

**MISSION
INNOVATION**

accelerating the clean energy revolution

POA MATERIALI AVANZATI PER L'ENERGIA**PROGETTO IEMAP - Piattaforma Italiana Accelerata per i Materiali per
l'Energia****D1.6 - Descrizione completa della piattaforma IEMAP
e delle sue modalità di utilizzo**

Sergio Ferlito, Massimo Celino



D1.6 - Descrizione completa della piattaforma IEMAP e delle sue modalità di utilizzo

Sergio Ferlito (ENEA), Massimo Celino (ENEA)

Dicembre 2024

Report MISSION INNOVATION

Ministero dell'Ambiente e della Sicurezza Energetica - ENEA
Mission Innovation 2021-2024 - III annualità

Progetto: Piattaforma accelerata per i Materiali per l'Energia
Work package: MAP: Italian Energy Materials Acceleration Platform
Linea di attività 1.6: Utilizzo DB ENEA
Responsabile del Progetto: Massimo Celino (ENEA)
Responsabile della LA: Marialuisa Mongelli (ENEA)

Indice

1	Risultati attesi.....	6
2	Risultati ottenuti.....	7
2.1	Avanzamento della ricerca rispetto allo stato dell'arte internazionale con riferimento ai risultati ottenuti.....	8
3	Prodotti attesi.....	10
4	Prodotti ottenuti.....	11
5	Analisi degli scostamenti su attività e risultati.....	12
6	Sintesi delle attività svolte.....	13
7	Dettaglio delle attività svolte.....	13
7.1	Implementazione e rilascio della piattaforma.....	13
7.2	Sviluppo del modulo Python iemap-mi.....	23
7.3	Integrazione del modello AI (geoCGNN).....	25
7.4	Realizzazione della documentazione.....	26
7.5	Implementazione dell'autenticazione multi-utente.....	26
7.6	Attività per la conformità FAIR.....	27
7.7	Sviluppo della funzionalità di query semantica.....	30
7.8	Webinar e supporto tecnico.....	35
7.9	Applicazione Panel per data analysis di processi di deposizione Perovskite per moduli fotovoltaici 35	
7.10	Riferimenti Bibliografici di approfondimento per soluzioni utilizzate nella piattaforma IEMAP.....	37
8	Contributo delle eventuali consulenze alle attività sopra descritte.....	38
9	Pubblicazioni scientifiche.....	38
10	Eventi di disseminazione.....	38
11	Descrizione dei risultati ottenuti (manuale d'uso piattaforma IEMAP).....	39
11.1	Introduzione alla piattaforma IEMAP.....	39
11.2	Gestione Dati.....	41
11.2.1	<i>Inserimento dei metadati.....</i>	<i>41</i>
11.2.2	<i>Inserimento dei file associati ad un dato progetto.....</i>	<i>42</i>
11.2.3	<i>Formato di dati supportati.....</i>	<i>42</i>
11.2.4	<i>Linee guida per la preparazione e il caricamento dei dati.....</i>	<i>42</i>
11.2.5	<i>Politiche di conservazione e sicurezza dei dati.....</i>	<i>46</i>
11.3	Guida interfaccia utente: front-end.....	46
11.4	Utilizzo REST API.....	57
11.5	Utilizzo modulo Python iemap-mi.....	62

11.6	Utilizzo funzionalità AI: geoCGNN	72
11.7	Query semantica (in fase di test, non disponibile in produzione).....	74
11.8	Utilizzo applicazione Python Panel per Data Analysis dati da processo deposizione Perovskite per celle fotovoltaiche	79
11.9	Risoluzione problemi e FAQ	81
11.10	Glossario Tecnico Piattaforma IEMAP (suddiviso per aree tematiche).....	82

Indice delle figure

Figura 1. Esempio documento MongoDB relativo ai metadati dei progetti per IEMAP	19
Figura 2. Esempio documento nella Time Series Collection relativo ai dati sperimentali di Perovskite	20
Figura 3. Esempio documento collezione dati sintetici processi deposizione Perovskite.....	21
Figura 4. Specifiche Virtual Machine (VM) servizi IEMAP	21
Figura 5. Servizi in container Docker in esecuzione su VM IEMAP.....	22
Figura 6. Esempio risposta endpoint REST API servizio AI per modello geoCGNN	25
Figura 7. Esempio documento MongoDB collezione per autenticazione utenti.....	27
Figura 8. Schema soluzione Data Analysis per processo deposizione Perovskite per PV	36
Figura 9. Schema tecnico sintetico relativo all'intera piattaforma IEMAP	40
Figura 10. Template Excel inserimento metadati.....	43
Figura 11. Template Excel inserimento metadati - convalida dati	44
Figura 12. Template Excel inserimento metadati - convalida dati (valori).....	44
Figura 13. . Template Excel inserimento metadati - convalida dati aggiungi nuovi.....	45
Figura 14. Template Excel inserimento metadati - convalida dati attiva	45
Figura 15. Home page sito web (front-end) IEMAP.....	47
Figura 16. Home page front-end IEMAP: sezione inferiore	48
Figura 17. Front-end IEMAP: Get Started.....	48
Figura 18. Front-end IEMAP: pagina documentazione.....	50
Figura 19. Front-end IEMAP: pagina login.....	50
Figura 20. Front-end IEMAP: pagina validazione account.....	51
Figura 21. Esempio e-mail per attivazione account IEMAP.....	51
Figura 22. Front-end IEMAP: conferma attivazione account	52
Figura 23. Front-end IEMAP: pagina ricerca progetti.....	52
Figura 24. Front-end IEMAP: esempio filtro progetti per range date	53
Figura 25. Front-end IEMAP: esempio filtro progetti per Materiale.....	53
Figura 26. Front-end IEMAP: pagina progetti utente.....	54
Figura 27. Dettaglio Front-end IEMAP per utente che ha effettuato login.....	55
Figura 28. Front-end IEMAP: Upload metadati tramite file Excel o JSON.....	55
Figura 29. Front-end IEMAP: metadati caricati	56
Figura 30. Front-end IEMAP: pagina progetto utente con file associati	57
Figura 31. Schema funzionamento REST API.....	58
Figura 32. Pagina Swagger di documentazione e test REST API IEMAP	60
Figura 33. Dashboard Qdrant	74
Figura 34. Dashboard Qdrant: esempio punto inserto con payload costruito dai dati in MongoDB.....	75
Figura 35. Grafo ad albero dei dati IEMAP in Qdrant.....	76
Figura 36. Grafico 3D proiezione UMAP vettori densi dei dati IEMAP in Qdrant	78
Figura 37. Applicazione Python Panel per data analysis processo deposizione Perovskite per PV	80
Figura 38. Esempio grafici Data Analysis processo deposizione Perovskite.....	81

Indice delle tabelle

Tabella 1. FAIRness Score - punteggio sintetico raggiungimento principi FAIR	8
Tabella 2. Comparazione prestazioni principali web frameworks	16
Tabella 3. Valutazione FAIRness per piattaforma IEMAP.....	29
Tabella 4. Punteggio sintetico FAIRness piattaforma IEMAP	29

1 Risultati attesi

I risultati attesi nell'ambito della **Linea di Attività LA 1.6 – Piattaforma IEMAP** riguardano lo sviluppo, il rilascio e la documentazione tecnica della piattaforma digitale integrata **IEMAP (Italian Energy Materials Acceleration Platform)**, progettata per supportare la gestione, l'interrogazione, l'elaborazione e la valorizzazione dei dati computazionali e sperimentali relativi a materiali avanzati per l'energia.

I principali risultati attesi sono:

- **Completamento dell'implementazione della piattaforma IEMAP** con interfacce standard per l'interazione con utenti umani e applicativi software. La piattaforma è costituita da:
 - **front-end** sviluppato in VueJS v3, accessibile via browser all'indirizzo <https://iemap.enea.it>,
 - **servizio REST API** realizzato in Python tramite FastAPI, con autenticazione utente basata su JWT (tramite FastAPIUsers),
 - **database NoSQL MongoDB Enterprise**, installato on-premise,
 - **servizio REST API e front-end** eseguiti in ambienti virtualizzati su infrastruttura VMware ed esposti tramite NGINX,
 - **modulo Python open-source** iemap-mi, disponibile su PyPI, per interfacciarsi con la piattaforma in modalità programmatica,
 - **funzionalità di query semantica tramite DB vettoriale**, per l'accesso intelligente ai contenuti attraverso tecniche basate su LLM e AI (al momento in fase di test non ancora disponibile online).
- **Rilascio della documentazione tecnica e utente per l'utilizzo della piattaforma (il presente documento)**, includendo:
 - white paper descrittivo dell'architettura e delle modalità operative della piattaforma,
 - documentazione per l'utilizzo sia del servizio REST API che del modulo Python disponibile pubblicamente su PyPI, nonché delle funzionalità AI e di query semantica (DB vettoriale),
 - analisi caso d'uso sull'integrazione in piattaforma dei dati provenienti da un processo sperimentale di deposizione di materiale perovskite per celle solari, con indicazioni sulla replicabilità della procedura.
- **Verifica conformità ai principi FAIR (Findable, Accessible, Interoperable, Reusable)** per i dati raccolti e archiviati nonché dei vari elementi componenti la piattaforma IEMAP, anche in ottica di successiva scalabilità tramite soluzioni basate su **Docker** e orchestrazione con **Kubernetes (K8s)**.
- **Predisposizione dell'infrastruttura per l'integrazione con flussi computazionali e sperimentali**, sia interni che esterni all'ambiente ENEA, con supporto per l'importazione di dati da fonti eterogenee, nel rispetto dei formati interoperabili e dei metadati condivisi all'interno dell'ecosistema IEMAP.
- **Implementazione delle modalità di autenticazione e autorizzazione multi-utente**, con gestione delle policy di accesso attraverso JWT e FastAPIUsers, per garantire un utilizzo controllato della piattaforma da parte di sviluppatori, ricercatori, partner industriali e istituzionali.
- **Accessibilità pubblica della piattaforma attraverso canali sicuri e documentati**, per favorire la fruizione dei dati e dei servizi, sia da parte della comunità scientifica nazionale che di stakeholder industriali, nel rispetto delle policy di protezione del dato definite a livello di progetto.

- **Modularità e scalabilità dell'architettura software**, per permettere l'evoluzione della piattaforma in scenari distribuiti o cloud-native, e l'integrazione con ulteriori componenti analitiche, predittive o di automazione basate su AI.
- **Organizzazione di webinar divulgativi** destinati ai partner e agli utenti della piattaforma, finalizzati a illustrare l'architettura, le modalità di accesso e le funzionalità principali della piattaforma IEMAP.
- **Raccolta strutturata di feedback e supporto tecnico dedicato** con i partner del progetto, utili all'identificazione e alla risoluzione di problemi segnalati durante la fase di test e primo utilizzo operativo della piattaforma.

Obiettivo ultimo della linea LA1.6 è il completamento e la validazione dell'interfaccia computazionale della piattaforma IEMAP, rendendo disponibili i dati raccolti e processati nei diversi WP e LA secondo una logica di accesso regolamentato, interoperabilità e replicabilità della soluzione.

2 Risultati ottenuti

Tutti i risultati attesi descritti nella sezione precedente sono stati raggiunti nell'ambito della Linea di Attività LA 1.6 – Piattaforma IEMAP. La piattaforma digitale integrata è stata completata, documentata e rilasciata in forma stabile, risultando già operativa e accessibile su infrastruttura ENEA.

Di seguito si riportano i principali risultati conseguiti:

- **Rilascio della piattaforma IEMAP:**
 - **Prima versione pubblica rilasciata nel primo trimestre del 2023.**
 - **Versione finale completata nel corso del 2024, con risoluzione degli ultimi bug e ottimizzazioni infrastrutturali avvenute nel corrente trimestre.**
 - **Piattaforma attualmente accessibile online** all'indirizzo <https://iemap.enea.it>, con front-end sviluppato in VueJS, REST API FastAPI e modulo Python iemap-mi disponibile su PyPI.
 - **Servizi erogati da infrastruttura virtualizzata VMware, esposti tramite NGINX, con database MongoDB Enterprise installato on-premise.**
- **Documentazione tecnica e utente** completata e distribuita, in particolare tramite il presente documento che funge da manuale d'uso completo. Include:
 - *White paper* architettuale,
 - Documentazione per REST API, modulo Python,
 - Caso d'uso relativo all'integrazione dei dati sperimentali di deposizione di perovskite per celle solari tramite realizzazione applicazione Panel in Python.
- **Funzionalità di query semantica (DB vettoriale) sviluppata ma attualmente in fase di test:** l'integrazione con i moduli LLM per la ricerca semantica è stata implementata, ma non è ancora disponibile in ambiente di produzione. Il rilascio finale è previsto in un successivo aggiornamento iterativo.
- **Conformità ai principi FAIR perseguita parzialmente:** è stato effettuato un primo lavoro di mappatura dei metadati, standardizzazione dei formati e definizione di criteri di accesso strutturati. Di seguito una tabella sintetica che riassume il soddisfacimento dei principi FAIR:

Commentato [SF1]: Massimo Colino Date corrette? Credo di sì o sbaglio

Commentato [MC2R1]: Credo vada bene

FAIRness Score:

Componente	Findable (6)	Accessible (6)	Interoperable (6)	Reusable (6)	Totale (24)	FAIRness (%)
MongoDB	5/6	6/6	3/6	5/6	19/24	79.2%
REST API	5/6	6/6	4/6	6/6	21/24	87.5%
Sito Web	4/6	5/6	3/6	4/6	16/24	66.7%

Tabella 1. FAIRness Score - punteggio sintetico raggiungimento principi FAIR

Ulteriori dettagli in merito sono riportati nella sezione 7.6

- **Autenticazione e gestione utenti** pienamente operativa: sistema JWT tramite FastAPIUsers, con policy di accesso in scrittura per i partner del progetto mentre in lettura non è richiesta alcuna autenticazione.
- **Scalabilità e interoperabilità** dell'architettura garantite da un design modulare e containerizzabile, predisposto per una futura orchestrazione tramite Docker/Kubernetes in ambienti distribuiti o cloud-native (ad oggi non implementata).
- **Webinar divulgativi** sono stati organizzati per illustrare il funzionamento e l'utilizzo della piattaforma ai partner di progetto e ad altri utenti interessati.
 1. 5/9/2024, "IEMAP – Front-end walkthrough and REST API "
 2. 1/10/2024, "Enhancing IEMAP with Semantic Search Capabilities for Files and Metadata Retrieval", registrazione disponibile al link https://www.youtube.com/watch?v=ijzIY4B_KIM
 3. 1/10/2024, "IEMAP – Front-end walkthrough and REST API", registrazione disponibile al link <https://www.youtube.com/watch?v=aKIW70fyJWk>
 4. 7/11/2024, "Vector Databases & Retrieval-Augmented Generation (RAG), a comprehensive exploration"
- **Attività di raccolta feedback e supporto tecnico** effettuate tramite sessioni su Microsoft Teams e scambi e-mail dedicati con i partner progettuali, finalizzate all'identificazione e risoluzione di problematiche segnalate durante la fase di test e primo utilizzo.

