione della ricerca scientifica. Come in Scopus le metriche non si limitano al conteggio delle citazioni, ma includono indicatori normalizzati che permettono di confrontare pubblicazioni e riviste in discipline diverse.

A livello di autore, WoS, come scopus, fornisce: Numero di pubblicazioni, Numero di Citazioni, Citazioni per documento e H-index. A livello di rivista e articolo vengono integrate metriche proprietarie sviluppate da Clarivate, tra le quali quelle di maggiore importanza sono:

- Impact Factor (IF): calcolato come il numero medio di citazioni ricevute in un anno dagli articoli pubblicati nei due anni precedenti. È la metrica storicamente più utilizzata per valutare le riviste indicizzate nel *Journal Citation Reports* ed è il corrispettivo del CiteScore di Scopus.

- Journal Citation Indicator (JCI): metrica normalizzata introdotta da Clarivate nel 2021, che permette di confrontare l'impatto delle riviste indipendentemente dal campo disciplinare.

Limitazioni e Bias

Web of Science è stato oggetto di critiche per bias geografici e linguistici, con una sovrarappresentazione di riviste nordamericane ed europee pubblicate in inglese [2], per far fronte a questo nel 2008 WoS ha integrato delle sezioni dedicate a articoli in altre lingue pubblicati in regioni non di lingua inglese, ma non ha risolto il problema.

L'approccio selettivo di WoS garantisce un buon livello qualitativo dei suoi contenuti, tuttavia tente a escludere numerose pubblicazioni che sono invece presenti su Scopus [8].

Come Scopus anche WoS non da' nessuna informazione sulla rilevanza della conferenza in cui vengono pubblicati gli atti e non è in grado di fornire alcune informazioni proprietarie come lo SNIP o lo Scimago Journal Ranking (SJR) sulle riviste in quanto sono metriche proprietarie di altre aziende.

1.1.3 DBLP

La Digital Bibliography & Library Project è il riferimento online per le informazioni bibliografiche relative all'ambito informatico a accesso gratuito. Negli anni è diventata un servizio di open data molto popolare per l'intera comunità scientifica.

A partire da gennaio 2024, DBLP indicizza oltre 7 milioni di pubblicazioni, prodotte da più di 3,4 milioni di autori. A tal fine, DBLP indicizza circa 55.000 volumi di riviste, più di 55.000 atti di conferenze e workshop, e oltre 140.000 monografie [9].

Al contrario dei DBs analizzati finora DBLP è focalizzato all'ambito informatico e non fornisce molti metadati quali: numero di citazioni, h-index e impact factor o CiteScore. Tuttavia l'accesso a DBLP è libero e gratuito.

Validità dei Metadati

DBLP solitamente ottiene tutti i metadati necessari direttamente dall'editore del volume stesso o dall'organizzatore degli eventi. Quando questo non è possibile i metadati vengono inviati al portale da volontari. Ottenuti i dati un editore del team li verifica aiutandosi con degli algoritmi e infine aggiunge i dati al dataset del sito. Nei giorni successivi vengono segnalate da degli script le inconsistenze sui dati aggiunti recentemente [9].

Si ritiene che i dati di DBLP siano estremamente accurati e affidabili, ma talvolta DBLP non riesce a identificare correttamente autori omonimi [10].

DBLP rende disponibili i propri dati attraverso diverse interfacce senza restrizioni dovute a licenze. Il database ha dell'API che permettono di ottenere metadati xml senza accedere a interfacce grafiche, dunque è possibile inserire i dati di DBLP in siti o applicazioni indipendenti senza limiti di utilizzo o costi.

Limitazioni

La principale limitazione di DBLP è la mancanza di informazioni su citazioni, riviste e conferenze. Per ottenere dati come le citazioni di una rivista o un articolo bisogna usufruire di Scopus, Google Scholar o di WoS, tuttavia anche questi mancano di informazioni riguardo le conferenze. Infine DBLP non copre altre aree scientifiche oltre a quella informatica, rendendolo dunque inadatto per programmi e studi più ampi. Dunque DBLP è un'opzione valida solo dove l'informatica ha un'importanza centrale.

Queste caratteristiche lo rendono uno strumento da integrare con altri che non fornisce abbastanza informazioni sulle ricerche in autonomia.

1.1.4 Spiegazione delle Metriche Bibliometriche

In questa sezione sono state introdotte numerose metriche, prodotte soprattutto da WoS e Scopus, queste metriche sono fondamentali per la valutazione delle riviste e del prestigio degli autori, dunque di seguito vengono approfondite per permettere una maggiore comprensione dell'argomento.

H-index

La prima è l'h-index (indice di Hirsch), metrica che indica il numero massimo N di pubblicazioni i tali per cui $N <= min(C_i)$ dove C_i è il numero di citazioni della pubblicazione i. Jorge Hirsch propose la metrica nel 2005 e ad oggi è uno dei dati di maggior rilievo per valutare un autore ed è presente sia su Scopus che su WoS, tuttavia visto che il valore di h-index è calcolato dalle riviste e dalle citazioni che può variare tra Scopus e WoS [11].

Questa metrica misura assieme quantità e qualità delle pubblicazioni offrendo un indicatore dell'impatto di un autore sulla comunità scientifica.

Vantaggi dell'h-index:

- non può essere facilmente incrementato con pubblicazioni di bassa qualità;
- è semplice da calcolare e interpretare;
- può riguardare anche le riviste, conferenze o istituzioni oltre che le persone.

Tuttavia l'h-index talvolta fornisce informazioni fuorvianti, infatti la correlazione tra h-index e riconoscimenti scientifici è calata dal 2010 [7]. Questo sembra parzialmente dovuto a diverse criticità:

- Non tiene conto del numero di autori di un articolo. Originariamente Hirsch suggerì di ripartire le citazioni tra i vari coautori, questo indice è noto come fractional h-index, ma non è diffuso tra i vari strumenti automatici [7].
- Non considera le differenze tra le varie aree di studio, dunque gli autori di discipline diverse non sono confrontabili con l'h-index.
- È insensibile a pubblicazioni eccezionalmente influenti.
- Favorisce ricercatori senior.

Field-Weighted Citation Impact (FWCI)

Il Field-Weighted Citation Impact è una metrica introdotta e utilizzata da Elsevier in Scopus con il fine di normalizzare le citazioni in base a area disciplinare e tempo. Il FWCI è il rapporto tra il numero di citazioni che ci si aspetta da una pubblicazione di un certo tipo e il numero di citazioni effettivamente ricevute.

Quindi la formula che ne deriva è:

$$FWCI = \frac{\text{Citazioni ricevute}}{\text{Citazioni attese per campo, anno e tipo di documento}}$$

Alla luce di questa definizione si intuisce che un FWCI:

- uguale a 1: rappresenta una produzione scientifica nella media.
- minore di 1: indica che la pubblicazione è stata poco citata.
- maggiore di 1: indica un numero di citazioni maggiore delle aspettative per la pubblicazione.

Grazie al suo funzionamento il FWCI permette di confrontare tra loro pubblicazioni di tipo diverso, pubblicate in periodi diversi e di categorie diverse. È quindi una metrica che permette di valutare in modo obiettivo atti di convegno, articoli o altre pubblicazioni senza bias legati alla loro categoria disciplinare o al tempo da cui sono state pubblicate.

Impact Factor

L'impact Factor (IF) è una metrica molto importante usata da Clarivate Analytics, produttore di WoS, per valutare le riviste. L'impact factor misura il numero di citazioni ricevute da una rivista su articoli degli ultimi due anni, quindi l'IF misura l'impatto che ha avuto una rivista sulla comunità scientifica negli ultimi anni.

$$IF_{anno} = \frac{\text{Citazioni nell'ultimo anno a pubblicazioni di (anno-1) e (anno-2)}}{\text{Pubblicazioni citabili in (anno-1) e (anno-2)}}$$

L'Impact Factor è stato storicamente molto criticato per svariati motivi quali:

- L'algoritmo non è trasparente poichè non si può sapere quali articoli vengono presi in considerazione per il calcolo.

- Il termine "Pubblicazioni Citabili" non è chiaro e non ne fanno parte alcune pubblicazioni che però vengono effettivamente citate alterando il valore finale.
- La finestra temporale è limitata (2 anni), il che non permette sempre una valutazione obiettiva della rivista.
- È facilmente manipolabile dalle riviste che possono richiedere agli autori di citare i loro stessi articoli.
- È molto variabile tra argomenti e discipline, non rispecchia la qualità effettiva delle opere.
- Le riviste vengono selezionate da WoS che esclude molte pubblicazioni valide.

CiteScore

Il CiteScore è una misura creata da Elsevier per sostituire l'IF, le principali differenze stanno nella totale trasparenza e riproducibilità del calcolo e nella finestra di 4 anni e non di 2

$$CiteScore = \frac{\text{Citazioni ricevute in 4 anni}}{\text{Documenti pubblicati in 4 anni}}$$

Questa metrica offre maggiore stabilità rispetto all'Impact Factor ma ne condivide alcuni difetti legati alla manipolabilità dei dati e all'impossibilità di paragonare riviste trattanti argomenti diversi.

Source Normalized Impact per Paper

Il Source Normalized Impact per Paper (Snip) come il CiteScore è una misura calcolata da Elsevier/Scopus che come il CiteScore cerca di misurare l'impatto di una rivista, ma viene normalizzato per campo disciplinare. Dunque permette di paragonare tra loro riviste di tipologia diversa [12].

Lo Snip è definito come il rapporto tra il Raw Impact per Paper (RIP) e il Database Citation Potential (DCP).

Il RIP è il rapporto tra il numero di citazioni ricevute da una rivista e il numero di pubblicazioni degli ultmi 3 anni.

$$RIP = \frac{\text{citazioni rivista in 3 anni}}{\text{pubblicazioni rivista in 3 anni}}$$

Invece il DCP è il numero di riferimenti a pubblicazioni degli ultimi tre anni che appartengono alla categoria disciplinare della rivista.

$$DCP = \frac{1}{3} \times \frac{n}{\frac{1}{p_1 r_1} + \frac{1}{p_2 r_2} + \cdots + \frac{1}{p_n r_n}}$$

Dove:

- n = numero di pubblicaioni del campo della rivista
- r_i = numero di citazioni ricevuti negli ultimi 3 anni dalla pubblicazione i
- p_i = data una pubblicazione i p_i è il rapporto tra tutte le pubblicazioni della rivista dell'anno in cui è stata pubblicata p_i con almeno una citazione negli ultimi 3 anni e il numero totale di pubblicazioni della rivista dell'anno in cui è stata pubblicata i

Dunque:

$$SNIP = \frac{RIP}{DCP}$$

1.2 Strumenti per la Valutazione Qualitativa

Con il progressivo sviluppo dei database bibliografici si sono affermati degli strumenti il cui scopo principale è la valutazione qualitativa delle riviste e delle conferenze. Attraverso questi strumenti è possibile contestualizzare le pubblicazioni sulla base della qualità della pubblicazione di cui fanno parte. Due articoli con un simile numero di pubblicazioni possono avere un peso diverso nella valutazione di una carriera di un autore sulla base del valore della rivista su cui ha pubblicato, dunque questi dati rappresentano uno strumento fondamentale alla valutazione bibliometrica.

Gli strumenti principali che verranno introdotti in questa sezione sono COmputing Research and Education (CORE) Ranking e SciMago Journal Ranking (ScimagoJR), strumenti che valutano, rispettivamente, le conferenze e le riviste in vari modi.

1.2.1 CORE Rankings

COmputing Research and Education (CORE) Rankings è uno degli strumenti di valutazione qualitativa di riferimento per quanto riguarda conferenze nel campo informatico.