2.1 Avanzamento della ricerca rispetto allo stato dell'arte internazionale con riferimento ai risultati ottenuti

La piattaforma **IEMAP** rappresenta un significativo passo avanti nell'ambito delle infrastrutture digitali per la gestione e l'analisi dei dati relativi a materiali avanzati per l'energia. Pur essendo una soluzione emergente, IEMAP integra funzionalità innovative che la distinguono nel panorama internazionale.

Confronto con piattaforme internazionali

Numerose piattaforme internazionali offrono servizi per la gestione e l'analisi di dati sui materiali:

- **Materials Project** fornisce un ampio database di proprietà dei materiali, con strumenti per la predizione di nuove strutture e proprietà.

- **NOMAD (Novel Materials Discovery)** offre un'infrastruttura per l'analisi interattiva basata su AI di dati FAIR, con strumenti come il NOMAD AI Toolkit.
- **AiiDA e Materials Cloud** si concentrano sulla tracciabilità della provenienza dei dati e sull'automazione dei workflow computazionali.
- **AFLOW** fornisce un repository di dati computazionali con strumenti per la predizione di proprietà termodinamiche ed elettroniche.

Rispetto a queste piattaforme, IEMAP si distingue per:

- **Integrazione di un modello geoCGNN** per la predizione del potenziale redox e dell'energia di formazione a partire da file CIF, validato e ulteriormente addestrato da ENEA sull'infrastruttura CRESCO.
- **Predisposizione per l'integrazione di dati sperimentali e computazionali**, favorendo l'interoperabilità tra diverse fonti di dati.
- **Orientamento verso l'adozione dei principi FAIR**, con una roadmap per migliorare la conformità in termini di accessibilità e riutilizzabilità dei dati.
- **Integrazione ricerca semantica**, sebbene ancora in test e non disponibile online, per la ricerca, tramite linguaggio naturale, nei metadati dei progetti IEMAP (eventualmente estendibile anche ai file associati ai progetti, cosa che renderebbe tale funzionalità ancora più utile)
- **Applicazione Data Analytics per dati sperimentali da deposizione Perovskite** per realizzazione celle fotovoltaiche

Innovazioni e prospettive

L'integrazione del modello geoCGNN consente a IEMAP di offrire predizioni rapide e accurate di proprietà chiave dei materiali, facilitando la selezione e l'ottimizzazione di materiali per applicazioni energetiche. Questa funzionalità, combinata con la capacità di gestire dati sia sperimentali che computazionali, posiziona IEMAP come una piattaforma versatile e orientata all'innovazione.

Sebbene IEMAP offra attualmente un set di servizi più limitato rispetto a piattaforme consolidate come Materials Project o NOMAD, la sua architettura modulare e la focalizzazione su casi d'uso specifici la rendono una base solida per futuri sviluppi e integrazioni.

Riferimenti per approfondimento

- **Materials Project:** <https://materialsproject.org/Wikipedia+1Wikipedia+1>
- **NOMAD AI Toolkit:** <https://nomad-lab.eu/ai-toolkit>
- **AiiDA e Materials Cloud:** <https://www.materialscloud.org/arXiv+1arXiv+1>
- **AFLOW:** <http://aflow.org/arXiv>
- **Articolo su geoCGNN:** <https://www.nature.com/articles/s43246-021-00194-3>

3 Prodotti attesi

Nell'ambito della Linea di Attività LA 1.6 – Piattaforma IEMAP, i prodotti attesi comprendono componenti software, documentazione tecnica, moduli di interfaccia e funzionalità AI, in grado di abilitare la fruizione e l'integrazione della piattaforma IEMAP in contesti di ricerca e sviluppo sui materiali per l'energia.

I principali prodotti attesi sono:

- Piattaforma IEMAP in versione accessibile al pubblico, composta da:
 - interfaccia front-end web sviluppata in VueJS (v.3),
 - servizi REST API sviluppati in Python tramite FastAPI,
 - database MongoDB Enterprise on-premise, predisposto per l'integrazione di dati computazionali e sperimentali,
 - infrastruttura virtualizzata (VMware), con servizi esposti tramite NGINX.
- Modulo Python open-source **iemap-mi**, distribuito pubblicamente su PyPI (<https://pypi.org/project/iemap-mi/>), per l'interazione programmatica con la piattaforma e l'esecuzione di operazioni di caricamento, interrogazione, scarico e visualizzazione dei dati.
- Documentazione tecnica completa destinata a utenti e sviluppatori:
 - white paper descrittivo dell'architettura logica e tecnica,
 - manuale d'uso per l'interfaccia web e REST API,
 - guida al modulo Python,
 - riferimenti all'integrazione delle funzionalità AI e query semantiche.
- Sistema di autenticazione e autorizzazione degli utenti, implementato tramite FastAPIUsers con autenticazione JWT, che abilita l'accesso controllato e multi-ruolo alla piattaforma.
- Integrazione (prevista ma al momento in test e validazione) del sistema di query semantica tramite DB vettoriale, per l'accesso intelligente ai dati attraverso **embedding** e **modelli LLM**.
- Modulo AI per la predizione del potenziale redox e dell'energia di formazione di materiali cristallini, basato sulla **rete neurale geoCGNN**, richiamabile tramite REST API o modulo Python, alimentato da file CIF forniti dall'utente.
- Documentazione per l'integrazione dei dati sperimentali relativi al caso d'uso sulla deposizione di materiali Perovskite, utile a dimostrare l'adattabilità della piattaforma a scenari di laboratorio.
- Sessioni formative e webinar divulgativi, finalizzati all'introduzione alla piattaforma e al suo utilizzo, destinati a partner e stakeholder progettuali.

4 Prodotti ottenuti

Nel corso della Linea di Attività LA 1.6, sono stati sviluppati, rilasciati e documentati i seguenti prodotti hardware/software, in coerenza con quanto previsto e con alcune estensioni rispetto al piano iniziale:

- Piattaforma IEMAP rilasciata in versione completa e operativa, accessibile pubblicamente all'indirizzo <https://iemap.enea.it>. Il sistema è costituito da:
 - interfaccia web sviluppata in VueJS v3,
 - servizi REST API sviluppati in Python con FastAPI, autenticazione JWT via FastAPIUsers,
 - database MongoDB Enterprise, installato on-premise e gestito in ambiente virtualizzato VMware,
 - servizi esposti tramite NGINX, con configurazione di sicurezza e accesso da rete esterna controllato.
- Modulo Python iemap-mi, rilasciato su PyPI (<https://pypi.org/project/iemap-mi/>), che consente a utenti avanzati e sviluppatori di interfacciarsi programmaticamente con la piattaforma per:
 - invio di dati computazionali o sperimentali,
 - esecuzione di interrogazioni e operazioni CRUD su MongoDB,
 - accesso a servizi predittivi e AI disponibili via API.
- Funzionalità AI integrate nella piattaforma, tra cui:
 - predizione del potenziale redox e dell'energia di formazione di cristalli a partire da file CIF tramite modello geoCGNN, validato e addestrato da ENEA su infrastruttura CRESCO (basato su Nature Communications Materials 2021),
 - API per invocare il modello da front-end o modulo Python.
- Documentazione tecnica e utente, fornita tramite il presente deliverable, comprendente:
 - white paper dell'architettura della piattaforma,
 - guida tecnica per REST API e modulo Python,
 - documentazione funzionale delle componenti AI e delle modalità di autenticazione.
- Sistema di autenticazione multi-utente, attivo e collaudato, basato su JSON Web Token (JWT), con gestione ruoli e permessi per utenti registrati.
- Funzionalità di query semantica tramite DB vettoriale, attualmente disponibile solo in ambiente di test, in attesa di rilascio in produzione dopo la validazione finale.
- Caso d'uso dimostrativo, sviluppato e documentato, per l'integrazione di dati da un processo sperimentale di deposizione di materiale perovskite, con relativa pipeline di acquisizione e gestione dati replicabile.
- Sessioni di presentazione e disseminazione tecnica, svolte tramite webinar online con i partner di progetto.

- Supporto tecnico e raccolta feedback, effettuati tramite canali di contatto diretti (email e sessioni Teams), finalizzati alla risoluzione tempestiva delle problematiche segnalate durante le fasi di test e primo utilizzo.

5 Analisi degli scostamenti su attività e risultati

Nell'ambito della Linea di Attività LA 1.6 – Piattaforma IEMAP non si sono verificati scostamenti significativi rispetto agli obiettivi tecnico-scientifici e al piano di lavoro inizialmente previsto. Tutte le attività fondamentali sono state completate nei tempi stabiliti, e i prodotti software attesi sono stati rilasciati con piena funzionalità. Di seguito si riportano i principali elementi di verifica:

- Piattaforma rilasciata e operativa: l'architettura applicativa è stata completamente implementata, testata e messa in produzione, includendo front-end, REST API, modulo Python iemap-mi, database MongoDB e infrastruttura di accesso sicura.
- Funzionalità AI disponibili tramite API: il modello geoCGNN per la predizione del potenziale redox e dell'energia di formazione è stato correttamente integrato e validato. L'invocazione è disponibile da front-end o via modulo Python, e accessibile in ambiente di produzione.
- Documentazione tecnica e manuale d'uso: il presente documento funge da deliverable di riferimento per l'intero sistema, includendo white paper architetturale, guida utente, documentazione per REST API e modulo Python, oltre al caso d'uso su materiali perovskite.
- Query semantica su DB vettoriale: rappresenta l'unico punto parzialmente in ritardo. La funzionalità è stata sviluppata e testata con successo in ambiente di sviluppo, ma non è ancora stata promossa in ambiente di produzione. Questo è stato deliberatamente posticipato al fine di completare la validazione tecnica e migliorare l'accuratezza dei risultati nella fase di retrieval. Il rilascio definitivo è pianificato per il prossimo aggiornamento iterativo della piattaforma. Non si tratta di un'inadempienza, bensì di un perfezionamento evolutivo basato su feedback tecnici ricevuti dai partner e dalle prime sessioni di utilizzo.
- Conformità ai principi FAIR: la compliance FAIR è stata perseguita con strumenti e criteri coerenti con quanto descritto nei documenti progettuali. È prevista la presentazione di una tabella FAIR con i relativi indici di avanzamento (vedi sezione 2.1). Il livello di conformità è attualmente parziale, ma in evoluzione verso una piena aderenza attraverso successive estensioni funzionali e aggiornamenti infrastrutturali.
- Supporto tecnico e gestione delle criticità: eventuali problematiche minori segnalate dai partner (es. nella configurazione degli accessi, formati dei dati, oppure instabilità su connessioni API) sono state prontamente risolte grazie a canali di comunicazione efficaci (e-mail e sessioni Teams). Nessuna criticità strutturale ha ostacolato l'ottenimento dei risultati.

In conclusione, l'attività LA 1.6 ha rispettato il piano esecutivo approvato, raggiungendo tutti gli obiettivi previsti. Le minime variazioni tecniche riscontrate sono state affrontate con efficacia, senza impatto sulle milestone o sui benefici attesi per la comunità scientifica e tecnologica coinvolta nel progetto IEMAP.

6 Sintesi delle attività svolte

Le attività della LA 1.6 hanno permesso di completare, rilasciare e documentare la piattaforma IEMAP (<https://iemap.enea.it>). Sono stati realizzati il front-end (realizzato come SPA tramite VueJS v.3), le REST API (realizzate in Python tramite framework FastAPI), il database (MongoDB on-premise versione Enterprise ospitato sull'infrastruttura di CRESCO ENEA) e l'infrastruttura di hosting (VMware/NGINX).

È stato rilasciato su PyPI il modulo Python iemap-mi per l'interazione programmatica.

È stato integrato un modello AI (geoCGNN) per predizioni (potenziale redox, energia di formazione) da file CIF via API. È stata prodotta documentazione completa (white paper, guide API/modulo, caso d'uso perovskiti). È stata implementata l'autenticazione multi-utente (JWT tramite modulo FASTAPIUsers).

È stata perseguita parzialmente la conformità FAIR. La funzionalità di query semantica è sviluppata ma in attesa di rilascio. Sono stati svolti webinar e fornito supporto tecnico.

Gli obiettivi sono stati raggiunti senza scostamenti significativi.

7 Dettaglio delle attività svolte

Questa sezione descrive in dettaglio le attività svolte nell'ambito della Linea di Attività LA 1.6 - Piattaforma IEMAP. Si evidenzia il processo di sviluppo, le tecnologie utilizzate, le sfide affrontate e le soluzioni implementate per raggiungere i risultati attesi. Sono incluse anche le attività che, pur avendo richiesto risorse, non hanno portato direttamente ai risultati previsti, al fine di fornire un quadro completo e trasparente delle attività progettuali.

7.1 Implementazione e rilascio della piattaforma

- **Sviluppo del front-end in VueJS:**

-

Il front-end della piattaforma IEMAP è stato sviluppato utilizzando il framework VueJS (versione 3, <https://vuejs.org/>). La scelta di VueJS è stata dettata dalla sua leggerezza, reattività e dalla disponibilità di un ecosistema ricco di librerie e componenti. Lo sviluppo ha incluso la creazione di interfacce utente per l'interazione con i dati, la visualizzazione dei risultati e la gestione delle funzionalità della piattaforma. Sono stati utilizzati componenti specifici per la gestione dei form, la visualizzazione tabellare dei dati e la creazione di grafici interattivi.

Per realizzare il front-end di IEMAP si è partiti da un template commerciale che è stato adeguato al contesto specifico in termini di colori, immagini e font in modo da mantenere una uniformità, per quanto possibile, con il portale principale del progetto Mission Innovation (<https://mission-innovation.it/>). Si tratta di una applicazione SPA (Single Page Application) in cui la navigazione tra le vari pagine avviene totalmente client side grazie ad un meccanismo di "routing" fornito dal componente dedicato fornito da VUEJS [1]. Grazie a tale metodica di sviluppo di UI si ottiene un front-end altamente responsivo in quanto solamente i dati dinamici vengono richiesti di volta in volta al server senza dover caricare l'intera pagina. Tali dati sono serializzati in formato JSON e forniti tramite endpoint REST API, di questi alcuni richiedono un'autenticazione (prevalentemente tutti gli endpoint per la scrittura dei dati nel DB), fornita tramite JWT [2], [3], mentre altri sono liberi.

L'accesso alle sezioni riservate, ovvero che richiedono autenticazione, è garantito sia dal meccanismo di "navigation guards" fornito da VueRouter (<https://router.vuejs.org/>) che dagli

endpoint REST API protetti da autenticazione (basata su JWT) che nel caso di accesso tramite web avviene grazie all'utilizzo di cookie HttpOnly [4], [5] utilizzati per la memorizzazione dei JWT necessari. Ulteriori moduli JS sono stati utilizzati per lo sviluppo delle funzionalità complete della piattaforma, di seguito un elenco dei moduli principali e delle funzionalità svolte da ogni componente:

1. *@vee-validate/zod* per la validazione dei dati JSON relativi ai progetti da inserire in IEMAP o dei dati JSON estratti dal template .xlsx che è possibile utilizzare in alternativa per inserire un progetto in IEMAP
2. *@bugsnag/js* per la notifica remota di eccezioni non gestite e/o alert a seguito di specifici eventi
3. *@vueuse* per utilities utilizzate per estendere le funzionalità di componenti base di VUEJS 3
4. *unplugin-vue-router*, per la gestione del routing automatico basato su organizzazione dei file in directory
5. *apexcharts* per i grafici
6. *axios* per le richieste asincrone agli endpoint REST API
7. *pinia* per la gestione dello stato lato client-side
8. *vue-i18n* per l'internazionalizzazione (non implementata appieno)
9. *vue3-json-viewer* per la visualizzazione del JSON relativo ai metadati del progetto nel front-end
10. *bulma-css* per la gestione dei CSS
11. *tippy* per gestire i tooltip
12. *notify* per la visualizzazione di popup di notifica

- **Sviluppo delle REST API con FastAPI:**

I servizi REST API sono stati sviluppati in Python utilizzando il framework FastAPI (<https://fastapi.tiangolo.com/> [6]), scelto per la sua elevata velocità di esecuzione, l'adozione di standard moderni (ASGI, OpenAPI, JSON Schema) e la capacità di generare automaticamente una documentazione interattiva tramite Swagger UI [7] e ReDoc. FastAPI consente inoltre di scrivere API tipizzate, sicure e auto-documentate, grazie all'integrazione nativa con Pydantic (<https://docs.pydantic.dev/latest/> [8]) per la validazione dei dati e con Python type hints.

Le API realizzate gestiscono le seguenti funzionalità principali della piattaforma IEMAP:

- autenticazione e autorizzazione degli utenti tramite JWT e FastAPIUsers;
- gestione degli account (registrazione, login, refresh token, cambio password);
- operazioni CRUD su collezioni del database MongoDB (dati computazionali, sperimentali, metadati);
- upload e parsing dei file CIF;
- accesso e interrogazione del modello AI geoCGNN;
- interrogazioni avanzate su database vettoriale (query semantica) – attualmente in fase di test;
- integrazione con sistemi esterni (modulo Python, front-end web) tramite chiamate asincrone.

Ovviamente FASTAPI non è l'unica soluzione possibile, di seguito alcuni elementi che hanno indotto alla scelta di tale soluzione:

- Supporto nativo per asynchronous I/O, utile per gestire carichi elevati.
- Documentazione automatica e interattiva generata da Swagger UI.
- Integrazione naturale con Pydantic per la validazione dei dati in ingresso/uscita.
- Elevata manutenibilità e leggibilità del codice grazie alla struttura modulare.
- Ecosistema moderno e in rapida crescita, compatibile con strumenti DevOps.
- Semplice integrazione con modelli AI sviluppati in Python
- Semplice estensione con moduli quali FastAPIUser per la gestione delle funzionalità di autenticazione

FastAPI è una scelta eccellente per lo sviluppo rapido di API REST in Python, grazie alla sua sintassi chiara, alla validazione automatica dei dati e alla generazione di documentazione interattiva. Tuttavia, presenta alcune limitazioni, soprattutto in termini di prestazioni e gestione delle dipendenze, che possono influenzare la scalabilità e la manutenzione di progetti di grandi dimensioni.

Limiti di FastAPI

Prestazioni inferiori rispetto a linguaggi compilati: Essendo basato su Python, FastAPI non può competere con framework sviluppati in linguaggi compilati come Go o Rust in termini di throughput e latenza.

Ad esempio, benchmark indicano che FastAPI gestisce circa 4.500 richieste al secondo, mentre Gin (Go) può raggiungere fino a 10.000 richieste al secondo, con una latenza media inferiore [9], [10].

Gestione delle dipendenze complessa: Python ha un sistema di gestione delle dipendenze meno robusto rispetto ad altri linguaggi. Strumenti come pip, virtualenv o poetry possono introdurre complessità, soprattutto in ambienti di produzione o in progetti con molte dipendenze. Nel caso specifico si è utilizzato **poetry** ma si consiglia anche **uv** [11].

Utilizzo di risorse: Applicazioni FastAPI tendono a consumare più memoria e CPU rispetto a equivalenti sviluppati in Go o Bun [12], rendendole meno adatte per ambienti con risorse limitate o per applicazioni ad alta intensità di traffico.

Comunità e supporto: Sebbene in crescita, la comunità di FastAPI è ancora più piccola rispetto a framework più maturi, il che può limitare la disponibilità di risorse, plugin e supporto.

Framework	Linguaggio	Prestazioni (req/s)	Latenza media	Note principali
Gin	Go	~10.000	~0,8 ms	Alte prestazioni, basso consumo di risorse, ideale per microservizi
Elysia	TypeScript	~2.450	~0,5 ms	Sviluppato su Bun, eccellente per applicazioni ad alte prestazioni
Fastify	Node.js	~4.300	~1,3 ms	Leggero e veloce, buona alternativa in ambiente JavaScript
Spring Boot	Java	~7.800	~1,4 ms	Robusto e maturo, adatto per applicazioni enterprise
Express	Node.js	~1.100	~2,3 ms	Semplice da usare, ma meno performante rispetto ad alternative più moderne

Tabella 2. Comparazione prestazioni principali web frameworks

Avendo evidenziato pro e contro della soluzione adottata si vuole inoltre evidenziare che una soluzione basata su FASTAPI qualora si vogliano integrare/esporre modelli AI realizzati in Python resta tra le migliori soluzioni possibili e dalla più rapida e semplice realizzazione sebbene le prestazioni in termini unicamente di performance servizio REST API possano essere inferiori a soluzione compilate.

L'elenco degli endpoint disponibili nella piattaforma IEMAP è organizzato in "namespace" logici (es. /health, /projects, /stats, /users, /ml).

Di seguito un esempio (l' url base per ogni endpoint è <https://iemap.enea.it/rest>):

GET /api/v1/health → controlla che il servizio REST API sia online e la connessione a DB funzionante

GET /api/v1/project/list → fornisce la lista dei progetti caricati nel portale IEMAP

GET /api/v1/ml/model1/{file_ciph} → invoca il modello AI geoCGNN che accetta l'upload di un file .cif per fornire il potenziale redox e l'energia di formazione predetta dal modello AI

L'elenco completo è disponibile al link <https://iemap.enea.it/rest/docs>.

E' possibile utilizzare CURL oppure l'interfaccia fornita da Swagger al link precedente per invocare gli endpoint senza dover scrivere codice.

Sicurezza, validazione e testing:

I dati in input/output sono validati tramite modelli Pydantic, che permettono controllo su tipi, lunghezze, pattern regex, vincoli.

La sicurezza è garantita tramite autenticazione JWT con scadenza e refresh, CORS restriction e header HTTP per protezione da XSS, non esposizione di dati sensibili nei messaggi di errore, grazie anche alle funzionalità di Pydantic.

La protezione da attacchi CSRF non è necessaria in quanto si tratta di un'API REST stateless, ma è supportata nel front-end se richiesto.

È stato implementato un set base minimale di unit test e integration test (Pytest + HTTPX) per validare endpoint, errori, gestione token e edge case.

Configurazione e gestione del database MongoDB:

Il database utilizzato per la piattaforma IEMAP è MongoDB Enterprise, installato on-premise. MongoDB è un database NoSQL che offre flessibilità nello schema e scalabilità orizzontale. La configurazione del database ha incluso la definizione degli schemi per la memorizzazione dei dati computazionali e sperimentali, l'ottimizzazione delle query e l'implementazione di procedure di backup e ripristino.

I metadati di ogni singolo progetto caricato da utenti registrati e validati sulla piattaforma IEMAP sono memorizzati in un cluster MongoDB (ver. 6) gestito tramite Ops Manager tramite documenti aventi il seguente schema:

```
{
  "_id": ObjectId,
  "identifier": String | null,
  "iemap_id": String,
  "provenance": {
    "email": String,
    "affiliation": String,
    "createdAt": Date,
    "updatedAt": Date
  },
  "project": {
    "name": String,
    "label": String,
    "description": String
  },
  "process": {
    "isExperiment": Boolean,
    "method": String,
    "agent": {
      "name": String,
      "version": String | null
    }
  },
  "material": {
    "formula": String,
    "elements": [String],
    "input": Object | null,
    "output": Object | null
  },
  "parameters": [ ... ]
}
```

```
{
  // esempio ipotetico:
  "name": String,
  "value": Number,
  "unit": String,
  "file": String | null
}
],
"properties": [
{
  "name": String,
  "value": Number,
  "unit": String,
  "file": String | null
}
],
"v": String,
"files": [
{
  "hash": String,
  "name": String,
  "extention": String,
  "size": String,
  "createdAt": Date,
  "updatedAt": Date
}
]
}
```

Alla data odierna, **9 maggio 2025**, risultano memorizzati **782 progetti/metadati**. Ogni documento di progetto include un campo array denominato **files**, che raccoglie i riferimenti a tutti i file associati al progetto stesso. Come già descritto in precedenza, nel database **MongoDB** vengono archiviati **solo i metadati** del progetto. I file veri e propri sono invece salvati su un **file system distribuito Ceph** [13] sui server del centro ricerche ENEA di Brindisi. Di conseguenza, il campo files non contiene i file stessi, ma piuttosto i riferimenti ai file memorizzati su Ceph, insieme ad alcune informazioni descrittive dei file stessi (dimensione, data creazione, etc.).

Il processo di caricamento e associazione di un file a un progetto avviene come segue:

1. Il file viene caricato in modo **incrementale**, attraverso uno **stream di dati a blocchi** (con dimensione massima di 1 MB), verso il **back-end** implementato in **FastAPI**.
2. Questo approccio consente di mantenere il server **responsivo** anche in presenza di upload simultanei, evitando il rischio di esaurire la memoria disponibile.
3. Una volta completato l'upload, il back-end calcola l'**hash** del file.
L'hash è una **stringa alfanumerica univoca** generata da una funzione crittografica applicata al contenuto del file. Anche una minima modifica del file produce un hash completamente diverso, rendendolo un riferimento affidabile per verificarne l'**integrità** e l'**unicità**.

4. L'hash così ottenuto, insieme ad altre proprietà del file (es. nome, formato, dimensione, data), viene memorizzato nel campo `files` del documento di progetto in MongoDB, come riferimento al file memorizzato su Ceph.

Ovviamente qualsiasi operazione di ricerca, download o cancellazione (l'endpoint relativo alla cancellazione file è protetto da autenticazione con l'ulteriore verifica che l'utente che ha creato il file è lo stesso che sta cercando di cancellarlo) di files avviene tramite endpoint dedicati.

Un estratto dal DB **iemap** è riportato a titolo di esempio di seguito:

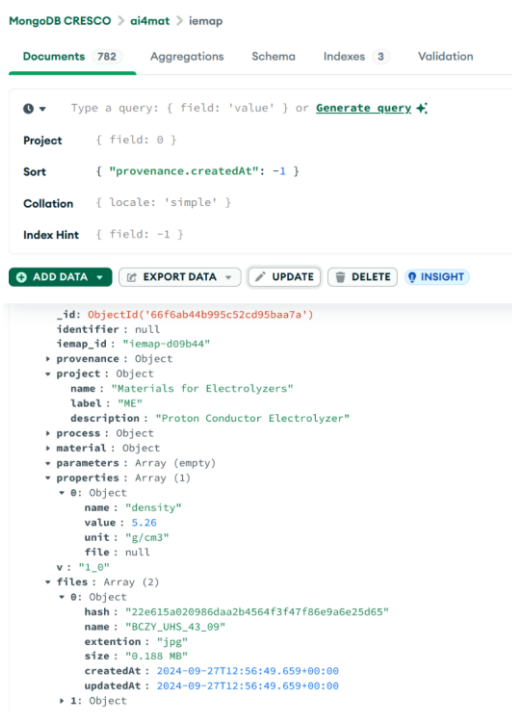


Figura 1. Esempio documento MongoDB relativo ai metadati dei progetti per IEMAP

Il database MongoDB utilizzato per i metadati di IEMAP è denominato **“ai4mat”** ed i metadati sono memorizzati nella collezione **“iemap”**. Tali metadati sono estratti programmaticamente lato client (e validati prima dell’invio al server) dal file `template.xlsx` che l’utente può scaricare dal front-end e compilare anche offline. In alternativa è possibile definire i metadati come JSON e caricarli direttamente sempre dal front-end o ancora tramite modulo Python **“iemap-mi”**. Anche in questi casi si ha una validazione dei metadati prima dell’invio al server, una ulteriore verifica viene fatta grazie al modulo `Pydantic` lato back-end.