CORE è un progetto australiano del 2005 con lo scopo di assegnare un ranking alle conferenze seguite dagli studenti di informatica e telecomunicazioni. Negli anni ha acquisito un'importaza sempre maggiore e ad oggi il ranking assegnato da CORE è riconosciuto a livello internazionale.

Metodologia di Valutazione

La metodologia CORE si basa su un processo di peer review condotto da gruppi di esperti nelle loro rispettive aree. I criteri di valutazione includono il prestigio della conferenza, l'impatto degli atti presentati, l'importanza degli autori che vi partecipano e l'importanza per la comunità informatica.

Il sistema utilizza una scala di classificazione a quattro livelli nel seguente ordine di importanza:

- A*: Conferenze e riviste di eccellenza internazionale, sono le conferenze migliori o al pari delle migliori del loro settore, gli atti che ne fanno parte sono molto citati e sono note anche all'infuori della comunità strettamente informatica.

- A: Conferenze con altissimo livello qualitativo che però sono meno note delle conferenze A*, i paper che ne fanno parte sono tipicamente paragonabili a quelli A* come qualità.

- B: Conferenze buone e molto buone, gli atti sono tipicamente scritti da autori che non sono necessariamente di grande importanza, ma alcuni atti di queste conferenze sono comunque di alto livello.

- C: Conferenze che raggiungono il livello considerato minimo da CORE, solitamente rispetto le altre conferenze hanno una rilevanza minore.

Questa classifica è utilizzata da ricercatori, università e istituzioni per valutare le conferenze e individuare quelle di maggior rilievo, questo ranking è il fulcro di CORE e il motivo della sua esistenza.

Processo di Aggiornamento

Per valutare le conferenze si riuniscono dei comitati di esperti internazionali che rappresentano diverse università in modo di mantenere la maggiore imparzialità regionale possibile. Un comitato per valutare una conferenza considera un grande numero di dati come le citazioni degli atti che appartengono alla conferenza in questione e l'autorevolezza (basata sui dati di scopus) degli autori che partecipano alla conferenza. Dal 2023 database di CORE verrà aggiornato ogni tre anni, per essere inserite le conferenze devono rispettare le seguenti regole:

- la conferenza non deve essere nazionale/regionale, a meno che non sia di grande importanza;

- gli atti della conferenza devono essere inseriti su Scopus e su DBLP;

- gli atti devono aver superato il processo di peer review;

- le conferenze devono svolgersi a cadenza regolare e devono pubblicare mediamente più di 10 atti per anno.

Il processo di CORE è trasparente e queste regole stringenti rendono il ranking di core un dato di grande importanza nella valutazione delle conferenze.

1.2.2 SciMago Journal Rank

SCImago Journal & Country Rank è un portale pubblico che include indicatori scientifici di riviste sviluppati dalle informazioni contenute in Scopus. Questi indicatori possono essere utilizzati per valutare e analizzare i domini scientifici. Le riviste possono essere raggruppate per area (27 aree tematiche), categorie (313 categorie specifiche) o per paese. I dati delle citazioni provengono da oltre 34.100 testate da oltre 5.000 editori [13].

Metodologia e Algoritmo

Scimago valuta le riviste in particolare attraverso l'uso del suo indicatore SJR. La filosofia è che le citazioni identificano il valore di una rivista e che le citazioni che avvengono su riviste più autorevoli hanno un peso maggiore.

L'SJR considera citazioni ricevute nell'anno corrente riguardanti articoli pubblicati nei tre anni precedenti. Dunque ha una finestra maggiore rispetto all'Impact Factor.

L'algoritmo dell'SJR è ispirato all'algoritmo di PageRank di Google. Esso si basa su un grafo completo in cui ogni rivista è un nodo a cui è assegnato un valore, il valore dei nodi aumenta con le citazioni ricevute e diminuisce con le citazioni fatte. L'algoritmo segue i seguenti passaggi:

1. A ogni rivista viene assegnato un valore di SJR, il valore non è rilevante sul risultato finale, ma solo sul numero di iterazioni.

2. Se A cita B allora parte del SJR di A viene passato a B. Dunque B diventa più prestigiosa.

   In particolare A ha passato a B SJR/citazioni fatte da A.

3. Avviene uno scambio completo di valori tra tutti i nodi.

4. Si rieseguono i passi precedenti mantenendo però l'SJR invariato.

5. Quando i valori di SJR cambiano meno di un certo limite tra le iterazioni il risultato converge.

Il processo iterativo converge verso un SJR stabile per ogni rivista che riflette sia la quantità che la qualità delle citazioni ricevute.

La formula matematica semplificata per il calcolo dell'SJR è:

$$SJR_i = \frac{1-d}{N} + d\sum_{j \neq i} \frac{C_{ji}}{C_j} \cdot SJR_j$$

dove d è una costante, di solito $0,85$, N il numero totale di riviste, C_{ji} il numero di citazioni dalla rivista j alla rivista i, e C_j il numero totale di citazioni della rivista j.

I Quartili

Sfruttando l'SJR viene calcolata anche l'altra metrica importante ottenuta da SCImago ovvero i quartili.

I quartili sono un modo per calssificare le riviste sulla base del loro SJR all'interno della loro categoria disciplinare. Nel portale ci sono, infatti, 313 categorie e ogni rivista appartiene a una o più di queste. Per calcolare i quartili vengono selezionate tutte le riviste di ognuna delle categorie e vengono ordinate secondo l'SJR. Dunque si fa una divisione in quartili in cui:

- Q1: riguarda il top 25% delle riviste;
- Q2: tra il 25% e il 50% superiore.
- Q3: tra il 50% e il 75%.
- Q4: 25% più basso.

Le riviste Q2 sono generalmente considerate di buona/alta qualità nelle valutazioni accademiche, invece quelle in Q1 sono le riviste più prestigiose del settore.

1.3 La Frammentazione dei Dati

Ognuno dei database e strumenti analizzati offre una grande mole di informazioni per valutare riviste, autori, conferenze e pubblicazioni, ma ottenere

un'informazione completa risulta complesso. Infatti se un utente decide di ottenere informazioni riguardo a una specifica pubblicazione deve:

- Individuare la pubblicazione su Scopus per avere FWCI, numero di citazioni, autore e tipologia della rivista.
- Ipotizzando che si tratti di un atto di una conferenza deve andare su CORE e trovare il ranking della conferenza di cui è stato parte in quell'anno per avere delle informazioni sul valore della conferenza.
- Se invece si tratta di una rivista deve restare su scopus per trovare SNIP e CiteScore della rivista in quell'anno e poi passare a SCImago nell'eventualità che si vogliano avere delle informazioni sui quartili.

Questo lavoro non solo è uno spreco di tempo, ma richiede anche all'utente di navigare tra siti molto diversi tra di loro creando confusione, in più i vari siti spesso assegnano nomenclature diverse alle conferenze e alle riviste e questo può portare frustrazione nell'utente. In questa sezione verranno analizzate in dettaglio le problematiche derivanti da questa frammentazione delle informazioni.

1.3.1 Eterogeneità delle Interfacce Utente

Il problema principale è l'estrema eterogeneità delle interfacce utente:

Scopus: Scopus è caratterizzato da una interfaccia utente moderna con numerosi grafici e schermate interattive, è pensata per ricercatori che necessitano di strumenti avanzati. Tuttavia non è sempre intuitivo trovare ciò che si cerca nel sito per un pubblico inesperto.

DBLP: DBLP invece ha un approccio minimalista, l'interfaccia è molto semplice e intuitiva, al contrario di Scopus non presenta alcun grafico e ha un estetica spartana, probabilmente dovuta al fatto che DBLP nasce da un progetto universitario.

CORE portal: CORE presenta l'interfaccia più semplice di tutte, ricorda un motore di ricerca testuale, come Google Scholar, e i risultati che offre sono raggruppati tipicamente in una sola grande tabella.

SCImagoJR: Scimago ha un interfaccia moderna e intuitiva, presenta i dati delle varie riviste con l'ausilio di grafici e tabelle interattive, ottime al fine di confrontare le riviste.

Ognuno di questi siti ha optato per diversi paradgmi per offrire una propria versione dei dati, ognuno è contraddistinto dai propri colori, schemi e "family fealing" basato su scelte estetiche e funzionali caratteristiche, che ne riflettono gli obiettivi, ma che non si amalgamano bene tra di loro.

1.3.2 Inconsistenze nei Dati

Oltre ai problemi legati alla UI, sono presenti numerosi problemi riguardo alla raccolta dei dati e delle informazioni stesse.

Copertura Informativa

Ognuna delle fonti di dati fornisce solo parte dell'infomazioni, l'utente deve memorizzare dove trovare le informazioni mancanti e cosa cercare su ogni sito:

- Scopus fornisce tutte le informazioni riguardo un autore e le sue pubblicazioni, ma fornisce solo il nome e la data delle conferenze e fornisce varie informazioni sulle riviste, tra cui CiteScore e SNIP, ma non fornisce informazioni sui quartili.
- DBLP fornisce numerose informazioni riguardo alle pubblicazioni gratuitamente, ma non fornisce informazioni riguardo alle citazioni e fornisce solo pochi dati riguardo a riviste e conferenze legati alle pubblicazioni.
- Scimago fornisce, tra i dati più importanti, SJR e Quartili di una rivista, senza fornire nulla sulle pubblicazioni che ne fanno parte, o sugli autori.

- CORE fornisce il ranking di alcune conferenze e altri dati legati ad essa, senza alcuna infomazione sugli atti che ne fanno parte.

Un utente inconsapevole può quindi avere un'idea incompleta di una pubblicazione.

1.3.3 Impatto sulla User Experience

Ogni piattaforma ha optato per una propria interfaccia utente (UI), con diversi stili, colori e logiche di navigazione. Tuttavia questo ha creato un sistema disomogeneo che può ostacolare l'utente e compromette la User Experience (UX).

Per trovare le informazioni che servono un utente deve:

- Ricordare il sito da cui trarre le informazioni.
- Saper interagire correttamente con tutte le UI di tutti i siti.
- Riuscire a collegare tra di loro le diverse informazioni per individuare su siti diversi lo stesso oggetto.

Conseguentemente l'utente deve imparare a navigare tra i vari siti memorizzando le informazioni necessarie a ricercare gli oggetti tra i vari portali e il funzionamento delle loro interfacce.

Capitolo 2

Tecnologie Utilizzate

In DocRanks sono stati usati vari softare e applicativi web con lo scopo di creare un'applicazione funzionale per mostrare i dati ottenuti da vari portali in un unico software che li raduna tutti.

I sistemi utilizzati sono:

- HyperText Markup Language (HTML): linguaggio utilizzato per scrivere il codice delle varie pagine di DocRanks.
- Cascading Style Sheets (CSS) e Bootstrap: utilizzati per garantire una grafica accattivante.
- Hypertext Preprocessor (PHP): linguaggio utilizzato per fare il backend del sito, fare le richieste alle api dei vari portali da cui prendere le informazioni, interpretare i file per inserire i dati nel database e leggere e scrivere i dati sul database stesso di DocRanks.
- MySQL: il Database Management Sistem (DBMS) relativo al database usato nel progetto.
- Apache: software per usare le pagine web tramite l'uso di un browser.
- Docker: softare utilizzato per far funzionare il sito tramite la creazione di container.

- Git: usato per versionare DocRanks.

In questo secondo capitolo vengono mostrate in dettaglio le tecnologie sopracitate spiegandone lo scopo nel progetto e soffermandosi anche sulla loro storia di sviluppo e cosa le ha rese famose.

2.1 Frontend

DocRanks è una web app, dunque nel suo sviluppo si è fatto uso di HTML e CSS. Questi linguaggi si occupano esclusivamente della parte grafica e presentazionale del sito e sono alla base del frontend del progetto.