I dati sperimentali del processo di deposizione della perovskite per moduli fotovoltaici sono memorizzati in un database separato dal nome “perovskite” in due collezioni apposite:

1. “sensors_readings” di tipo time series (TS) collection che raccoglie I dati di misura
2. “source_events” che raggruppa i dati sintetici e statistici dei processi di deposizione

Di seguito un esempio di tali dati:

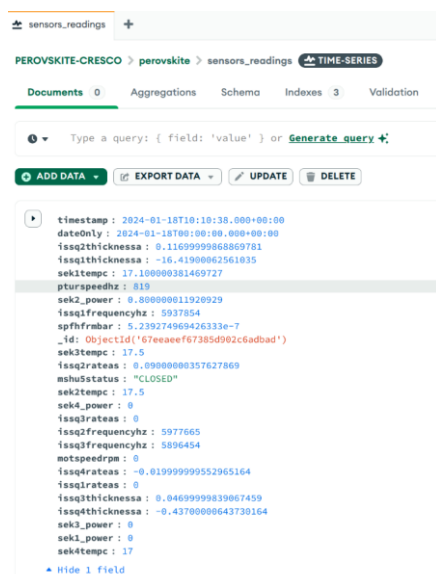


Figura 2. Esempio documento nella Time Series Collection relativo ai dati sperimentali di Perovskite



- **Realizzazione dell'infrastruttura di hosting:**

L'infrastruttura di hosting per la piattaforma IEMAP è basata su VMware vSphere per l'host della macchina virtuale Linux con le seguenti caratteristiche:



Tutti i servizi sono esposti tramite **NGINX**, che svolge il ruolo di web server e reverse proxy. L'infrastruttura virtualizzata consente di isolare i servizi, migliorando al contempo l'efficienza nell'utilizzo delle risorse hardware.

NGINX [15] (versione 1.18.0) gestisce le richieste HTTP/HTTPS, il bilanciamento del carico e contribuisce alla sicurezza della piattaforma, anche grazie all'integrazione con **Fail2ban** [16]. Il front-end, sviluppato come Single Page Application (SPA) con **Vue.js 3**, e tutte le REST API sono esposti tramite direttive "location", che reindirizzano specifici path/url verso container Docker in ascolto su porte interne dedicate.

Il rinnovo automatico dei certificati **TLS**, rilasciati da **Let's Encrypt**, è gestito da NGINX in combinazione con **Certbot** [17], utilizzando il meccanismo challenge http-01 [18].

In sintesi, l'infrastruttura di hosting è composta da diverse componenti, ciascuna distribuita ed eseguita in container Docker [19] distinti.

Sebbene questi container non siano direttamente accessibili dall'esterno, vengono rese pubbliche tramite NGINX, che funge da unico punto di accesso esterno.

I container in esecuzione sulla VM sono i riportati nell'immagine seguente:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
8fcd9e077c5	sergioana/iemap-perovskite-rest-api:0.8	"uwicorn app.main:re..."	3 weeks ago	Up 10 days	0.0.0.0:8010->8000/tcp, [::]:8010->8000/tcp
6998168832ca	sergioana/iemap-rest-api:0.8.2	"uwicorn main:app --..."	3 months ago	Up 10 days (healthy)	8000/tcp, 0.0.0.0:8005->80/tcp, [::]:8005->80/tcp
932c6e0d0ff3	sergioana/geogm-geoconv:1.0	"uwicorn main:app --..."	2 months ago	Up 10 days	0.0.0.0:8000->8000/tcp, [::]:8000->8000/tcp
5854795ee55	iemap-api:latest	"uwicorn main:app --..."	17 months ago	Up 10 days	0.0.0.0:8000->80/tcp, [::]:8000->80/tcp

Figura 5. Servizi in container Docker in esecuzione su VM IEMAP

1. Container FASTAPI per i servizi REST API relativi al processo sperimentale di deposizione perovskite
2. Container con applicazione FASTAPI per le funzionalità REST API generiche
3. Container che espone tramite REST API il modello AI geoCGNN
4. Container relativo alle API IEMAP usate dal modulo Python iemap-mi

Dunque, la soluzione realizzata è essenzialmente una architettura a microservizi che può costituire la base di una soluzione altamente scalabile grazie all'integrazione di un orchestratore di container quali Kubernetes [20]. Il modello di intelligenza artificiale è disponibile **esclusivamente in modalità di inferenza**, ovvero è già stato **pre-addestrato** e non può essere ulteriormente aggiornato o fine-tuned. L'inferenza avviene all'interno di un **container Docker** che include il framework **PyTorch** in versione **CPU-only**, poiché la macchina virtuale di esecuzione non dispone di GPU.

Sebbene il container sia stato ottimizzato tramite una multi-stage build, questa soluzione non rappresenta un'opzione di deployment ideale in termini di efficienza. Tuttavia, alcuni test empirici hanno mostrato tempi di risposta medi intorno ai 300 ms, rendendo la soluzione comunque adeguata per scenari a bassa o media intensità di carico. Per un deployment più performante, si consiglia la conversione del modello nel formato **ONNX**, pensato per l'inferenza efficiente e supportato da diversi runtime ottimizzati. In ambienti CPU-only, l'uso di **ONNX Runtime** o **OpenVINO** consente di migliorare sensibilmente le prestazioni rispetto a PyTorch, riducendo i tempi di inferenza grazie all'ottimizzazione dell'esecuzione su CPU multi-core. In presenza di una GPU, ONNX Runtime permette invece di sfruttare backend ad alte prestazioni come **CUDA** o **TensorRT**, ottenendo tempi di risposta ancora più bassi e una maggiore scalabilità, rendendo il deployment adatto anche a scenari di carico elevato.

7.2 Sviluppo del modulo Python *iemap-mi*

Il modulo Python **iemap-mi** è stato sviluppato con l'obiettivo di facilitare l'interazione programmatica con la piattaforma IEMAP (Italian Energy Materials Acceleration Platform), una piattaforma di gestione e valorizzazione dei dati sperimentali e computazionali nel dominio dei materiali avanzati per l'energia. Questo modulo consente di accedere ai servizi REST messi a disposizione dalla piattaforma, in particolare per l'autenticazione, la gestione dei progetti, il caricamento e la consultazione dei file associati, oltre all'esecuzione di predizioni basate su modelli di intelligenza artificiale (AI) integrati nel sistema. Tale modulo è disponibile su Pypi al link

Caratteristiche principali

Il modulo è concepito per un utilizzo **asincrono**, e utilizza la libreria **httpx** per l'invio di richieste HTTP non bloccanti, garantendo prestazioni migliori e una migliore scalabilità nei contesti applicativi. La validazione dei dati in input e in output è affidata a **pydantic**, una libreria ampiamente utilizzata in Python per la gestione rigorosa di modelli di dati. L'adozione di queste tecnologie moderne rende il modulo **robusto, leggibile e facilmente integrabile** in workflow automatizzati o pipeline scientifiche.

Tra le funzionalità più rilevanti offerte dal modulo si evidenziano:

- Autenticazione basata su **JSON Web Token (JWT)** con funzioni di login, refresh e logout;
- Creazione e gestione di **metadati di progetto** strutturati secondo uno schema JSON compatibile con il backend della piattaforma;
- Caricamento di **file associati ai progetti** su un filesystem distribuito (CephFS), con gestione automatica delle estensioni supportate;
- Interrogazione di progetti esistenti tramite **paginazione**;
- Invocazione di **modelli AI** integrati per la predizione di proprietà dei materiali, in particolare tramite file CIF.

Struttura dei metadati

I metadati dei progetti vengono memorizzati nel backend MongoDB della piattaforma IEMAP e seguono uno schema definito, disponibile anche in formato JSON Schema nel modulo. Questo schema comprende i seguenti elementi principali:

- **project**: contiene informazioni di base come nome, etichetta e descrizione;
- **material**: specifica la formula chimica del materiale oggetto dello studio;
- **process**: descrive il metodo utilizzato (es. simulazione, esperimento) e lo strumento/algoritmo impiegato;
- **parameters e properties**: liste di oggetti che descrivono i parametri di input e le proprietà osservate o calcolate, con nome, valore e unità di misura.

Un esempio minimale di creazione di un payload conforme può essere costruito tramite la classe **IEMAPProject**, oppure usando il metodo **ProjectHandler.build_project_payload()**:

```
from iemap_mi.model.project import IEMAPProject
from iemap_mi.handlers.project_handler import ProjectHandler

# Costruzione del payload
project_payload = ProjectHandler.build_project_payload({
    "project": {
```

```

    "name": "NalonBatteryStudy",
    "label": "Study of Na-Ion cathode",
    "description": "DFT + ML based analysis of Na-ion battery cathodes"
  },
  "material": {
    "formula": "NaFePO4"
  },
  "process": {
    "method": "DFT+GNN",
    "agent": {
      "name": "geoCGNN",
      "version": "1.0"
    }
  },
  "isExperiment": False
},
"parameters": [
  {"name": "temperature", "value": 300, "unit": "K"}
],
"properties": [
  {"name": "formation_energy", "value": "-3.12", "unit": "eV"}
]
})

```

Funzionalità AI: il modello *geoCGNN*

Uno degli aspetti più innovativi del modulo è l'integrazione con un modello AI basato su **Graph Neural Networks (GNN)**, ispirato a *Crystal Graph Convolutional Neural Networks (CGCNN)*, con estensioni e ottimizzazioni sviluppate da ENEA. Il modello, denominato **geoCGNN**, è stato addestrato su CRESCO, l'infrastruttura HPC di ENEA, e consente di effettuare previsioni su proprietà chiave di materiali per batterie, come l'energia di formazione e il potenziale redox.

Per la predizione, l'utente deve fornire un file CIF (Crystallographic Information File) relativo allo **stato scarico del materiale**, che costituisce il riferimento strutturale. Il modello è stato addestrato su dataset provenienti da *Materials Project* (oltre 150.000 materiali per l'energia di formazione, e circa 4.000 per il potenziale redox), opportunamente filtrati per escludere elementi nobili.

Un esempio di uso potrebbe essere:

```
await client.ai_handler.predict_redox_potential("NaFePO4.cif")
```

Questa funzione restituisce il valore stimato del potenziale redox per il materiale descritto.

Documentazione e contributi

La documentazione completa è pubblicata su [ReadTheDocs](https://iemap.enea.it/rest/docs), mentre la documentazione API è disponibile in formato Swagger alla pagina <https://iemap.enea.it/rest/docs>.

Il modulo è **open-source**, sotto licenza MIT, ed è ospitato su GitHub. Contributi esterni sono incoraggiati attraverso pull request, a condizione che il codice sia testato e conforme agli standard esistenti. Per ulteriori domande è possibile scrivere a iemap.support@enea.it.

Conclusione

Lo sviluppo del modulo **iemap-mi** rappresenta un'infrastruttura software flessibile e scalabile a supporto della piattaforma IEMAP. L'integrazione di funzionalità asincrone, la validazione rigorosa dei dati e l'accesso a strumenti di AI avanzati lo rendono uno strumento chiave per la comunità scientifica impegnata nello sviluppo di materiali innovativi per l'energia. Grazie alla sua struttura modulare e alla disponibilità di esempi pratici, il modulo può essere facilmente incorporato in pipeline di ricerca automatizzate, promuovendo al tempo stesso la **FAIRness** dei dati e la riproducibilità scientifica.

7.3 Integrazione del modello AI (geoCGNN)

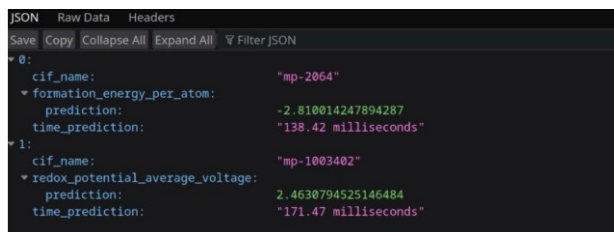
Una delle caratteristiche innovative della piattaforma IEMAP è l'integrazione di un modello di intelligenza artificiale per la predizione delle proprietà dei materiali. È stato integrato il modello geoCGNN (geometric convolutional graph neural network) per la predizione del potenziale redox e dell'energia di formazione a partire da file CIF (Crystallographic Information File).