Questa scelta è stata fatta alla luce di vari vantaggi e considerazioni tra cui le principali sono:

- la possibilità di mettere online l'app e accedere ovunque.
- l'intercompatibilità con ogni dispositivo a condizione che possa accedere a un browser.
- la possibilità di utilizzare tecnologie semplici come HTML e CSS e numerosi framework per lavorare alle grafiche del sito.
- in caso di deploy gli aggiornamenti di una interfaccia web sono centralizzati, dunque non a carico dell'utente
- dato che DocRanks per sua stessa natura dipende da internet non è un problema la necessità di avere accesso a internet per usare una web app.

Tuttavia questo porta a importanti svantaggi dal punto di vista delle prestazioni, infatti un sito è notevolmente più lento di un'app nativa, in particolare nel caso di DocRanks la navigazione è completametne gestita lato server, quindi ogni volta che un utente clicca su un link o invia un modulo viene inviata una richiesta HTTP al server che risponde con un'intera pagina HTML.

2.1.1 HTML5

HTML5 è l'ultima iterazione di HTML ed è la tecnologia utilizzata per strutturare le pagine di DocRanks. HTML è stato sviluppato da Tim Berners Lee con il protocollo HTTP che serviva a trasferire documenti HTML. La quinta iterazione è stata introdotta nel 2014 dal World Wide Web Consortium (W3C), ha anche introdotto molti elementi semantici e ha reso possibile la creazione di applicazioni web interattive.

Oggi HTML è il linguaggio più utilizzato per i documenti web. La scelta dell'HTML per la creazione di siti web è infatti obbligata, tutti i browser leggono questo formato seguendo le linee guida dello standard ufficiale, utilizzare un'interfaccia web implica il dover usare l'HTML come linguaggio di mark-up o comunque doverne convertire uno a HTML.

Caratteristiche Principali

HTML descrive le modalità di impaginazione e le funzioni delle sue sezioni attraverso dei tag. Ogni pagina HTML inizia con la dichiarazione del tipo dell'elemento con la scrittura `<!DOCTYPE html>` che indica che verrà usato HTML5. Dopodiché viene tipicamente usato il tag `<html>` in cui viene racchiusa l'intera pagina. Ogni pagina ha poi un'intestazione `<head>` che contiene informazioni non visibili sulla pagina e il `<body>` che racchiude le informazioni che vengono effettivamente mostrate a schermo.

DocRanks fa uso di numerosi elementi semantici per strutturare le varie parti dell'interfaccia tra cui i principali sono:

- `<nav>` per la creazione della barra di navigazione
- `<main>` per identificare il contenuto principale
- `<section>` per suddividere in parti le pagine
- `<article>` per mostrare dettagli su articoli e atti di conferenza

2.1.2 Grafica

Ad oggi l'aspetto visivo di una pagina web rappresenta una componente fondamentale del successo di un sito, poiché è ciò che trasmette agli utenti la prima impressione e ne definisce l'esperienza d'uso. Un design ben fatto porta l'utente ad avere un maggior interesse per il sito. Inoltre una grafica ben strutturata semplifica la navigazione e guida l'utente verso le azioni desiderate.

In questo contesto gli strumenti principali utilizzati nella progettazione di DocRanks sono stati CSS3 in combinazione con Bootstrap.

CSS3: è lo standard più diffuso e utilizzato per formattare le pagine web.

Bootstrap: è un framework di CSS che fornisce vari componenti grafici che aiutano a rendere il sito più accattivante.

2.1.3 CSS3

CSS3 è l'ultima iterazione del CSS ed è utilizzato per rendere più accattivanti le pagine web.

Prima della nascita del CSS per migliorare la grafica delle pagine si era iniziato a fare uso di tag HTML non standard e a usare le tabelle del tag `<table>` non per fare delle tabelle, ma al fine di impaginare le pagine web in modi diversi.

Per risolvere questi problemi il W3C definì le specifiche del CSS che voleva portare la formattazione e lo stile grafico su file completamente separati da quelli che comprendevano il codice HTML.

Caratteristiche Principali

Il CSS viene solitamente inserito nel tag `<head>` della pagina HTML attivareso la riga:
`<link rel="stylesheet" type="text/css" href="style.css"/>`,
dove viene inserito nell'attributo "href" il percorso del file CSS contente le

regole di stile da applicare alla pagina. In alternativa è possibile scrivere il CSS direttamente all'interno del tag `<style>` sempre nell'head della pagina.

La struttura del CSS è simile ad un elenco di regole, dove un selettore indica un elemento del documento HTML a cui applicare le regole, una proprietà definisce un elemento da modificare, come il colore dello sfondo, e infine si associa un nuovo valore alla proprietà. Un esempio di codice css è mostrato nel codice 2.1.

Listing 2.1: Esempio di codice CSS

```
selettore {
    proprietà: valore;
}
```

2.1.4 Bootstrap

Bootstrap è un framework CSS open source sviluppato da Mark Otto e Jacob Thornton presso Twitter. Fornisce numerosi strumenti utili alla creazione di pagine web che rendono le applicazioni dinamiche e accattivanti.

Caratteristiche Principali

Bootstrap usa un sistema a 12 colonne che definiscono il layout della pagina e permettono di adattare automaticamente quest'ultimo alle dimensioni dello schermo o della finestra del browser favorendo la fruizione dei contenuti sia da pc che da telefono. Inoltre include numerose componenti grafiche predefinite, come pulsanti, alert, tabelle, ecc... che possono essere implementate in un tag tramite l'attributo "class".

Bootstrap è uno strumento molto semplice da utilizzare ed è anche uno dei più noti e diffusi al giorno d'oggi.

Alternative

Mentre l'uso di HTML e CSS sono ovvie derivazioni del fatto che si è scelto di utilizzare un'interfaccia web, la scelta del framework CSS da utilizzare non è obbligata, oltre a Bootstrap esistono varie alternative molto famose che sono state prese in considerazione in fase progettuale.

Secondo il quanto riportato da `state of css` nel 2024 Bootstrap non è il framework più usato [14], ma il più usato è invece `Tailwind CSS` (Fig. 2.1).

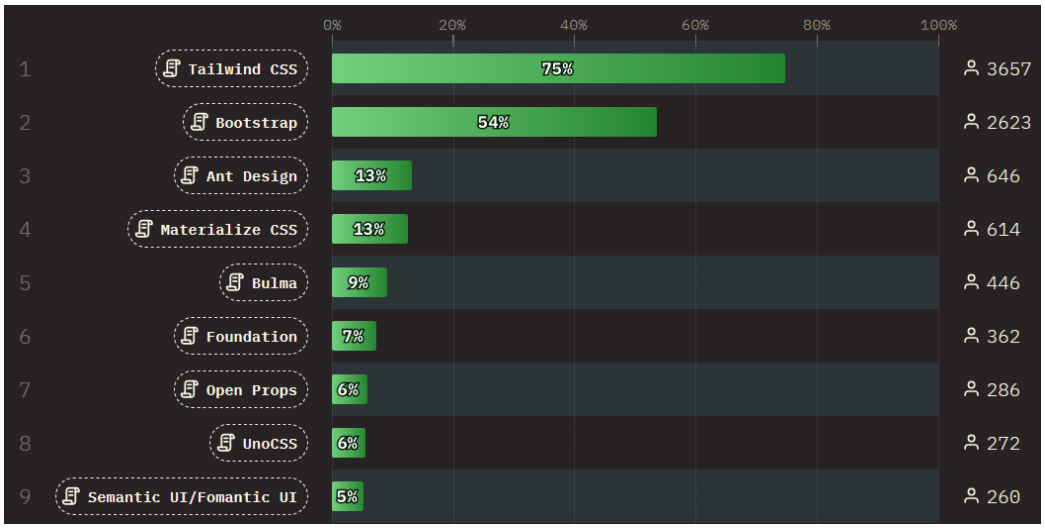


Figura 2.1: il report di stateofcss2024 mostra l'uso dei vari framework CSS

Le principali alternative a Bootstrap prese in considerazione sono state:

Tailwind CSS: tra tutte le possibili alternative la più famosa e utilizzata, le sue classi rappresentano tipicamente una singola proprietà del CSS. È molto efficiente e permette un maggior controllo di Bootstrap, ma la sua implementazione richiede più tempo.

Bulma: è un altro framework preso in considerazione per il progetto, ma alla fine si è preferito l'uso di Bootstrap poiché, come Tailwind, Bulma richiede un maggior tempo per essere implementato avendo classi simili alle proprietà del CSS stesso.

2.2 Backend

Il PHP è un linguaggio di scripting interpretato alla base del backend dell'applicazione, nel progetto si occupa di varie funzioni tra cui:

- lettura di file `.csv`,
- comunicazione e integrazione con API esterne,
- interazione con il database.

Il PHP nasce nel 1994 e allora il significato originale era Personal Home Page [15]. Dopo venne inserita la possibilità di far interagire il PHP con l'HTML per creare pagine dinamiche e nel tempo iniziò ad acquisire sempre più popolarità.

2.2.1 Programmazione a Oggetti

La programmazione orientata agli oggetti (OOP) è un paradigma di programmazione che organizza il codice in oggetti che interagiscono tra di loro. Ogni oggetto rappresenta un'entità con attributi e metodi che ne definiscono il comportamento.

OOP in PHP

Il sviluppo dell'OOP in PHP è stato avviato con la terza versione di PHP, ma solo con la versione 5 si è iniziato a avere un supporto completo all'OOP con l'introduzione di classi, ereditarietà, incapsulamento e polimorfismo. Un esempio di ereditarietà è mostrato al codice 2.2.

Listing 2.2: Esempio di classe PHP con ereditarietà

```
class Product {
    protected $name;
    protected $price;

    public function formatPrice() {
        return "$" . number_format($this->price, 2);
    }
}

class DigitalProduct extends Product {
    private $downloadLink;

    public function getDownloadLink() {
        return $this->downloadLink;
    }
}
```

2.2.2 Lettura dei file in PHP

PHP offre numerose funzionalità per leggere da file.

Un modo basilare è attraverso la funzione `file_get_contents()`, che legge l'intero contenuto di un file con una sola operazione come mostrato nel codice 2.3.

Listing 2.3: Esempio di lettura di un file con file_get_contents()

```
$content = file_get_contents("path/to/file.txt");
if ($content === false) {
    throw new Exception("Errore nella lettura del file"
        );
}
```

Invece per leggere file di grandi dimensioni si usa `fopen()` in combinazione con:

- `fgets()` per file generici come nel codice 2.4,
- `fgetcsv()` per i `.csv`.

Listing 2.4: Esempio di lettura di un file con fopen e fgets

```
$handle = fopen("data.csv", "r");
if ($handle === false) {
    throw new Exception("Impossibile aprire il file");
}

while (($line = fgets($handle)) !== false) {
    // Processa ogni riga
}
fclose($handle);
```

Al posto di `fgets()` nel progetto DocRanks si è usato sempre `fgetcsv()` visto che tutti i file erano in formato `.csv`, la funzione è definita come segue nel codice 2.5.

Listing 2.5: Definizione della funzione fgetcsv()

```
function fgetcsv(
    $stream,
    ?int $length = null,
    string $separator = ",",
    string $enclosure = '"',
    string $escape = "\\"
): array|false { }
```

dove la `$length` è la lunghezza di una riga, si può usare 0 per righe di lunghezza non nota, il `$separator` è il carattere di separazione dei campi csv,

`$enclosure` è il carattere utilizzato per racchiudere i campi che contengono caratteri speciali o spazi e infine `$escape` è il carattere utilizzato per inserire il carattere di delimitaizone nel contenuto di un campo. La funzione restituisce `$false` in caso di errore o al raggiungimento dell'End of File (EOF)

2.2.3 Interazione con Server MySQL

L'estensione `MySQLi` permette al PHP di interagire con i database MySQL in modo sicuro ed efficiente attraverso l'uso dei prepared statements, che prevengono le SQL injection e migliorano le performance.

Per stabilire una connessione viene creato un oggetto di tipo `mysqli` a cui vengono passati: nome dell'host, nome utente, password e nome del database come nel codice 2.6.

Listing 2.6: Esempio di connessione a un database MySQL

```
$servername = "localhost";
$username = "root";
$password = "";
$database = "docranks";

$mysqli = new mysqli($servername, $username, $password,
    $database);

if ($mysqli->connect_error) {
    die("Errore di connessione al database");
}

$mysqli->set_charset("utf8");
```

Per interagire con il database invece si usano i prepared statements che pemettono di eseguire query evitando le SQL injection come nel codice 2.7.

Listing 2.7: Esempio di interazione con il database MySQL

```
$stmt = $mysqli->prepare("SELECT nome, cognome FROM
    autori WHERE scopus_id = ?
");
$stmt->bind_param("s", $scopus_id);
$stmt->execute();
$result = $stmt->get_result();

while ($row = $result->fetch_assoc()) {
    echo $row["nome"] . " " . $row["cognome"];
}
$stmt->close();
```

2.2.4 Interazioni API Rest

PHP permette vari approcci per interagire con le API.

`file_get_contents()`: è il modo più semplice per ottenere i dati, infatti permette di salvare in una stringa i dati di un Uniform Resource Locator (url), che risulta molto comodo quando si parla di API REST.

client URL (`cURL`): metodo per API complesse che richiedono autenticazioni con header personalizzati.

Se la risposta delle API è in formato JSON, PHP offre la funzione `json_decode()` per convertire la stringa JSON in un dizionario, che poi andrà gestito con funzioni apposite per ottenere il formato richiesto.

2.3 Ambiente di Sviluppo

Nel caso di DocRanks la maggior parte dello sviluppo è avvenuta tramite XAMPP e successivamente è stato creato un ambiente Docker. Durante tutto lo sviluppo si è usato Git per versionare il software e Github per ospitare il

progetto in remoto. In questa sezione verranno approfondite le sezioni su Apache, MySQL, Docker, Git e Github.

2.3.1 XAMPP

XAMPP (Cross-platform, Apache, MySQL, PHP, Perl) è un paradigma di sviluppo web che collega vari software per creare un unico ambiente di sviluppo completo.

Le sue componenti principali sono:

- Apache: web server che gestisce le richieste HTTP.
- MySQL: DataBase Management System (DBMS) alla base del database.
- PHP: linguaggio di scripting per il backend.

Oltre al paradigma XAMPP, esistiono varie alternative tra cui le più simili non multipiattaforma sono:

- LAMP: specifico per Linux.
- WAMP: specifico per Windows.
- MAMP: specifico per macOS.

Infine ci sono anche altri paradigmi di sviluppo alternativi e moderni come Docker, che sfrutta una architettura conteinerizzata indipendente dal sistema operativo, oppure gli ambienti basati su Node.js che ha una grande diffusione e permette di sviluppare un backend completamente in JavaScript o TypeScript.

2.3.2 Apache

Apache HTTP Server è uno dei web server più utilizzati al mondo, tra i sui vantaggi principali ci sono la sua compatibilità con linguaggi server-side,

come il PHP, e la sua licenza open source molto libera e permissiva che ne hanno permesso la diffusione.

Nel contesto di XAMPP, Apache gestisce le richieste HTTP e restituisce il file al browser in risposta alle richieste web, passando all'interprete PHP lo script corrispondente.

2.3.3 MySQL

MySQL è un sistema di gestione di database relazionale (RDBMS) open source sviluppato da MySQL AB e attualmente mantenuto da Oracle. Il logo di MySQL è mostrato nella Figura 2.2.

MySQL viene utilizzato in XAMPP per la gestione dei dati persistenti, è infatti il linguaggio alla base dello sviluppo del database dei progetti su questo paradigma. I database relazionali non si limitano a contenere le informazioni, ma garantiscono anche che i dati inseriti rispettino alcune regole e restrizioni inserite dal programmatore.

Figura 2.2: Il logo di MySQL, il delfino raffigurato si chiama Sakila, la mascotte ufficiale di MySQL

Per semplificare la creazione del database e la gestione di quest'ultimo sono stati usati:

- DB-Main: un software che permette di creare schemi e diagrammi relazionali da cui poi estrapolare un file `.sql` per creare un database.

- MySQL Workbench: un software che permette di creare database MySQL accettandolo come linguaggio di programmazione.

DB-Main

DB-Main è uno strumento di ingegneria del software assistita da un computer (CASE) gratuito che offre varie funzionalità fondamentali alla progettazione di database relazionali, infatti permette di:

- Creare diagrammi entità-relazione (ER), con cui il programmatore crea un modello concettuale del database.
- Trasformare il diagramma ER in un diagramma relazionale, che rappresenta il modello finale del database con le tabelle che effettivamente devono esservi inserite, i loro attributi e le relazioni tra esse.
- Esportare il diagramma relazionale in file `.sql` compatibili con vari DBMS tra cui MySQL.

MySQL Workbench

MySQL Workbench è un ambiente di sviluppo integrato (IDE) per MySQL sviluppato dalla stessa Oracle.

Permette di progettare, implementare e gestire i database MySQL in un singolo ambiente di sviluppo. Si distingue da altri software analoghi in particolare per la sua funzionalità di visualizzazione del design del database, che permette di risalire allo schema relazionale di quest'ultimo.

Nel contesto di progetto è stato utilizzato in particolare per:

- Creare il database progettato in DB-Main.
- Visualizzare i dati delle tabelle.
- Testare le query SQL.
- Modificare le tabelle attraverso la sua funzionalità di visualizzazione del design del database.

2.3.4 Docker

Docker è una piattaforma che permette agli sviluppatori di creare, condividere ed eseguire container. Nella programmazione web è utilizzato per creare ambienti di sviluppo riproducibili su ogni macchina, permettendo ai vari sviluppatori di lavorare in condizioni identiche [16].

Nel progetto DocRanks Docker è stato sostituito a XAMPP per favorire l'utilizzo dell'applicazione su ogni sistema in ambienti di lavoro uguali, quindi è stato creato un file Docker Compose che permette l'uso dell'app in locale, mantenendo la compatibilità con MySQL e PHP.

Storia

Docker è un progetto creato da Solomon Hykes nel 2010 con il nome di dotCloud (Fig. 2.3) che nasce con lo scopo di semplificare lo sviluppo di applicazioni cloud. Nel 2013 dotCloud diventa Docker che si sviluppa come piattaforma per gestire e creare container. Negli anni acquisisce sempre più popolarità per la sua semplicità d'uso e la sua capacità di risolvere il problema del "works on my machine" grazie alle sue caratteristiche di riproducibilità degli ambienti di sviluppo [17].

Figura 2.3: Il logo originale di dotCloud, il progetto da cui è nato Docker

Container

Un container è un'unità di un software che impacchetta il codice e tutte le sue dipendenze in modo che l'applicazione possa essere eseguita rapidamente in vari ambienti informatici [16].

In un'applicazione web si usa solitamente un container per ogni servizio facendoli lavorare insieme. Per farlo è possibile sfruttare il comando `docker compose`, che permette di gestire applicazioni multi-container crendoli secondo una configurazione definita dal programmatore all'interno di un file denominato `docker-compose.yaml`, inoltre è possibile creare immagini personalizzate all'interno di un file `Dockerfile`.

Dockerfile

un Dockerfile è un file testuale con varie istruzioni che permettono di creare automaticamente un'immagine Docker. Ogni istruzione indica un layer che viene aggiunto a partire da un'immagine base. Un esempio di Dockerfile è scritto nel codice 2.8.

Le istruzioni più comuni in un Dockerfile sono:

- `FROM`: specifica l'immagine di base da cui partire.
- `WORKDIR`: imposta la cartella di lavoro all'interno del container.
- `COPY`: copia file o cartella dal contesto di lavoro al container.
- `RUN`: esegue comandi all'interno del container durante la build.
- `EXPOSE`: espone una porta specifica del container.
- `CMD`: specifica il comando da eseguire all'avvio del container.

Listing 2.8: Esempio di Dockerfile per un'applicazione Python

```
FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .
RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["python", "main.py"]
```

Docker Compose

Docker Compose è uno stumento utile a gestire applicazioni multicontainer. Basa il suo funzionamento su un file `.yaml`, di cui si ha un esempio nel codice 2.9, in cui vengono definite delle regole e dipendenze per i vari container che verranno creati. Ogni container è definito come un servizio, `sevices`, e per ogni servizio è possibile definire varie opzioni che ne influenzano il comportamento, le più importanti per un app web sono:

- `image`: l'immagine da utilizzare.
- `build`: il percorso del Dockerfile per costruire l'immagine.
- `ports`: le porte da esporre.
- `volumes`: i volumi da montare in caso in cui si voglia che i dati siano persistenti.
- `environment`: le variabili d'ambiente da impostare.
- `depends_on`: le dipendenze tra i servizi.
- `restart`: la politica di riavvio del container.

Listing 2.9: Esempio di file docker-compose.yaml per un'applicazione web

```
services:
  db:
    image: mysql
    environment:
      MYSQL_ALLOW_EMPTY_PASSWORD: true
    ports:
      - 3306:3306
    volumes:
      - data_db:/var/lib/mysql
      - init.sql:/docker-entrypoint-initdb.d/init.sql:
          ro

  server:
    build: .
    ports:
      - 8080:80
    depends_on:
      - db
    environment:
      - DB_HOST=db

volumes:
  data_db:
```

2.3.5 Git

Git è uno dei software di controllo versione distribuito più usati al mondo. È stato sviluppato da Linus Torvalds nel 2005 per gestire lo sviluppo del kernel Linux. Negli anni ha acquisito grande popolarità grazie alla sua grande efficienza che lo metteva in risalto rispetto alla concorrenza.

Git, al contrario di altri software di controllo versione come SVN, memorizza istanatanee del progetto a ogni commit piuttosto che le differenze tra le versioni, migliorando le prestazioni. Ogni file di Git inizialmente è parte della cartella di lavoro, dopo il comando `git add` il file viene spostato nell'area di staging, dove rimane fino al commit, che lo sposta nella cronologia del progetto (repository). Durante questi passaggi i file vengono compressi e salvati dopo la decuplicazione md5, che permette di risparmiare spazio.

Git LFS

Per la sua natura per cui a ogni commit viene salvata un'istantanea del progetto, Git non è adatto a gestire file di grandi dimensioni in quanto modifiche a questi file causano grandi aumenti di spazio occupato non necessari.

Non solo, ma quando un progetto viene clonato, Git scarica l'intera cronologia del progetto che in caso di file grandi che hanno subito molte modifiche tra i commit porta a un notevole aumento dello spazio occupato e del tempo di download.

Per risolvere questi problemi è stato sviluppato Git Large File Storage (Git LFS), un'estensione di Git che permette di gestire file di grandi dimensioni in modo più efficiente. Git LFS sostituisce i file scelti dal programmatore con dei puntatori, file di testo che contengono informazioni su dove si trovano i file reali. I file scelti vengono poi memorizzati in un server separato, riducendo lo spazio occupato nella cronologia del progetto e migliorando le prestazioni. È possibile rimuovere i file seguiti da Git LFS dalla cache locale di progetto nel caso in cui non siano più necessari, riducendo lo spazio occupato localmente.