Il modello è stato validato e ulteriormente addestrato da ENEA sull'infrastruttura di calcolo ad alte prestazioni CRESCO. L'accesso al modello è fornito tramite REST API, consentendo sia al front-end che al modulo Python "iemap-mi" di utilizzarlo, **ma al momento l'accesso è disponibile solo tramite modulo Python e/o REST API.**

Gli endpoint associati a tale servizio sono due:

1. https://iemap.enea.it/ai/health_geoCGNN
2. https://iemap.enea.it/ai/ai_materials:predict_what (dove *predict_what* può assumere i valori "redox-potential"/"formation_energy")

Il primo endpoint consente una rapida verifica dello stato di funzionamento del modello AI (utilizza un file CIF di input presente sul server) e fornisce una risposta del tipo:



```

JSON  Raw Data  Headers
Save Copy Collapse All Expand All Filter JSON
0:
  cif_name: "mp-2064"
  formation_energy_per_atom:
    prediction: -2.810014247894287
    time_prediction: "138.42 milliseconds"
1:
  cif_name: "mp-1003402"
  redox_potential_average_voltage:
    prediction: 2.4630794525146484
    time_prediction: "171.47 milliseconds"
  
```

Figura 6. Esempio risposta endpoint REST API servizio AI per modello geoCGNN

L'endpoint 2) è facilmente utilizzabile tramite curl, ad esempio:

```
curl -X POST "https://iemap.enea.it/ai/ai_materials:predict_what" \
-F "file=@./test/database/cif_batteries/mp-1003402.cif" \
-F "predict_what=redox-potential"
```

La documentazione Swagger è disponibile alla url <https://iemap.enea.it/ai/docs>

Il modello messo in produzione tramite PyTorch ed esposto con servizio REST API implementato in FASTAPI in container Docker dedicato prende spunto dal modello pubblicato su Nature: “A geometric-information-enhanced crystal graph network for predicting properties of materials” (il codice del modello proposto in tale paper è disponibile al link <https://github.com/Tinystormiojo/geo-CGNN>).

Come accennato al paragrafo precedentemente nella sezione dedicata all’hosting dei servizi, per un deployment ideale di modelli AI si consiglia di convertire il modello Pytorch in formato ONNX [21] con successivo deployment tramite TensorRT [22], nel caso si disponga di GPU, oppure tramite OpenVINO qualora si disponga solamente di CPU.

7.4 Realizzazione della documentazione

La documentazione è un elemento cruciale per garantire l'usabilità e la comprensione della piattaforma IEMAP. È stata prodotta una documentazione completa, destinata sia agli utenti finali che agli sviluppatori. La documentazione include il presente documento che descrive l'architettura della piattaforma, guida per l'utilizzo delle REST API e del modulo Python, nonché la guida all'uso della applicazione Panel distribuita tramite Docker relativa all'integrazione di dati sperimentali derivanti dal processo di deposizione di perovskite. Oltre quanto riportato nel presente paragrafo una vera e propria guida all'uso è riportata nel paragrafo 11 più avanti. Ulteriore documentazione è disponibile online ai link:

- <https://iemap.enea.it/rest/docs> (documentazione create in automatico da Swagger)
- <https://pypi.org/project/iemap-mi/> (repository ufficiale con link a documentazione, note d’esempio, codice d’esempio nella directory /examples all’interno del codice scaricabile)
- <https://iemap.enea.it/ai/docs> (documentazione Swagger per le REST API relative al modello geoCGNN)
- <https://iemap.enea.it/perovskite/docs> (documentazione Swagger per le REST API relative ai dati sperimentali del processo di deposizione perovskite)

7.5 Implementazione dell'autenticazione multi-utente

Per garantire un accesso sicuro e controllato alla piattaforma, è stato implementato un sistema di autenticazione e autorizzazione multi-utente. Il sistema è basato su JSON Web Token (JWT) per l'autenticazione e sulla libreria FastAPIUsers per la gestione degli utenti e dei permessi. Questo sistema consente di definire diversi ruoli e policy di accesso, garantendo che solo gli utenti autorizzati possano accedere a determinate funzionalità e dati. Nel caso specifico non sono stati implementati ruoli ma si è provveduto all’implementazione del solo sistema di autenticazione in modo che solamente gli utenti con un account valido possono creare dei progetti ed inserire i relativi metadati e file associati. Il meccanismo di creazione dell’account richiede semplicemente l’inserimento di una mail valida ed avviene come segue:

1. Alla ricezione/inserimento della e-mail, oltre all’affiliazione, viene creato un account “temporaneo” nel database MongoDB. A tale scopo si è utilizzata una collezione “UserAuth” con un indice TTL [23] di 24h, definito sul campo “is_verified” in questo modo se tale campo non diventa True entro 24 ore l’account viene automaticamente eliminato
2. Una volta fatta la richiesta di creazione account, l’utente che si è appena registrato riceve una mail contenente un link che rimanda ad una opportuna pagina che invia, in maniera non visibile all’utente, un token temporaneo che contiene le informazioni specifiche dell’utente per consentir una validazione sicura dell’account, in altre parole non basta

conoscere l'url di validazione e/o la mail per validare una qualsivoglia e-mail realizzando così una protezione efficace da eventuali attacchi.

Gli endpoint necessari alle varie fasi nonché all'eventuale modifica password e/o recupero della stessa sono create in automatico dal framework FASTApiUsers, tali endpoints così come le possibili metodiche di autenticazione sono descritte nel dettaglio nella documentazione all'indirizzo <https://fastapi-users.github.io/fastapi-users/latest/> (da notare che tale framework utilizza Beanie ODM che è stato integrato nel modulo Motor utilizzato per tutte le funzioni di accesso a MongoDB).

Con riferimento a tale documentazione sono state adottate le seguenti scelte:

- Metodo di trasporto: Authorization Header e Cookie (HTTP-Only Cookie)
- Strategia: JWT

In IEMAP non è stato utilizzato ma è anche possibile integrare il supporto per OAuth2 [24]. La classe base fornita da FASTApiUser è stata ampliata in modo da integrare l'informazione "affiliation", "created_at" e "last_login" in modo da tener traccia di tale utili informazioni.

Nel complesso tale soluzione si è rivelata efficace sebbene non fornisca funzionalità più evolute come una semplice creazione di ruoli e/o supporto di metodi di autenticazione social come è ad esempio possibile con Auth0 (<https://auth0.com/>) o Frontegg (<https://frontegg.com/>)

Di seguito un esempio di documento nella collezione adibita all'autenticazione degli utenti della piattaforma IEMAP:

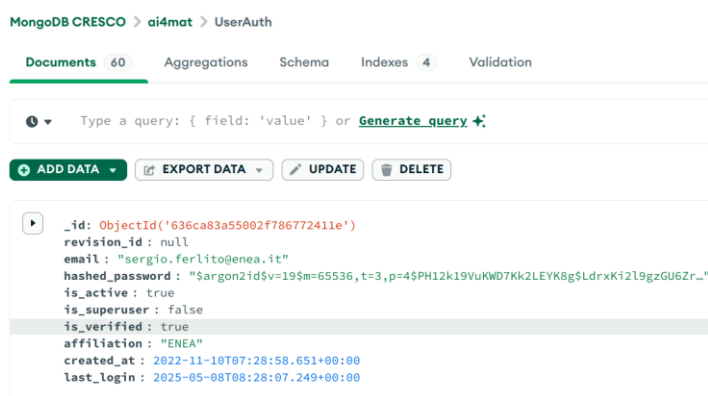


Figura 7. Esempio documento MongoDB collezione per autenticazione utenti

7.6 Attività per la conformità FAIR

I principi FAIR (Findable, Accessible, Interoperable, Reusable) sono linee guida per la gestione dei dati scientifici volte a renderli:

- Trovabili da persone e macchine tramite identificatori e metadati ricercabili,
- Accessibili secondo protocolli standard e sicuri,
- Interoperabili grazie a vocabolari e formati condivisi,
- Riutilizzabili con metadati completi e licenze chiare.

L'obiettivo della piattaforma IEMAP è aderire a questi principi per garantire che i dati raccolti e generati possano essere utilizzati in maniera trasparente, tracciabile e integrabile in ecosistemi di ricerca nazionali e internazionali.

Scelte tecnologiche e limitazioni

L'infrastruttura dati di IEMAP si basa su MongoDB Enterprise, scelto per la sua flessibilità, scalabilità e supporto a strutture documentali eterogenee. Tuttavia, MongoDB non supporta nativamente modelli semantici (RDF, OWL, SPARQL) come i triplestore (es. Virtuoso, Blazegraph) o i database concettuali come TypeDB.

Di conseguenza:

- **la semantica dei dati è stata implementata esternamente come ontologia OWL**, derivata dallo schema MongoDB e pubblicabile su repository aperti (es. GitHub);
- **l'interoperabilità semantica non è ancora completamente operativa all'interno della piattaforma, ma è formalizzata e documentata**, e rappresenta un potenziale evolutivo verso architetture knowledge-graph compatibili (es. TypeDB, Neo4j + RDF).

Semantica definita per IEMAP

A partire dalla struttura dei documenti MongoDB e dal template Excel, è stata definita una ontologia minimale in OWL che rappresenta concetti e relazioni chiave:

Classi OWL: Project, Process, Material, Parameter, Property, Agent, Provenance

Proprietà tipizzate: hasName, hasLabel, hasValue, hasFormula, hasMethod, hasVersion, ecc.

Proprietà oggetto: hasProject, hasProcess, hasMaterial, hasAgent, ecc.

Classi dinamiche aperte: per Parameter, Property, Agent, l'utente può inserire nuovi valori (flessibilità garantita, ma semantica formalizzabile in futuro)

L'ontologia è pensata per essere espansa, versionata e collegabile a vocabolari standard come EMMO, ChEBI, QUDT.

L'ontologia è stata pubblicata su Zenodo al link <https://zenodo.org/records/1539669>.

In considerazione di quanto sopra si riporta la seguente tabella di valutazione FAIRness per la piattaforma IEMAP:

Principio	Criterio	MongoDB	REST API	Sito Web
Findable	Identificatore persistente (iemap_id)	2	2	2
	Metadati ricercabili via Python / Web	2	1	1
	Documentazione di discovery (Swagger)	1	2	1
Accessible	Accesso via HTTPS	2	2	2
	Sicurezza con JWT	2	2	1
	Licenza MIT/Open Source	2	2	2
Interoperable	Formati standard (JSON, CSV)	2	2	2
	Ontologie/vocabolari condivisi	1	2	1
	Versioning documenti/API	2	2	1
Reusable	Documentazione utente	1	2	1
	Licenza di riuso chiara	2	2	2

Tabella 3. Valutazione FAIRness per piattaforma IEMAP

Da quanto sopra se ne deriva il seguente punteggio sintetico.

FAIRness Score:

Componente	Findable (6)	Accessible (6)	Interoperable (6)	Reusable (6)	Totale (24)	FAIRness (%)
MongoDB	5/6	6/6	3/6	5/6	19/24	79.2%
REST API	5/6	6/6	4/6	6/6	21/24	87.5%
Sito Web	4/6	5/6	3/6	4/6	16/24	66.7%

Tabella 4. Punteggio sintetico FAIRness piattaforma IEMAP

Analisi critica

- I punteggi per **Findable** e **Accessible** restano molto buoni, grazie all'uso di identificatori chiari (iemap_id), accesso via HTTPS con autenticazione JWT, e alla presenza di documentazione Swagger per l'API REST.
- Il principio **Reusable** è ben supportato dal versionamento automatico dei documenti, dalla presenza di una licenza MIT, e da una struttura dei metadati coerente e documentata.

- Il punteggio per **Interoperable** è stato **significativamente migliorato** in seguito alla pubblicazione dell'ontologia IEMAP in formato OWL, Turtle e JSON-LD su Zenodo (<https://zenodo.org/records/15396693>, DOI [10.5281/zenodo.15396692](https://doi.org/10.5281/zenodo.15396692)), con licenza aperta e documentazione associata.

In particolare:

- L'ontologia è ora formalmente disponibile e può essere referenziata in strumenti semantici esterni;
- Tuttavia, non è ancora integrata operativamente nei meccanismi di query o validazione all'interno della piattaforma (es. controllo automatico dei valori);
- Inoltre, mancano ancora riferimenti a vocabolari esterni consolidati (es. per unità di misura standard, proprietà chimico-fisiche o agenti sperimentali), che limitano la piena interoperabilità semantica;
- Non è ancora disponibile un endpoint RDF nativo o una serializzazione diretta dei documenti MongoDB in JSON-LD o OWL.

Prospettive evolutive

Per incrementare la FAIRness, si prevede di:

- pubblicare l'ontologia IEMAP in formato OWL su GitHub con versionamento;
- mappare i concetti principali a vocabolari come EMMO, ChEBI, QUDT;
- sperimentare **integrazioni future con DB semantici** (es. TypeDB o Neo4j RDF);
- aggiungere **endpoint SPARQL federati** o trasformatori RDF in pipeline di export.

7.7 Sviluppo della funzionalità di query semantica

È stata sviluppata una funzionalità di query semantica per consentire agli utenti di interrogare la piattaforma utilizzando il linguaggio naturale. Questa funzionalità si basa sull'integrazione tra database vettoriali (QDrant) e modelli linguistici di grandi dimensioni (LLM), con l'obiettivo di migliorare l'accessibilità e l'intelligenza della ricerca su dataset complessi legati ai materiali energetici.

La query semantica differisce dalla ricerca tradizionale basata su parole chiave, in quanto consente di trovare contenuti rilevanti anche se formulati in modo diverso.

Ad esempio, un utente potrebbe chiedere:

"Trova esperimenti con capacità specifica superiore a 80 $\mu\text{F}/\text{cm}^2$ in materiali a base di litio"

e ricevere risultati pertinenti anche se la descrizione tecnica nei documenti utilizza formulazioni diverse come:

"specific capacitance: 82 $\mu\text{F}/\text{cm}^2$ using LiMn_2O_4 composite".

Per abilitare questa funzionalità, è stato necessario definire un processo di **trasformazione dei documenti e metadati originariamente memorizzati in MongoDB**. Questo perché la versione on-premise di MongoDB (inclusa quella Enterprise) **non supporta nativamente le query di tipo semantico o ibrido**, funzionalità rese disponibili solo nella versione cloud (MongoDB Atlas) e limitatamente ad alcune API avanzate.

Pertanto, si è optato per la **migrazione verso QDrant**, un database vettoriale specializzato nella gestione di embedding densi e sparsi.