Quando un progetto Git LFS viene clonato viene scaricata l'intera storia dei file seguiti da Git LFS attraverso l'uso di puntatori e i file dell'ultimo commit sono gli unici che vengono effettivamente scaricati dal server con Git LFS. In questo modo il programmatore può lavorare come se tutti i file

fossero normali file Git, ma in realtà i file di grandi dimensioni sono gestiti in modo più efficiente.

2.3.6 GitHub

GitHub è una piattaforma compatibile con Git e Git LFS che ospita vari progetti open source e privati. È stata fondata nel 2008 e acquisita da Microsoft nel 2018.

Il progetto di DocRanks è stato pubblicato su GitHub rendedolo accessibile per il download e la collaborazione da parte degli utenti.

Capitolo 3

Progetto DocRanks

DocRanks è un'applicazione web con lo scopo di semplificare la raccolta dei dati sugli autori e le loro pubblicazioni accademiche. Integra dati provenienti da diverse fonti autorevoli e permette agli utenti di inserirne alcuni al fine di fornire una sola vista unificata dei risultati delle ricerche. Il sistema raccoglie e presenta informazioni sugli autori in particolare nell'ambito informatico, sulle loro pubblicazioni e sulle metriche che le riguardano, combinando i dati ottenuti da Scopus, DBLP, CORE e SCImago.

La piattaforma si pone l'obbiettivo di fornire una panoramica completa delle pubblicazioni degli autori attraverso l'integrazione delle informazioni provenienti da vari siti offrendo una singola interfaccia web semplice e veloce da usare che permette di ridurre il tempo necessario alla ricerca delle informazioni attraverso l'uso di un solo sito.

La piattaforma è utile a ricercatori e istituzioni che necessitano di valutare l'impatto scientifico e la qualità delle pubblicazioni e dei loro autori.

Il sito è implemtato secondo un'architettura modulare con lo scopo di facilitare la manutenzione e l'aumento delle funzionalità, invece l'interfaccia è progettata per essere molto intuitiva permettendo agli utenti di impararne immediatamente il funzionamento.

3.1 Progettazione del Database

Il Database è un componente fondamentale del progetto, esso ha infatti lo scopo di archiviare i dati in modo organizzato per poterli mostrare agli utenti che usano il sito. Nel caso di DocRanks, che deve raccogliere i dati da diverse fonti, collegarli tra di loro e mostrarli all'utente, il Database ha un ruolo fondamentale.

Il Database deve memorizzare:

- i vari autori, sui quali è incentrato il sistema di DocRanks,
- tutte le pubblicazioni associate agli autori che riesce a reperire insieme a tutte le metriche,
- tutte le riviste su cui possono essere pubblicati articoli,
- tutte le conferenze in cui un autore può presentare un atto.

E infine deve collegare ogni sua parte in modo da mostrare i dati in un'interfaccia.

Creazione del Database

Per creare il database relazionale si sono seguiti i seguenti passaggi:

1. Progettazione Concettuale: creazione dello schema Entità Relazione (ER) con i rispettivi attributi.
2. Progettazione Logica: traduzione dello schema ER in tabelle relazionali.
3. Normalizzazione: applicazione delle forme normali.
4. Progettazione fisica: trasformazione in database.

3.1.1 Progettazione Concettuale

La progettazione concettuale è alla base dello sviluppo di un database relazionale, in questa fase vengono rappresentati i dati e le loro relazioni senza preoccuparsi di come verranno implementati fisicamente. A questo fine si fa uso di uno schema ER che permette di identificare:

- le entità, ovvero gli oggetti del dominio come Autore o Rivista;
- i loro attributi, ad esempio per un Autore possono essere nome e cognome;
- le relazioni che collegano le entità.

Nel caso di DocRanks ci sono numerosi dati e entità da unire nello schema ER, le principali sono:

AUTORI: Questa entità rappresenta i ricercatori presenti nel sistema, l'autore pubblica varie opere come articoli o atti di convegno assieme ad altri autori, è tipicamente associato a una serie di statistiche che possono essere ricavate da DBLP. Ogni autore è identificato univocamente dallo Scopus ID.

PUBBLICAZIONI: La pubblicazione è l'opera alla base del sito, esistono vari tipi di pubblicazione tra le quali le più importanti sono gli **ATTI DI CONVEGNO** e gli **ARTICOLI** che rappresentano rispettivamente gli atti di conferenza pubblicati nelle conferenze e gli articoli delle riviste. Ogni pubblicazione ha un titolo, un anno in cui viene pubblicata e un Digital Object Identifier (DOI) che la identifica. Il DOI è presente sia su Scopus che su DBLP ed è essenziale a collegare le pubblicazioni tra i due siti.

RIVISTE: Ogni articolo viene pubblicato su una rivista e uno dei compiti principali di DocRanks è collegare l'articolo alle metriche della rivista su cui viene pubblicato, ogni rivista appartiene a varie aree disciplinari e pubblica articoli di diverse categorie di quelle aree. Alcuni dati per

valutare una rivista sono l'SJR, ottenuto da SCImago, da cui derivano anche i quartili, e lo SNIP e il CiteScore, ottenuti invece da Scopus.

CONFERENZE: Ogni atto di convegno è parte di una conferenza, i dati delle conferenze sono salvati su CORE, uno dei compiti di DocRanks è associare un atto di convegno alla corrispettiva conferenza su CORE, ogni conferenza è identificata da un acronimo e viene valutata secondo il sistema di CORE.

Lo schema ER di DocRanks 3.1 è sviluppato in modo da poter associare a ogni autore le sue pubblicazioni, da cui poi è possibile ottenere i dati sulle riviste e conferenze a esse associate.

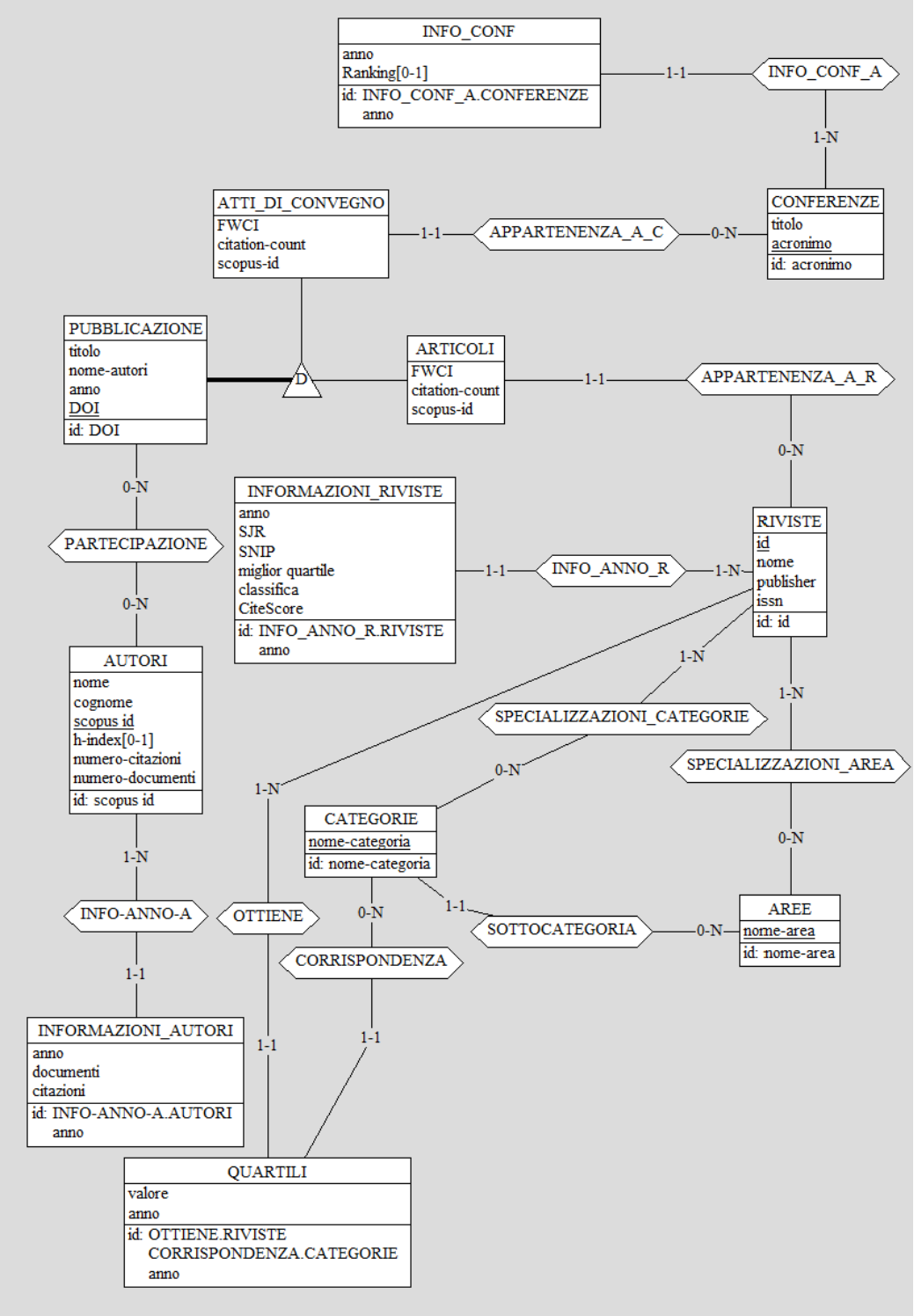


Figura 3.1: schema ER del database DocRanks

3.1.2 Progettazione Logica

La progettazione logica è la fase di sviluppo di un database in cui lo schema ER viene trasformato in uno schema relazionale che può poi essere trasformato in un database. In questo passaggio vanno definiti in modo chiaro tabelle, attributi, chiavi primarie e foreign key che identificano i dati.

Per trasformare lo schema ER in schema relazionale sono state usate le funzionalità di DB main dopo aver applicato le seguenti modifiche:

- collasso verso il basso dell'entità `PUBBLICAZIONE`: Le gerarchie non possono essere presenti in uno schema relazionale, visto che la gerarchia era incompleta e disgiunta si è deciso di creare la tabella `OTHERS` in cui vengono messe le pubblicazioni che non sono né atti di convegno né articoli e poi sono stati copiati tutti gli attributi della tabella `PUBBLICAZIONE` nei figli.

- trasformazione in tabelle delle relazioni $N : M$ (molti a molti).

- inserimento delle chiavi primarie come foreign key nelle relazioni $1 : N$ (uno a molti).

Nello schema ER non erano presenti attributi multipli, né relazioni $1 : 1$ (uno ad uno).

Normalizzazione

Lo schema creato a questo punto è in terza forma normale quindi rispetta le seguenti regole:

1. Ogni attributo è atomico e indivisibile,

2. Ogni attributo non chiave dipende dall'intera chiave primaria,

3. Nessun attributo non chiave dipende da un attributo non chiave.

3.1.3 Progettazione Fisica

La progettazione fisica trasforma uno schema relazionale in un database con l'ausilio di un DBMS, in questo caso MySQL. In questa fase viene definita la dimensione degli attributi, eventuali attributi NULL e ridondanze allo scopo di favorire l'implementazione del database. In DocRanks sono stati seguiti i seguenti passaggi attraverso DB-Main:

- Tipizzazione degli attributi del database e definizione delle dimensioni,
- Implementazione di attributi nullable,
- Inserimento di attributi ridondanti.

A questo punto si è fatto uso delle funzioni di DB-Main per generare il codice MySQL.
Lo schema che rappresenta il risultato finale è rappresentato nella Figura 3.2.

3.1.4 Descrizione

In questa sezione vengono descritte tutte le tabelle del database, le loro chiavi primarie (PK), chiavi esterne (FK), i loro attributi e come vengono ricavati.