Ogni documento MongoDB è stato analizzato per identificare:

- Campi da mantenere come **metadati** (es. nome progetto, ente di provenienza, identificatore univoco, data di creazione)
- Campi da trasformare in una **rappresentazione testuale aggregata** (combined_text), successivamente convertita in un **embedding denso** tramite modelli di tipo transformer
- **Termini tecnici e simboli** rilevanti (es. "LiClO₄", "W", "0.6 V") da convertire in **sparse embedding** per supportare **ricerche keyword-based** all'interno di una strategia di **query ibrida**

Cosa sono gli sparse embedding e perché usarli

Gli **sparse embedding** rappresentano i testi scientifici attraverso una struttura simile a un dizionario ponderato di parole chiave, ottenuta con tecniche come **TF-IDF** o **BM25**. A differenza degli embedding densi, non apprendono la semantica, ma catturano la **presenza esatta di termini** e la loro **rilevanza statistica** nel contesto.

Questa componente è essenziale quando si lavora con **terminologia specialistica**, come nel caso di:

"esperimenti con LiClO₄"

Un **LLM per Embedding generalista** potrebbe non riconoscere "LiClO₄" come un concetto importante, o addirittura spezzarlo in token ("Li", "Cl", "O₄"), riducendo l'efficacia dell'embedding semantico. Invece, uno **sparse embedding** assegna a "LiClO₄" un peso preciso e lo indicizza come termine esatto.

Questo assicura che:

- documenti contenenti "LiClO₄" vengano recuperati anche con match esatto
- documenti semanticamente simili vengano comunque inclusi grazie alla componente densa

Il modello di **embedding adottato è BAAI/bge-m3**, un modello multilingua e multi-task adatto all'estrazione semantica in contesti scientifici.

Pro:

- Eccellente bilanciamento tra precisione e dimensione
- Supporta retrieval instruction-based ("Query: ... Document: ...")
- Compatibile con il framework QDrant

Contro:

- Richiede formattazione curata del testo per ottenere buoni risultati
- Meno performante su linguaggi altamente tecnici rispetto a modelli di dominio

Per la generazione di sparse embedding si è adottato il modello **SPLADE (naver/splade-cocondenser)** [25], in grado di mappare ogni documento in uno spazio sparso dove i token chiave ricevono un peso basato sulla loro rilevanza statistica. Questo approccio permette il recupero accurato di espressioni esatte, mantenendo la compatibilità con la ricerca ibrida in QDrant.

In futuro si prevede di realizzare una soluzione completa RAG (eventualmente Graph-RAG), a tal riguardo si prevede di utilizzare come LLM Gemma3:4B, reso disponibile tramite **Ollama o vLLM**, si tratta di un modello in grado di generare, riscrivere e spiegare query tecniche in linguaggio naturale.

L'esecuzione della soluzione completa RAG richiederà necessariamente l'utilizzo di una GPU per avere risposte in tempi accettabili.

Pro:

- Requisiti di risorse hardware non particolarmente esose (solo 4B di parametri)
- Risposte fluide anche in italiano e inglese tecnico

Contro:

- Capacità limitata rispetto a modelli di dimensioni superiori
- Alcune espressioni numeriche o specialistiche vanno verificate manualmente

Come **database vettoriale** si è utilizzato **Qdrant**, una piattaforma open-source ad alte prestazioni progettata specificamente per la **ricerca semantica e la gestione di dati strutturati e non strutturati** in forma di vettori.

Qdrant è scritto in Rust, il che garantisce **elevata efficienza e basso consumo di risorse**, ed è particolarmente adatto a scenari di **intelligenza artificiale applicata** come il retrieval di documenti, l'indicizzazione semantica e i motori di raccomandazione.

L'inserimento dei dati nel DB richiede la generazione di quello che viene definito un **payload**, cioè una struttura che contiene:

- uno o più **termini vettoriali** (embedding denso e/o sparso) da usare per la ricerca semantica
- **metadati strutturati** (campi testuali, numerici, array, booleani, ecc.) utili per il **pre-filtraggio** dei dati mediante query tradizionali

Questo approccio ibrido consente di **combinare filtri precisi** (es. per data, materiale, o ente di ricerca) con la **flessibilità semantica** di un motore LLM avanzato.

Qdrant offre inoltre:

- **Supporto nativo per la ricerca ibrida** (dense + sparse embedding) con configurazione del bilanciamento (alpha)
- API gRPC e REST documentate
- Meccanismi di replica, snapshot e partizionamento automatico
- Interfacce Python (qdrant-client), JavaScript e altri SDK
- **Licenza open-source Apache 2.0**, che consente l'uso gratuito anche in contesti commerciali

Grazie a queste caratteristiche, Qdrant rappresenta una soluzione scalabile e facilmente integrabile per applicazioni di ricerca avanzata su contenuti scientifici e tecnici.

A tal scopo si è definito un campo **combined_text** contenente una descrizione arricchita dei campi del documento MongoDB originali seguenti:

- Metodo sperimentale
- Formula dei materiali
- Parametri e proprietà con valori numerici
- Strumentazione utilizzata

In aggiunta, sono stati mantenuti come **metadati** i campi: `iemap_id`, `project.name`, `provenance.affiliation`, `created_at`, `material.elements`.

Le query supportate includono:

- **Ricerca semantica:** formulata in linguaggio naturale, basata su combined_text
- **Query per metadati:** filtraggio per ente, progetto, data, ID
- **Query ibrida:** combinazione tra embedding denso e sparse embedding per catturare sia la semantica che le parole chiave (es. "LiClO4" o "W" come elementi chimici)

Di seguito il codice Python per la creazione del payload da inserire in Qdrant:

```
def build_combined_text(doc: dict) -> str:
    project_desc = f"Project description: {doc.get('project', {}).get('description', '')}."
    process_method = f"Process method: {doc.get('process', {}).get('method', '')}."
    agent_name = f"Instrumentation: {doc.get('process', {}).get('agent', {}).get('name', '')}."
    formula = f"Material formula: {doc.get('material', {}).get('formula', '')}."
    elements = doc.get('material', {}).get('elements', [])
    elements_text = f"Chemical elements involved: {' '.join(elements)}."

    parameters = doc.get('parameters', [])
    param_text = "Experimental parameters:\n" + "\n".join(
        f"- {p['name']}: {p['value']}" for p in parameters if 'name' in p and 'value' in p
    )

    properties = doc.get('properties', [])
    prop_text = "Measured properties:\n" + "\n".join(
        f"- {p['name']}: {p['value']}" for p in properties if 'name' in p and 'value' in p
    )

    return "\n".join([project_desc, process_method, agent_name, formula, elements_text,
                      param_text, prop_text])
```

La collezione creata su Qdrant ha il nome "" e contiene due "named vectors", uno "default" che è un dense vector e l'altro dal nome "sparse" che è di tipo sparse.

La funzione base per inserire i documenti nella collezione Qdrant è la seguente:

```
def upsert_document(client, collection_name: str, doc_id: str, dense_vector: List[float],
                    sparse_vector: Dict, payload: Dict):
    point = PointStruct(
        id=doc_id,
        vector={
            "default": dense_vector,
            "sparse": SparseVector(
                indices=sparse_vector["indices"],
                values=sparse_vector["values"]
            )
        },
        payload=payload
    )
    client.upsert(collection_name=collection_name, points=[point])
```

Dove il payload è costruito dai dati estratti da MongoDB con la funzione `build_combined_text` su riportata mentre i vettori `dense_vector` e `sparse_vector` risultano da:

```
embedder = DenseEmbedder(model_name="BAAI/bge-m3", use_fp16=True)
dense_vector = embedder.embed(combined_text)

splade = SpladeEncoder()
sparse_vector = splade.encode(combined_text)
```

Dove `SpladeEncoder` è fornito dalla libreria `Qdrant` mentre `DenseEmbedder` è costruito dal modello `BGE-m3` come indicato su HuggingFace <https://huggingface.co/BAAI/bge-m3>.

Pianificazione per rilascio futuro e sfide tecniche relative

Sebbene lo sviluppo e i test locali siano stati completati, il **rilascio in produzione è stato posticipato** per consentire:

- Verifica della robustezza del matching semantico su dati reali
- Ottimizzazione delle prestazioni nella ricerca ibrida
- Validazione da parte degli utenti finali (es. ricercatori, tecnologi)

Le principali **sfide da affrontare** includono:

- Validazione e test del modello di embedding selezionato
- Definizione di un set di query e relative risposte attese per valutare le prestazioni della soluzione
- Valutazione del modello LLM selezionato e relativo deployment ottimale

L'intera pipeline sinora sviluppata è stata testata con successo su un set di oltre 700 documenti estratti da MongoDB, con inserimento validato e ricerca ibrida tramite interfaccia REST e dashboard `QDrant` (ma tutto questo solo in ambiente di sviluppo). Ulteriori test su larga scala e query multiutente sono previsti nella fase di rilascio in produzione.

Architettura e integrazione con la piattaforma

L'integrazione con le altre componenti della piattaforma sarà realizzata secondo la seguente architettura **containerizzata**:

- **QDrant** in container Docker per lo storage e la ricerca vettoriale
- **FASTAPI** per esporre via REST il modello di embedding e processare i testi
- **Ollama** come server LLM locale (es. Gemma 3:4B) per la generazione e riformulazione delle query

La soluzione di semantic search qui dettagliata sarà resa disponibile all'interno del modulo Python **iemap-mi** già sviluppato e disponibile su Pypi al fine di consentirne un facile utilizzo da parte di ogni utente. Eventualmente solo in un secondo momento sarà integrata la funzionalità nel front-end di IEMAP.

⚠ Attualmente questa architettura è **presente solo in ambiente di sviluppo** e verranno fornite a corredo **solo demo ed esempi** ottenuti tramite esecuzione su notebook locale nella sezione 11.8 più avanti in questo documento.

7.8 Webinar e supporto tecnico

Per supportare l'adozione e l'utilizzo della piattaforma IEMAP, sono stati organizzati webinar divulgativi rivolti ai partner del progetto e ad altri utenti interessati. Inoltre, è stato fornito supporto tecnico continuo tramite canali di comunicazione diretti, come e-mail e sessioni Microsoft Teams, per risolvere eventuali problemi e rispondere alle domande degli utenti. Nel dettaglio sono stati tenuti i seguenti Webinar tramite la piattaforma Teams:

5. 5/9/2024, "IEMAP – Front-end walkthrough and REST API "
6. 1/10/2024, "Enhancing IEMAP with Semantic Search Capabilities for Files and Metadata Retrieval", registrazione disponibile al link https://www.youtube.com/watch?v=ijzJY4B_KIM
7. 1/10/2024, "IEMAP – Front-end walkthrough and REST API", registrazione disponibile al link <https://www.youtube.com/watch?v=aKIW70fyJWk>
8. 7/11/2024, "Vector Databases & Retrieval-Augmented Generation (RAG), a comprehensive exploration"

Tali webinar hanno visto la partecipazione in media di una cinquantina di partecipanti con maggiore partecipazione agli eventi 1 e 2.

In merito alle richieste di supporto oltre alla mail dedicata allo scopo visto il numero non così elevato di partner e di richieste si è utilizzato prevalentemente la mail dello scrivente e/o del responsabile di progetto e la risoluzione delle problematiche segnalate ha richiesto in media due settimane di tempo, si è trattato prevalentemente di piccoli bug e/o di chiarimento sulle metodiche di inserimento dei metadati o dei file di progetto.

7.9 Applicazione Panel per data analysis di processi di deposizione Perovskite per moduli fotovoltaici

Nell'ambito della linea di attività LA 4.3 – “Definizione di materiali ed architetture per celle solari a base di perovskite”, i colleghi impegnati nella sperimentazione hanno progressivamente adottato una metodologia accelerata per l'ottimizzazione dei processi di deposizione tramite evaporazione termica. A supporto di tale approccio, nell'ambito di LA 1.6 è stata sviluppata una soluzione integrata nella piattaforma IEMAP che consente funzioni di data analysis sui dati sperimentali dei processi di deposizione di film di perovskite per moduli fotovoltaici.

La soluzione si compone dei seguenti elementi:

- un'applicazione sviluppata in Go ed eseguita direttamente sul PC di controllo del processo di deposizione;
- un'interfaccia web-based costruita con il framework Panel di Python;

- una collezione dedicata nel database ai4mat per l'archiviazione strutturata dei dati raccolti;
- un servizio REST API esposto per operazioni di CRUD e query analitiche a supporto dell'applicazione Panel.

La soluzione è esemplificata nell'immagine seguente.

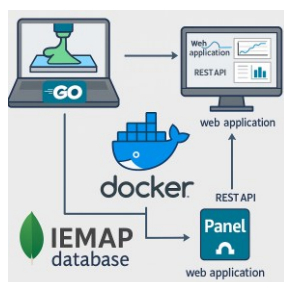


Figura 8. Schema soluzione Data Analysis per processo deposizione Perovskite per PV

L'intero sistema è fornito come immagine Docker pubblicata su Docker Hub:

<https://hub.docker.com/r/sergioeanx/iemap-perovskite-ui>

consentendo all'utente di avviare rapidamente un container sul proprio PC tramite Docker Desktop.

Senza dover installare manualmente alcun pacchetto o dipendenza, l'utente può così accedere in locale a un'interfaccia intuitiva per l'esplorazione dei dati e la visualizzazione dei profili temporali dei parametri di processo. Parallelamente, l'interazione programmata tramite REST API abilita scenari di automazione e integrazione in flussi di lavoro personalizzati.

Rispetto agli anni precedenti, in cui l'estrazione e il confronto dei dati avvenivano manualmente, la nuova soluzione consente una gestione strutturata, tracciabile e riusabile delle informazioni sperimentali, favorendo l'individuazione di pattern ripetibili, condizioni favorevoli e possibili cause di instabilità nei film realizzati. Quanto realizzato può essere interpretato come un caso d'uso replicabile, applicabile ad altri laboratori o processi sperimentali. La metodologia adottata, composta da un front-end Panel containerizzato, database condiviso su IEMAP e funzionalità analitiche esposte via REST, costituisce una base estendibile, anche con strumenti di intelligenza artificiale, per il monitoraggio e l'ottimizzazione automatizzata di esperimenti complessi.