AUTORI

Contiene le informazioni principali degli autori.

- **scopus_id** (PK): Identificativo univoco Scopus dell'autore, viene inserito dall'utente e usato per cercare le informazioni su Scopus.
- **nome e cognome**: Nome e Cognome dell'autore, inseriti dall'utente, necessari per le ricerche da DBLP.
- **h_index**: Indice di Hirsch dell'autore, viene inserito manualmente.
- **numero_citazioni**: Totale citazioni ricevute da Scopus.
- **numero_documenti**: Totale pubblicazioni su Scopus.

INFORMAZIONI_AUTORI

Statistiche annuali per ogni autore ottenute da Scopus.

- **scopus_id** (PK, FK): Riferimento all'autore.
- **anno** (PK): Anno di riferimento.
- **documenti**: Numero documenti pubblicati nell'anno.
- **citazioni**: Citazioni ricevute nell'anno.

Pubblicazioni

Tutte le pubblicazioni (articoli su riviste, atti di convegno e altre tipologie) condividono alcuni attributi comuni:

- **DOI** (PK): Identificativo univoco della pubblicazione.
- **titolo**: Titolo della pubblicazione.
- **anno**: Anno di pubblicazione.
- **nome_autori**: Lista completa degli autori. Questa voce non è ridondante perché non è garantito che su DocRanks siano presenti tutti gli autori che partecipano alla pubblicazione.

Invece non sono presenti in `OTHERS`, ma sono presenti nelle altre pubblicazioni:

- **numero_autori**: Numero totale di autori. Questa informazione è ridondante, poiché può essere derivata da **nome_autori**.
- **FWCI**: Field-Weighted Citation Impact, questo valore non può essere ottenuto con le api base di Scopus, dunque deve essere inserito manualmente, altrimenti è `NULL`.

- **EFWCI**: Expected Field-Weighted Citation Impact, questa voce è ridondante perché è il rapporto del numero di citazioni ottenute per un valore costante.
- **citation_count**: Numero di citazioni, viene preso da Scopus, come tutti i dati ricavati da Scopus può avere valore `NULL` per ridurre la possibilità di errori del sistema in caso di errori legati al server di Scopus.
- **scopus_id**: Identificativo Scopus della pubblicazione, anch'esso è ricavato da Scopus.

ARTICOLI

Pubblicazioni su riviste con dati integrati da Scopus e DBLP.
Oltre agli attributi comuni hanno anche:

- **dblpRivista**: Nome rivista come appare in DBLP, può sembrare ridondante, ma da vari test è emerso che il nome della rivista su DBLP e su SCImago talvolta differisce leggermente.
- **id** (FK): Riferimento alla rivista nel database, presente sia su Scopus che su SCImago, può essere cambiato dall'utente che può correggere eventuali errori del sistema.

ATTI_DI_CONVEGNO

Pubblicazioni in conferenze con dati integrati da Scopus e DBLP.
Oltre agli attributi comuni hanno anche:

- **acronimo_dblp**: Nome dell'acronimo della conferenza come appare in DBLP, non è ridondante perché talvolta l'acronimo di DBLP e di CORE differiscono leggermente.
- **acronimo** (FK): Riferimento alla conferenza CORE, può essere modificato dall'utente che può correggere eventuali errori.

OTHERS

Altre tipologie di pubblicazioni ottenute da DBLP.
Oltre agli attributi comuni, hanno anche:

- **dblp_venue**: Origine della pubblicazione secondo DBLP.
- **tipo**: Tipologia di pubblicazione.

RIVISTE

Informazioni sulle riviste ottenute da SCImago.

- **id** (PK): Il SourceID della rivista, è molto utile poiché è presente sia su Scopus che su SCImago, quindi garantisce un collegamento molto affidabile tra articoli e riviste.
- **nome**: Nome completo della rivista.
- **publisher**: Casa editrice.
- **issn**: Codice seriale della rivista.

INFORMAZIONI_RIVISTE

Metriche annuali delle riviste da SCImago e Scopus.

- **id** (PK, FK): Source ID della rivista in questione.
- **anno** (PK): Anno di riferimento.
- **SJR**: Metrica principale di SCImago.
- **SNIP**: SNIP di Scopus, viene ottenuto attraverso dei file csv, tuttavia può essere `NULL` poiché non sono stati scaricati tutti i file sulle riviste presenti su Scopus.
- **miglior_quartile**: Migliore quartile della rivista in quell'anno.

- **classifica**: Posizione nel ranking SCImago di quell'anno.
- **CiteScore**: Metrica di Scopus che può assumere valore `NULL` per lo stesso motivo dello SNIP.

AREE

Aree di SCImago per la classificazione delle riviste, il loro solo attributo e PK è **nome_area**.

CATEGORIE

Categorie di ricerca specifiche all'interno delle aree.

- **nome_categoria** (PK): Nome della categoria.
- **nome_area** (FK): Area di appartenenza.

QUARTILI

Quartili delle riviste di SCImago per categoria e anno.

- **nome_categoria** (PK, FK): Categoria di valutazione.
- **id** (PK, FK): Rivista valutata.
- **anno** (PK): Anno di riferimento.
- **valore**: Valore del quartile.

CONFERENZE

Informazioni sulle conferenze ottenute da CORE.

- **acronimo** (PK): Acronimo della conferenza di CORE.
- **titolo**: Nome completo della conferenza.

INFO_CONF

Ranking annuali delle conferenze secondo CORE.

- **acronimo** (PK, FK): Riferimento alla conferenza.
- **anno** (PK): Anno di riferimento.
- **valore** (FK): Ranking CORE.

RANKING_1

Valori possibili per i ranking delle conferenze CORE, il loro unico attributo e PK è **valore**.

Tabelle di Relazione

Le seguenti tabelle sono semplici relazioni molti a molti che collegano tra loro le tabelle precedenti:

- **REDAZIONE**: Collega autori e articoli.
- **PARTECIPAZIONE**: Collega autori e atti di convegno.
- **PUBBLICAZIONE_ALTRO**: Collega autori e le altre pubblicazioni.
- **SPECIALIZZAZIONI_AREA**: Collega riviste e aree di ricerca.
- **SPECIALIZZAZIONI_CATEGORIE**: Collega riviste e categorie di ricerca.

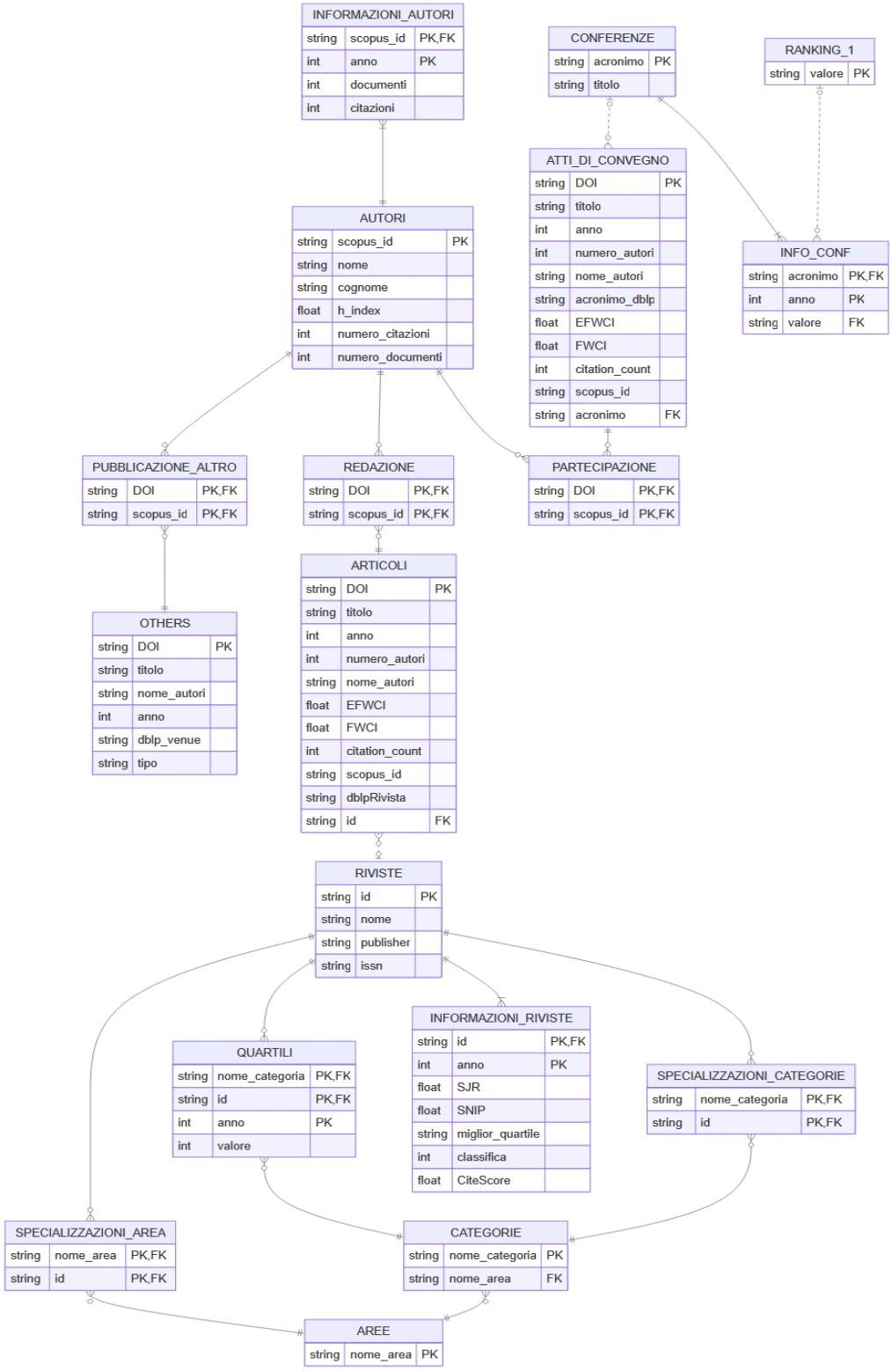


Figura 3.2: schema finale del database DocRanks

3.2 Integrazione con le API esterne

DocRanks si basa sulla raccolta dei dati presenti su Scopus, DBLP, CORE e SCImago; la maggior parte di questi dati è stata ottenuta attraverso dei file csv da cui vengono estratte le informazioni utili al sistema, ma una parte delle informazioni viene estratta e salvata sul sistema attraverso l'uso di API, in particolare vengono usate le API di Scopus e DBLP.

3.2.1 API Scopus

Le API di Scopus sono servizi "Representational State Transfer" (REST) quindi:

- Le entità possono essere trovate a degli URL univoci.
- Sono compatibili con HTTP.
- Ogni chiamata contiene tutte le informazione necessarie, due richieste uguali portano allo stesso risultato.
- Le risposte sono in formati standard come JSON.
- Ogni risposta alle richieste rispetta sempre le stesse regole.

Nel caso delle API di Scopus per accedere ai dati è necessaria una chiave API, ottenibile attraverso la registrazione sul portale di Elsevier. Per salvare questa chiave nel progetto è presente un file `.env` ignorato da git con all'interno la chiave che viene resa accessibile a tutti i file di progetto attraverso la funzione `putenv()` di PHP.

Nel caso di DocRanks viene usata la Scopus Search API, che permette di trovare numerosi astratti del portale. I dati che possono essere ottenuti con la key non presenti su DBLP in particolare sono:

- `SourceID`: l'identificativo della rivista su cui viene pubblicato un articolo, comune anche a SCImago.

- `citedby-count`: il numero di citazioni dell'articolo.
- `eid`: l'id che Scopus assegna all'articolo.