7.10 Riferimenti Bibliografici di approfondimento per soluzioni utilizzate nella piattaforma IEMAP

- [1] "Comprehensive Guide to Vue-Router | by Ossamakharbaq | Medium." Accessed: Apr. 28, 2025. [Online]. Available: <https://medium.com/@ossamakharbaq/comprehensive-guide-to-vue-router-de2f3bf55b85>
- [2] "JWT - FastAPI Users." Accessed: May 08, 2025. [Online]. Available: <https://fastapi-users.github.io/fastapi-users/latest/configuration/authentication/strategies/jwt/>
- [3] "JSON Web Token Introduction - jwt.io." Accessed: May 08, 2025. [Online]. Available: <https://jwt.io/introduction>
- [4] "Today's rabbit hole: securing JWTs for authentication, httpOnly cookies, CSRF tokens, secrets & more - DEV Community." Accessed: Apr. 28, 2025. [Online]. Available: <https://dev.to/petrussola/today-s-rabbit-hole-jwts-in-http-only-cookies-csrf-tokens-secrets-more-1jbp>
- [5] "Token in http-only cookie, secure way of authorization. | Multitude IT Labs." Accessed: Apr. 28, 2025. [Online]. Available: <https://medium.com/multitude-it-labs/store-token-in-http-only-cookie-239a15ea121f>
- [6] "FastAPI 101: A Comprehensive Tutorial for Building Lightning-Fast APIs | by Prince Samuel | The Modern Scientist | Medium." Accessed: May 08, 2025. [Online]. Available: <https://medium.com/the-modern-scientist/fastapi-101-a-comprehensive-tutorial-for-building-lightning-fast-apis-90832b45579d>
- [7] "An Introduction to Python FastAPI & Swagger UI | by Aysel Aydin | Medium." Accessed: May 08, 2025. [Online]. Available: <https://ayselaydin.medium.com/an-introduction-to-python-fastapi-swagger-ui-fb68d8861fad>
- [8] "Pydantic 101: A Crash Course On Everything You Need To Know | by azeeyoon | Medium." Accessed: May 08, 2025. [Online]. Available: <https://azeeyoon.medium.com/pydantic-101-a-crash-course-on-everything-you-need-to-know-4e625649509e>
- [9] "(2) FastAPI vs. Golang: A Benchmark Comparison | LinkedIn." Accessed: May 08, 2025. [Online]. Available: <https://www.linkedin.com/pulse/fastapi-vs-golang-benchmark-comparison-deepak-sabat-r8ajc/>
- [10] "Bench-marking RESTful APIs." Accessed: May 08, 2025. [Online]. Available: <https://gochronics.com/benchmark-restful-apis/>
- [11] "Poetry Was Good, Uv Is Better: An MLOps Migration Story | by Médéric Hurier (Fmind) | Medium." Accessed: May 08, 2025. [Online]. Available: <https://fmind.medium.com/poetry-was-good-uv-is-better-an-mlops-migration-story-f52bf0c6c703>
- [12] "Build an HTTP server using Elysia and Bun | Bun Examples." Accessed: May 08, 2025. [Online]. Available: <https://bun.sh/guides/ecosystem/elysia>
- [13] C. Maestas Daniel Parkes Franck Malterre Jean-Charles Lopez John Shubeck Jussi Lehtinen Suha Ondokuzmayis Vasfi Gucer, "IBM Storage Ceph Concepts and Architecture Guide."
- [14] "Time Series Quick Start - Database Manual v8.0 - MongoDB Docs." Accessed: May 09, 2025. [Online]. Available: <https://www.mongodb.com/docs/manual/core/timeseries/timeseries-quick-start/#std-label-timeseries-quick-start>
- [15] "NGINX - Zero To Hero: Your Ultimate Guide from Beginner to Advanced Mastery | by Saquib Khan | Medium." Accessed: May 09, 2025. [Online]. Available: <https://medium.com/@ksaquiib/nginx-zero-to-hero-your-ultimate-guide-from-beginner-to-advanced-mastery-57e2dad6a77a>
- [16] "Turn Your Nginx Server into a Fortress with Fail2ban and UFW." Accessed: May 09, 2025. [Online]. Available: <https://scalastic.io/en/ufw-fail2ban-nginx/>

- [17] "Generate Let's Encrypt certificate using Certbot On Linux. | by PI | Medium." Accessed: May 09, 2025. [Online]. Available: https://medium.com/@pi_45757/generate-lets-encrypt-certificate-using-certbot-on-linux-414d8adf2ff3
- [18] "Challenge Types - Let's Encrypt." Accessed: May 09, 2025. [Online]. Available: <https://letsencrypt.org/docs/challenge-types/>
- [19] "How to Build Fast API Application using Docker Compose | DigitalOcean." Accessed: May 09, 2025. [Online]. Available: <https://www.digitalocean.com/community/tutorials/create-fastapi-app-using-docker-compose>
- [20] "What Are the Steps to Integrate Docker with Kubernetes? | by Harold Finch | Medium." Accessed: May 09, 2025. [Online]. Available: <https://medium.com/@haroldfinch01/what-are-the-steps-to-integrate-docker-with-kubernetes-d29eb4c90078>
- [21] "Export a PyTorch model to ONNX — PyTorch Tutorials 2.7.0+cu126 documentation." Accessed: May 09, 2025. [Online]. Available: https://docs.pytorch.org/tutorials/beginner/onnx/export_simple_model_to_onnx_tutorial.html
- [22] "How To Create A TensorRT Engine Version 10.4.0 | by Max Melichov | Medium." Accessed: May 09, 2025. [Online]. Available: <https://medium.com/@maxme006/how-to-create-a-tensorrt-engine-version-10-4-0-ec705013da7c>
- [23] "The Ultimate Guide to MongoDB TTL Indexes." Accessed: May 09, 2025. [Online]. Available: <https://www.mydbops.com/blog/mongodb-ttl-indexes>
- [24] "OAuth vs. JWT: What Is the Difference & Using Them Together." Accessed: May 09, 2025. [Online]. Available: <https://frontegg.com/blog/oauth-vs-jwt>
- [25] "What is a Sparse Vector? How to Achieve Vector-based Hybrid Search - Qdrant." Accessed: May 18, 2025. [Online]. Available: <https://qdrant.tech/articles/sparse-vectors/>

8 Contributo delle eventuali consulenze alle attività sopra descritte

Nessuna consulenza assegnata in merito alle attività sin qui riportate.

9 Pubblicazioni scientifiche

Study of Cathode Materials for Na-Ion Batteries: Comparison Between Machine Learning Predictions and Density Functional Theory Calculations. Claudio Ronchetti, Sara Marchio, Francesco Buonocore, Simone Giusepponi, Sergio Ferlito, and Massimo Celino. Batteries 2024, 10, 431.
<https://doi.org/10.3390/batteries10120431>

10 Eventi di disseminazione

Di seguito i principali eventi di disseminazione inerenti LA 1.6:

- Sessione su invito a Nanoinnovation 2025 dal titolo "Data Platforms for Energy Materials". Hanno già accettato ENEA, Politecnico di Torino, IIT e Università di Roma Tor Vergata.

- Sessione su invito a Nanoinnovation 2024 dedicata alle piattaforme dati per il settore energetico, tenutasi il 3/12/2024; evento finale progetto IEMAP
- Webinar del 5/9/2024 intitolato: "IEMAP – Front-end walkthrough and REST API "
- Webinar del 1/10/2024 dal titolo: "Enhancing IEMAP with Semantic Search Capabilities for Files and Metadata Retrieval"
- Webinar del 1/10/2024 intitolato: "IEMAP – Front-end walkthrough and REST API"
- Webinar del 7/11/2024 e intitolato: "Vector Databases & Retrieval-Augmented Generation (RAG), a comprehensive exploration"

11 Descrizione dei risultati ottenuti (manuale d'uso piattaforma IEMAP)

Questa sezione fornisce una guida completa e dettagliata all'utilizzo della piattaforma IEMAP. È pensata per utenti con diversi livelli di competenza, dai ricercatori con conoscenze informatiche di base agli sviluppatori esperti. Per chiarezza e facilità di comprensione, verranno ampiamente utilizzati screenshot e diagrammi. I dettagli tecnici relativi all'implementazione della piattaforma sono descritti nella Sezione 7. In questa sezione verranno descritte le modalità di accesso alla piattaforma, le funzionalità dell'interfaccia web, l'utilizzo delle API e del modulo Python, e le funzionalità avanzate come il modello AI e la query semantica (quest'ultima come già detto in precedenza, al momento, 12-05-25, non è disponibile in produzione).

11.1 Introduzione alla piattaforma IEMAP

La piattaforma IEMAP è composta principalmente da:

1. Un'interfaccia web (front-end), che costituisce l'interfaccia utente principale ed è accessibile all'indirizzo <https://iemap.enea.it>.
2. Un insieme di servizi e funzionalità gestiti dai server ENEA (back-end).

Lo scopo della piattaforma IEMAP è raccogliere i dati di esperimenti e processi svolti dai partner del progetto in un unico repository, accessibile pubblicamente per successive query e condivisione dei dati. Oltre alle funzionalità di base di gestione dei dati, sono state sviluppate le seguenti funzionalità aggiuntive:

- **Modello AI:** Predizione del potenziale redox e dell'energia di formazione a partire da dati espressi in file .CIF.
- **Analisi dati perovskite:** Applicazione per la data analysis dei dati di deposizione per un processo di deposizione di perovskite per celle fotovoltaiche.
- **Query semantica:** Sistema avanzato di query che consente ricerche semantiche utilizzando database vettoriali e modelli linguistici di grandi dimensioni (LLM).

Attualmente, quest'ultima funzionalità è limitata ai metadati dei progetti archiviati in IEMAP, ma si prevede di estenderla ai file associati (per i dettagli tecnici si rimanda alla sezione 7). Si sottolinea inoltre che questa funzionalità è accessibile solo tramite il modulo Python e non tramite l'interfaccia web, ed è attualmente in fase di test e non ancora in produzione.

Le funzionalità aggiuntive sopra elencate sono esempi di come sia possibile rendere accessibili funzionalità avanzate alla comunità scientifica, richiedendo un background informatico/tecnico minimo agli utenti della piattaforma.

Di seguito un diagramma sintetico dell'intera piattaforma IEMAP.

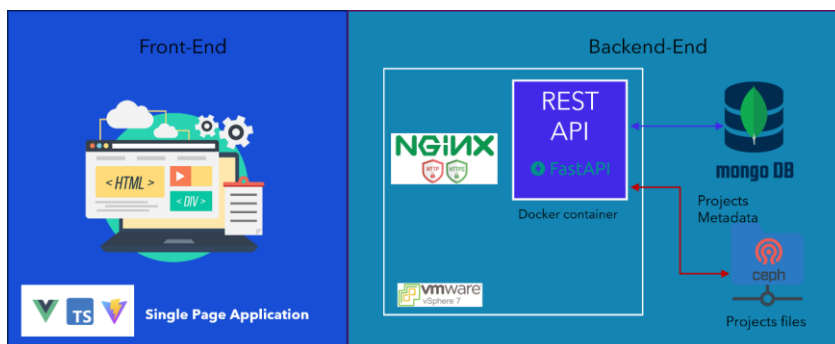


Figura 9. Schema tecnico sintetico relativo all'intera piattaforma IEMAP

Dall'immagine risulta ben chiaro la separazione tra il front-end, ovvero l'applicazione web disponibile al link <https://iemap.enea.it> ed i back-end con i servizi ivi contenuti.

Il front-end consente di:

- Registrare un account; necessario per inserire i metadati di progetto, account che deve essere validato tramite la procedura già descritta nella sezione 7 che sarà meglio dettagliata più avanti
- Recuperare/modificare/cancellare i dati già inseriti, ad esempio aggiungendo un file ai metadati di progetto
- Cercare un progetto/materiale e scaricare eventuali file associati

In back-end consente di:

- Esporre ad un link pubblico in maniera sicura il sito web della piattaforma IEMAP
- Esporre in maniera sicura le REST API usate dal front-end e dal modulo Python disponibile su Pypi ()
- Eseguire upload/download file; i metadati di tali file compreso l'hash del file stesso sono memorizzati in MongoDB, mentre i file sono memorizzati su file system remoto Ceph
- Memorizzare tutti i metadati su database MongoDB, sia quelli relativi ai progetti caricati dai partner sia i dati sperimentali del processo di deposizione Perovskite

11.2 Gestione Dati

Questa sezione descrive come la piattaforma IEMAP gestisce i dati degli esperimenti e dei processi, con particolare attenzione all'inserimento, alla memorizzazione e all'organizzazione dei dati stessi.

11.2.1 Inserimento dei metadati

Per semplificare e velocizzare l'inserimento dei metadati di progetto (i parametri di base comuni a più processi/esperimenti), si utilizza un file Excel che funge da template base. Questo file è composto da cinque fogli principali e da un foglio "how_to" con le istruzioni per la compilazione. I fogli principali sono:

1. Project
2. Process
3. Materials
4. Parameters
5. Properties

I metadati inseriti in questi fogli sono memorizzati nel database MongoDB e sono i parametri "interrogabili" della piattaforma. Altri dati di progetto (ad es., file PDF, immagini) possono essere associati al progetto dopo che i metadati sono stati inseriti. L'inserimento dei metadati è consentito solo agli utenti partner con un account validato. Da sottolineare che in tale contesto, il termine **"Project"** serve a descrivere il nome di progetto e **ad esso è associabile uno ed un solo processo o esperimento**.

Il foglio "project" consente di inserire il nome di progetto/attività e relativa label, attualmente vi sono i seguenti valori inseribili tramite controllo a discesa nella corrispondente cella Excel:

Materials for Batteries → MB
Materials for Electrolyzers → ME
Materials for Perovskite → MPV
Materials for photo-rechargeable devices → MPD
Materials for Photovoltaics → MPH

Le celle hanno una regola di validazione con una serie di valori predefiniti e presenti nel foglio how-to, per il campo project i valori utilizzabili devono rimanere quelli predefiniti ma più in generale è possibile estendere la lista di valori validi come descritto nella sezione 11.2.4 più avanti.