L'URL da creare per accedere a questi dati è stata sviluppata secondo la documentazione ufficiale di Elsevier. Il sistema usa l'endpoit base https://api.elsevier.com/content/search/scopus e poi invia richieste `GET` per recuperare le informazioni relative a un autore. Per identificare un autore viene usato il parametro `query` con valore `AU-ID`, ovvero l'id di Scopus dell'autore, mentre il parametro `field` limita i dati restituiti a solo quelli che sono stati inseriti nel Database, ovvero: il doi per permettere il collegamento con DBLP, il citedby-count, il source-id e l'eid che identifica l'articolo. Sono stati aggiunti altri attributi quali start e count per limitare il numero di richieste a 25 per batch per ridurre la possibilità di errori e sort per ordinare per anno le pubblicazioni.

L'URL che ne risulta è `https://api.elsevier.com/content/search/scopus?query=AU-ID(url)&start=0&count=25&sort=pubyear&field=doi,citedby-count,source-id,eid`.

Nel contesto del progetto DocRanks la classe `ScopusAuthorData` rappresenta il cuore dell'integrazione con Scopus, essa crea l'URL sopracitata e le associa gli headers necessari al funzionamento di scopus, che contengono l'api key, poi preleva i dati da Scopus su un autore e li salva in una struttura dati interna che viene usata nella classe `AuthorProcessor` che poi si occupa di salvare i dati con i vari `repository`.

3.2.2 API DBLP

Le API di DBLP sono ach'esse REST. DBLP offre informazioni in particolare riguardo al settore informatico, nel contesto di DocRanks vengono usate per raccogliere la maggior parte delle informazioni sulle pubblicazioni. Per ottenere le informazioni in questo caso viene usato il parametro `q` associato a Nome e Cognome dell'autore, che risulta uno dei sistemi più affidabili per accedere alle informazioni su di esso.

Da DBLP vengono ottenuti i seguenti dati sulle pubblicazioni:

- titolo della pubblicazione;
- autori della pubblicazione;
- origine della pubblicazione, che spesso rappresenta l'acronimo della conferenza o il titolo abbreviato della rivista;
- anno di pubblicazione;
- tipo della pubblicazione;
- DOI della pubblicazione.

Nel contesto di DocRanks questo viene svolto attraverso una funzione che esegue `GET` all'url:
`https://dblp.org/search/publ/api?q=NomeCognome&format=json`.
E poi restituisce una lista di pubblicazioni in cui ognuna è un dizionario che ha per chiave il nome del dato cercato e per valore il dato in sè. I dati ricevuti vengono gestiti dalla classe `AuthorProcessor` che li salva nel Database chiamando i metodi delle `repository`.

3.3 Sviluppo di DocRanks

DocRanks è una applicazione web scritta in HTML, CSS e PHP. Il codice ha lo scopo di interagire con il database e mostrare in un'interfaccia grafica i risultati.

3.3.1 Architettura del Sistema

Il sistema è organizzato secondo un'architettura che separa chiaramente la presentazione del sito dalla logica di interazione con il database.

Nella cartella più esterna sono contenuti i file:

- .env e il corrispettivo .env.example: che contiene la chiave delle API Scopus.
- Il file example.json: che mostra un esempio di come inserire degli autori tramite file JSON.
- I file Dockerfile e docker-compose.yaml: che contengono le configurazioni di Docker per usare il software in un ambiente containerizzato.
- I file alla radice che mostrano un'interfaccia grafica: index.php, reset_and_init.php e update_database.php. Questi file permettono di andare alla home page del sito e inserire i dati di: SCImago, CORE e alcuni dei dati di Scopus nel database in maniera automatizzata.
- Le varie sottocartelle e file di configurazione di git e docker.

uploads

La cartella `uploads` è una cartella che contiene file di grandi dimensioni ed è versionata da git lfs. Al suo interno ha tre sottocartelle:

- `core`, che contiene tutto il database di CORE in formato csv;
- `scimagojr`, che contiene tutti i dati di SCImago sulle riviste a partire dal 2014 in formato csv nell'area informatica;
- `scopus`, che contiene i dati sulle riviste informatiche più importanti a partire dal 2022.

I dati nella cartella `uploads` vengono caricati nel database quando viene cliccato il tasto "`Resetta e Inizializza il Sito`".

sql

La cartella `sql` contiene il file `docranks.sql` che permette di inizializzare il database in MySQL e contiene un file `db.md` con una rapida descrizione del database.

api

La cartella `api` contiene i due file `ScopusAuthorDataData.php` e `dblp_integrations.php` che permettono di interagire rispettivamente con le API Scopus e DBLP.

database

La cartella `db` contiene tutti i file e sottocartelle che interagiscono direttamente con il database con delle `query` MySQL quali:

- `connection.php`: il file che instaura la connessione al database.
- la sottocartella `importers`, che contiene i file con le `query` per inizializzare il database:
  - `CategoriesManager.php`: classe che inserisce nel database una serie di aree e categorie prestabilite.
  - `CoreImporter.php`: classe che inserisce nel database i dati nella cartella `uploads/core`.
  - `ScimagoImporter.php`: classe che inserisce nel database i dati nella cartella `uploads/scimagojr`.
  - `ScopusImporter`: classe che importa i dati nella cartella `uploads/scopus`.
- la sottocartella `migrations` che permette di inizializzare il database e di svuotarlo completamente.
- la sottocartella `repository` con tutti i file per interagire con le tabelle:
  - `PubblicationRepository.php` contiene i metodi comuni relativi all'inserimento delle principali pubblicazioni. Viene incapsulato in `PaperRepository.php` e `ArticleRepository.php`, che permettorno di gestire rispettivamente gli atti di convegno e gli articoli.

- `Others.php` si occupa di inserire e mostrare le pubblicazioni che non rientrano nelle categorie precedentemente citate.
- `ConferenceRepository.php` tenta di trovare l'acronimo CORE della conferenza a partire dall'origine di DBLP.
- `JournalRepository.php` Mostra i dati sulle riviste nel database e le collega agli articoli.
- `AuthorRepository.php` gestisce un autore con un certo `scopus id`.
- `AuthorsRepository.php` gestisce le funzioni che riguardano tutti gli autori presenti nel database.

home

Nella cartella `home` sono incluse le pagine del sito, è presente un file `index.php` che rappresenta la pagina principale da cui si può esplorare il sito in sè. Contiene una sottocartella autore con i file:

- `index.php` che permette di cercare un autore e mostrarne le informazioni generali.
- `aggiungi.php` che permette di aggiungere nuovi autori tramite barra di ricerca o file JSON.
- `conference_papers.php`, `journal_articles.php` e `other_publications.php` che mostrano rispettivamente tutti gli atti di conferenza, articoli e altre pubblicazioni di un autore con tutte le metriche delle riviste e delle conferenze nell'anno di pubblicazione associate.

includes e services

La cartella `includes` contiene funzioni e porzioni di codice usate in varie sezioni del progetto.

Invece la cartella `serivices` contiene il file `AuthorProcessor.php`, che collega api e database. Dato nome cognome e scopus id di un autore ne ricerca i dati con le API di Scopus e DBLP. Infine ne inserisce o aggiorna le informazioni nel database.

3.3.2 Funzionalità e Interfaccia

DocRanks è sviluppato per garantire all'utente la massima facilità di utilizzo, le ricerche sono veloci da effettuare e i risultati facili da interpretare, ogni pagina può essere facilmente navigata sia da pc che da dispositivo mobile 3.3b.

La navigazione è organizzata secondo una struttura gerarchica che mette la home page come pagina centrale da cui accedere alle funzioni del sito. La sezione degli autori garantisce funzioni di ricerca, visualizzazione e aggiunta di nuovi autori. Dal profilo di un autore è infine possibile accedere a viste dettagliate sulle sue singole pubblicazioni divise per tipo.

In ogni pagina è infine presente una barra di navigazione che semplifica ulteriormente la fruizione del sito.

Home Page

(a) Desktop (b) Mobile

Figura 3.3: La home page di DocRanks su desktop e mobile

La Home Page 3.3 ha una breve introduzione al sito e una descrizione delle funzionalità di cui dispone l'applicazione, permette di accedere alle pagine di ricerca degli autori presenti o di aggiungerne di nuovi.

Ricerca Autori

La pagina degli autori richiede lo Scopus ID come parametro per all'interno di una barra di ricerca per individuare gli autori. Inoltre presenta una tabella con tutti gli autori presenti nel sito. Per cercare un autore si può anche cliccare sull'apposito "Visualizza Profilo" nella tabella 3.4.

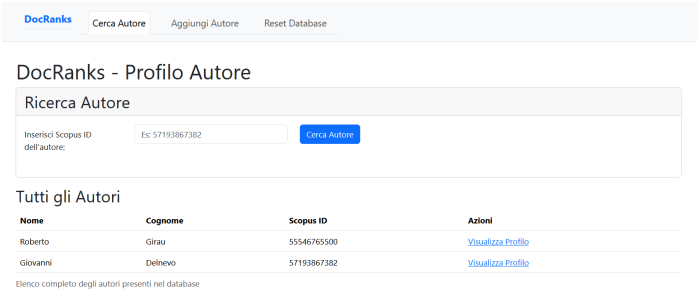


Figura 3.4: La pagina di ricerca autori

Quando un autore viene cercato avviene una richiesta con "GET" e viene aggiornata la pagina per mostrare alcune informazioni sull'autore in questione, da qui è anche possibile accedere ai dettagli sulle sue pubblicazioni 3.5.

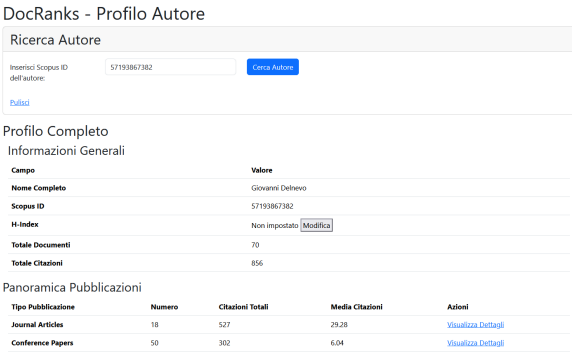


Figura 3.5: La pagina di ricerca con i dettagli di un autore

Aggiunta Autori

In questa pagina 3.6 è possibile inserire autori non presenti nel database o aggiornare i dati di quelli presenti. Per aggiungere nuovi autori sono necessari lo Scopus ID e poi si può scegliere se usare il Nome e il Cognome di un autore o se preferire il PID di DBLP.

Come alternativa si possono importare numerosi autori assieme tramite file JSON.

Dopo che un autore viene inserito si può cliccare l'apposito link per essere reinderizzati alla pagina di ricerca che ne mostra i dettagli.

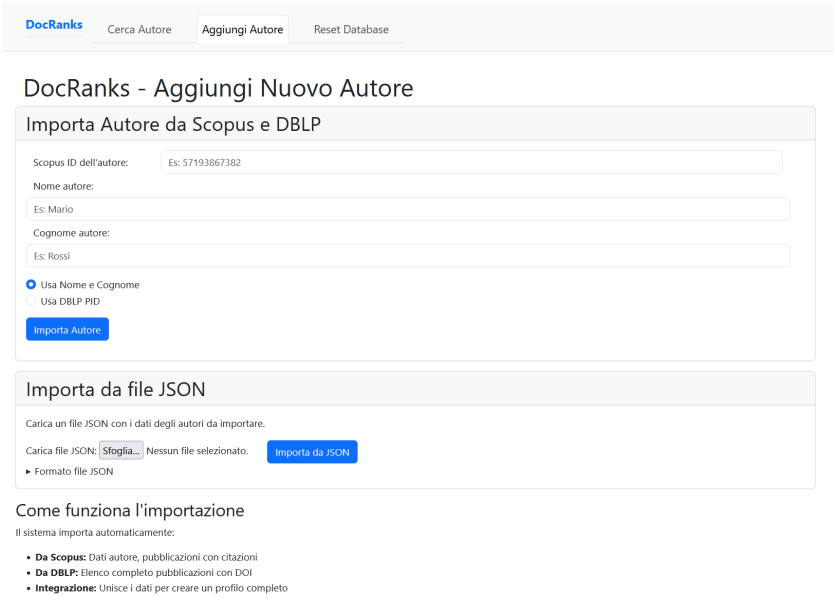
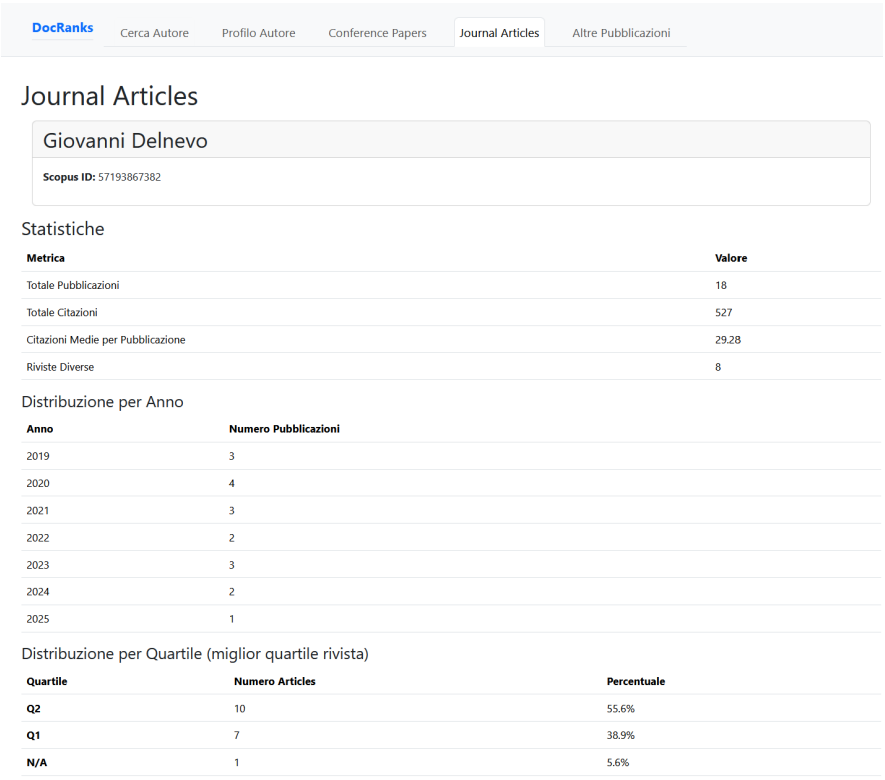


Figura 3.6: La pagina per aggiungere nuovi autori

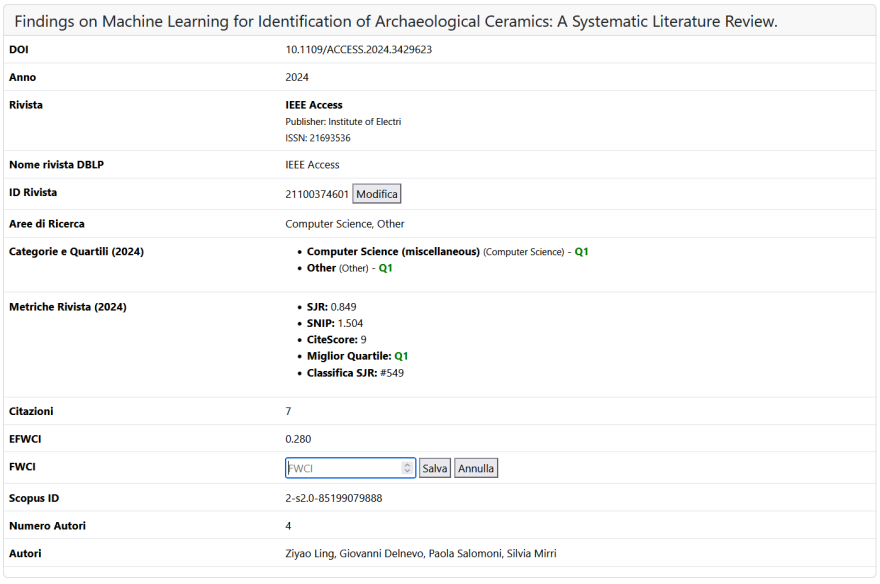
Pubblicazioni

Le pagine sulle pubblicazioni 3.7 sono formate da varie "Card" di Bootstrap con all'interno una serie di dettagli e metriche sulle pubblicazioni dell'autore e sulla loro origine 3.7b.

In cima alla pagina vengono mostrate delle tabelle di resoconto 3.7a e in fondo viene data la possibilità di navigare altre sezioni del sito.



(a) Pagina articoli



(b) Esempio di articolo

Figura 3.7: Le pagine delle pubblicazioni di un autore

Conclusioni

In questa tesi è stato esposto il problema della frammentazione delle metriche di valutazione bibliometrica ed è stato descritto il processo di sviluppo di un'applicazione web, DocRanks, che rappresenta una possibile soluzione.

L'obiettivo di DocRanks era quello di fornire una piattaforma unica che raccogliesse i dati provenienti da Scopus, DBLP, CORE e SCImago. In questo modo l'utente ha a disposizione un'unica interfaccia coerente con se stessa e non deve più navigare su numerosi siti eterogenei. Questo porta a un notevole risparmio di tempo, in particolare nel caso in cui si ricerchino informazioni sulle pubblicazioni di un singolo autore, per cui vengono associate tutte le ricerche alle metriche della rivista e della conferenza nell'anno corretto, riducendo anche le possibilità di errore.

Ovviamente anche DocRanks presenta delle limitazioni infatti:

- spesso non riesce ad associare atto di convegno e conferenza corrispettiva su CORE da DBLP,
- non riesce a reperire alcune metriche proprietarie come il Field-Weight-Citation Impact che prova a ricalcolare internamente in modo approssimativo
- le sue informazioni sono limitate all'ambito informatico, come DBLP e CORE.

Tuttavia, anche con questi limiti, DocRanks è un miglioramento significativo rispetto alla situazione attuale in cui un utente deve consultare e correlare manualmente informazioni da diverse piattaforme.

In futuro, per migliorare ulteriormente il progetto, si potrebbe mettere online l'applicazione e integrare grafici e altre funzionalità di ricerca al progetto. Rendendo così il progetto ancora più interessante e accessibile un pubblico più ampio.

Bibliografia

[1] Armen Yuri Gasparyan, Lilit Ayvazyan, and George D. Kitas. Multidisciplinary bibliographic databases. *Journal of Korean Medical Science*, 28(9):1270, 2013. ISSN 1598-6357. doi: 10.3346/jkms.2013.28.9.1270. URL http://dx.doi.org/10.3346/jkms.2013.28.9.1270.

[2] Raminta Pranckutė. Web of science (wos) and scopus: The titans of bibliographic information in today's academic world. *Publications*, 9(1): 12, March 2021. ISSN 2304-6775. doi: 10.3390/publications9010012. URL http://dx.doi.org/10.3390/publications9010012.

[3] Junwen Zhu and Weishu Liu. A tale of two databases: the use of web of science and scopus in academic papers. *Scientometrics*, 123(1):321–335, February 2020. ISSN 1588-2861. doi: 10.1007/s11192-020-03387-8. URL http://dx.doi.org/10.1007/s11192-020-03387-8.

[4] Kai Li, Jason Rollins, and Erjia Yan. Web of science use in published research and review papers 1997–2017: a selective, dynamic, cross-domain, content-based analysis. *Scientometrics*, 115(1):1–20, December 2017. ISSN 1588-2861. doi: 10.1007/s11192-017-2622-5. URL http://dx.doi.org/10.1007/s11192-017-2622-5.

[5] Jeroen Baas, Michiel Schotten, Andrew Plume, Grégoire Côté, and Reza Karimi. Scopus as a curated, high-quality bibliometric data source for academic research in quantitative science studies. *Quantitative Science Studies*, 1(1):377–386, February 2020. ISSN 2641-3337. doi: 10.1162/qss_a_00019. URL http://dx.doi.org/10.1162/qss_a_00019.

[6] Anne-Wil Harzing and Satu Alakangas. Google scholar, scopus and the web of science: a longitudinal and cross-disciplinary comparison. *Scientometrics*, 106(2):787–804, November 2015. ISSN 1588-2861. doi: 10.1007/s11192-015-1798-9. URL http://dx.doi.org/10.1007/s11192-015-1798-9.

[7] Abhaya V. Kulkarni. Comparisons of citations in web of science, scopus, and google scholar for articles published in general medical journals. *JAMA*, 302(10):1092, September 2009. ISSN 0098-7484. doi: 10.1001/jama.2009.1307. URL http://dx.doi.org/10.1001/jama.2009.1307.

[8] Philippe Mongeon and Adèle Paul-Hus. The journal coverage of web of science and scopus: a comparative analysis. *Scientometrics*, 106(1):213–228, October 2015. ISSN 1588-2861. doi: 10.1007/s11192-015-1765-5. URL http://dx.doi.org/10.1007/s11192-015-1765-5.

[9] dblp computer science bibliography. dblp computer science bibliography, 2024. URL https://dblp.org.

[10] Jinseok Kim. Evaluating author name disambiguation for digital libraries: a case of dblp. *Scientometrics*, 116(3):1867–1886, June 2018. ISSN 1588-2861. doi: 10.1007/s11192-018-2824-5. URL http://dx.doi.org/10.1007/s11192-018-2824-5.

[11] Arezoo Aghaei Chadegani, Hadi Salehi, Melor Md Yunus, Hadi Farhadi, Masood Fooladi, Maryam Farhadi, and Nader Ale Ebrahim. A comparison between two main academic literature collections: Web of science and scopus databases. *Asian Social Science*, 9(5), April 2013. ISSN 1911-2017. doi: 10.5539/ass.v9n5p18. URL http://dx.doi.org/10.5539/ass.v9n5p18.

[12] Ludo Waltman, Nees Jan van Eck, Thed N. van Leeuwen, and Martijn S. Visser. Some modifications to the snip journal impact indicator.

Journal of Informetrics, 7(2):272–285, April 2013. ISSN 1751-1577. doi: 10.1016/j.joi.2012.11.011. URL http://dx.doi.org/10.1016/j.joi.2012.11.011.

[13] SCImago. Scimago journal & country rank, 2025. URL https://www.scimagojr.com.

[14] Devographics. State of css 2024: Libraries & tools, 2024. URL https://2024.stateofcss.com/en-US/tools/.

[15] Rasmus Lerdorf. Announce: Personal home page tools (php tools). https://groups.google.com/g/comp.infosystems.www.authoring.cgi/c/XYy9dPRA018, June 1995. Post su groups.google.ch, consultato il 6 luglio 2011; archiviato il 22 dicembre 2008.

[16] Docker. https://www.docker.com/, 2025. Accessed: 9 August 2025.

[17] Diego Michel. What is docker? the history of docker. https://medium.com/@digomic_88027/what-is-docker-ae478467669b, February 2023.

[18] Vladlen Koltun and David Hafner. The h-index is no longer an effective correlate of scientific reputation. *PLOS ONE*, 16(6):e0253397, June 2021. ISSN 1932-6203. doi: 10.1371/journal.pone.0253397. URL http://dx.doi.org/10.1371/journal.pone.0253397.