Il foglio "process" sta a identificare la simulazione o esperimento appartenente ad una delle categorie definite in "project". Se si voglio caricare in IEMAP, ad esempio, 10 processi occorre compilare 10 foglio Excel ognuno specifico per l'esperimento svolto con i metadati che lo contraddistinguono, ovvero vi è una corrispondenza 1 a 1 tra "project" e "process". Tramite il foglio "process" è possibile inserire:

method → nome del processo, ad esempio DFT
agent.name → strumento o software (SW)
agent.version → versione dello strumento/SW
isExperiment → vero se si tratta di un esperimento

Tramite il foglio Excel "material" si inserisce la formula del materiale oggetto dell'attività, in tale foglio le celle non hanno alcuna regola di validazione.

I fogli "parameters" e "properties" consentono di inserire i parametri e/o le proprietà che vanno inserite come triple formate da nome, valore, unità (ad esempio "temperatura", "20", "gradi centigradi"). Le celle in tali fogli hanno delle regole di controllo di validità da una lista precompilata di materiali o parametri definiti nel foglio how-to.

11.2.2 Inserimento dei file associati ad un dato progetto

I file associati a un progetto (che rappresenta un singolo processo o esperimento) vengono gestiti come segue:

1. I metadati del progetto definiscono un documento nella collezione MongoDB.
2. Per ogni file da associare al progetto:
 - a. Viene calcolato un hash univoco del file. Un hash è un codice alfanumerico di lunghezza fissa che identifica univocamente il contenuto del file. Questo garantisce che i file duplicati vengano identificati.
 - b. L'hash viene memorizzato nel campo "files" (di tipo array) del documento MongoDB.
 - c. Il file stesso viene memorizzato sul file system remoto Ceph con il nome ` - d. Alcune informazioni sintetiche sul file (es., dimensioni in MB) sono memorizzate anche nella collezione MongoDB.

Utilizzare un approccio basato su **hash univoci** e **CephFS** consente di:

- garantire **unicità e integrità** dei file
- ottenere una **gestione scalabile e sicura** dello storage
- minimizzare la duplicazione
- mantenere un MongoDB leggero e indicizzabile solo con i metadati

Questo modello è particolarmente adatto per piattaforme scientifiche distribuite come **IEMAP**, dove i file (immagini, dati sperimentali, simulazioni, etc.) possono essere condivisi, referenziati e analizzati in maniera efficiente.

11.2.3 Formato di dati supportati

In merito ai metadati si è già dettagliato quanto necessario in precedenza, mentre per i file che si possono caricare sulla piattaforma sono supportati/consentiti i seguenti formati con un limite massimo di dimensione del singolo file di 1Gb:

csv, json, .doc, docx, .pdf, .xls, .xlsx, .png, .jpg, .in, .out, .dat, .cif.

11.2.4 Linee guida per la preparazione e il caricamento dei dati

Per garantire la coerenza e l'accuratezza dei dati, si prega di utilizzare il file `template_metadati.xlsx` disponibile sulla piattaforma, questo viene convertito in JSON in locale sul front-end prima di essere caricato/salvato sul back-end e quindi su DB, in alternativa è

possibile procedere in maniera programmatica utilizzando il modulo Python iemap-mi in tal caso è stato definito un tipo Pydantic per la semplice compilazione e verifica dei metadati, in questo caso si può procedere come segue:

Qualora si utilizzasse il file Excel per inserire i metadati va tenuto presente che ogni singolo file Excel consente di **inserire un singolo progetto con associato uno ed un solo processo (sia esso sperimentale o meno)**.

Alcuni campi del foglio Excel sono “precompilati” con i valori definiti nel foglio “how-to”, qualora si volesse aggiungere un valore non presente da quelli previsti si può procedere come segue:

1. Aprire il file metadata_template.xlsx (disponibile al link https://iemap.enea.it/metadata_template.xlsx)
2. Selezionare, ad esempio, il foglio “process” e quindi la cella per la quale attivare/modificare la convalida dati (è possibile selezionare più celle, in figura solo una, ma si può selezionare quella indicata ed N al di sotto di essa)

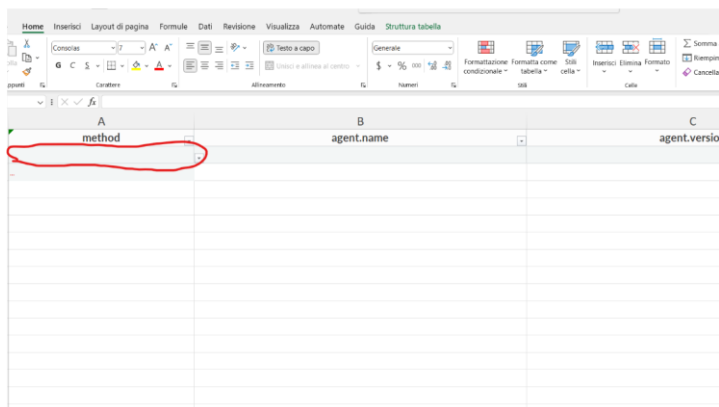


Figura 10. Template Excel inserimento metadati

3. Selezionare Dati e Convalida dati

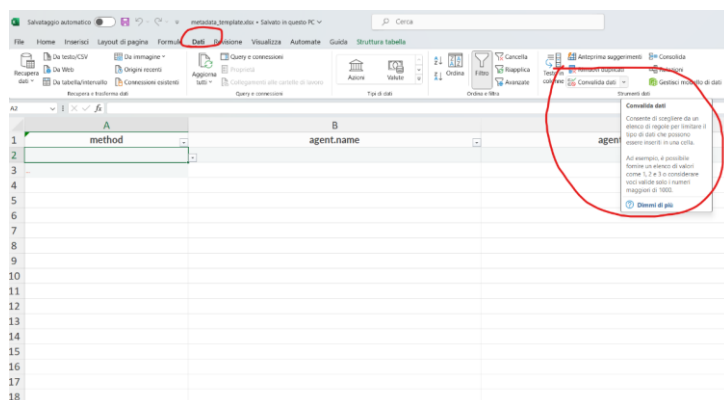


Figura 11. Template Excel inserimento metadati - convalida dati

4. Si aprirà la finestra di dialogo seguente, dove selezionando la freccia evidenziata in rosso, si verrà re-indirizzati al foglio "how to"

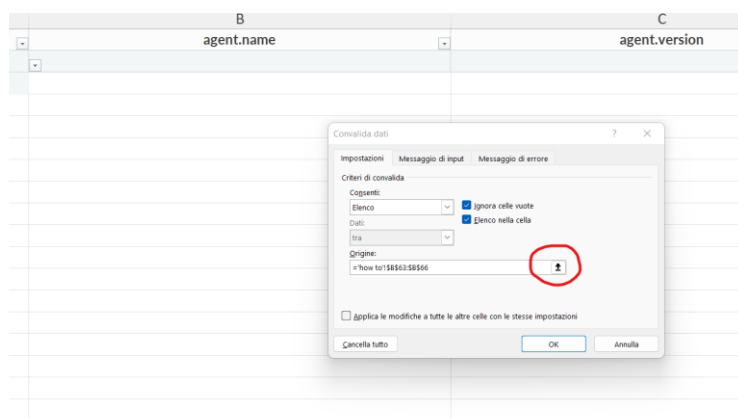


Figura 12. Template Excel inserimento metadati - convalida dati (valori)

5. Qui occorre selezionare le voci che vogliono abilitare o, meglio, considerare come valori "validi"

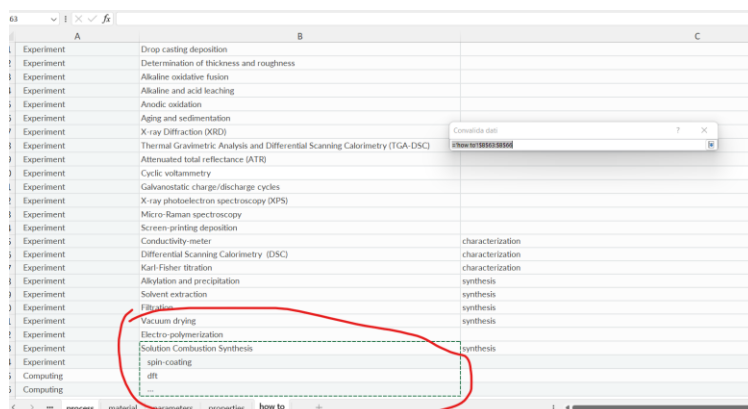


Figura 13. . Template Excel inserimento metadati - convalida dati aggiungi nuovi

6. A questo punto basta selezionare tutte le voci valide ovvero ='how to'!\$B\$24:\$B\$65
7. Dopodiché si potrà inserire il processo desiderato

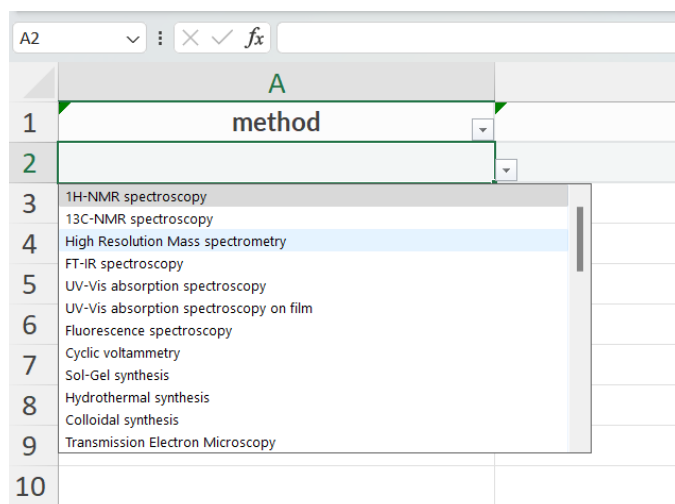


Figura 14. Template Excel inserimento metadati - convalida dati attiva

8. Qualora un processo, proprietà o parametro non fosse disponibile nella lista messa a disposizione, basta aggiungere una voce in "how to" e ripetere i passi su elencati
9. Se la validazione dei dati la si vuole estendere a celle in precedenza non incluse, basta portare il cursore in corrispondenza dell'ultima cella per la quale la validazione è attiva

(angolo in basso a destra, appare un segno “+”, basterà trascinare verso il basso per estendere la validazione alle altre celle)

11.2.5 Politiche di conservazione e sicurezza dei dati

La piattaforma IEMAP adotta rigorose politiche per garantire la sicurezza e l'integrità dei dati degli utenti e dei progetti. Di seguito sono descritte le principali misure adottate:

- Backup dei dati: Vengono eseguiti backup automatici e regolari del database MongoDB tramite OPS Manager. Questo sistema consente di ripristinare i dati in caso di guasti hardware, errori software o altri eventi imprevisti, minimizzando il rischio di perdita di informazioni.
- Riservatezza dei dati: La piattaforma IEMAP è progettata per proteggere la riservatezza dei dati degli utenti. Ad esempio, le REST API restituiscono i campi contenenti indirizzi e-mail oscurati tramite asterischi per prevenire la divulgazione non autorizzata di informazioni personali.
- Gestione degli account utente: La creazione e la gestione degli account utente sono gestite tramite il modulo FastAPIUsers. Questo modulo implementa policy di sicurezza predefinite per la gestione delle password, l'autenticazione e l'autorizzazione degli utenti. Gli utenti possono richiedere la rimozione del proprio account tramite un apposito form web. In tal caso, i dati associati all'account verranno eliminati dal database, nel rispetto delle policy in vigore.
- Sicurezza dell'infrastruttura: L'infrastruttura su cui è ospitata la piattaforma IEMAP è protetta da firewall e sistemi di rilevamento delle intrusioni. Vengono adottate le migliori pratiche di sicurezza per prevenire accessi non autorizzati e proteggere i dati da minacce esterne.
- Aggiornamenti di sicurezza: La piattaforma IEMAP e le sue componenti software vengono regolarmente aggiornate (nei limiti possibili e compatibilmente ad altri impegni progettuali) con le patch di sicurezza più recenti per correggere eventuali vulnerabilità.

11.3 Guida interfaccia utente: front-end

L'home page del sito web (front-end) di IEMAP, <https://iemap.enea.it/>, si presenta come segue:

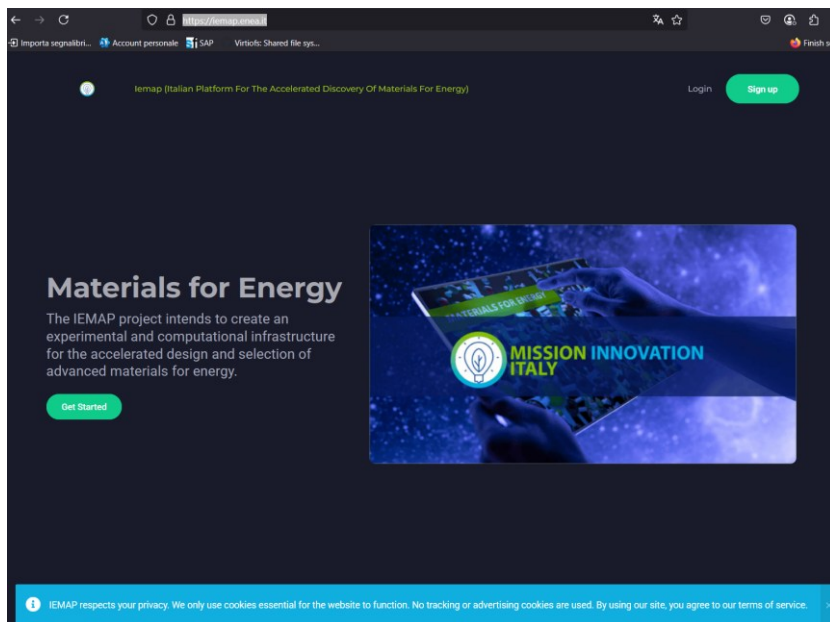


Figura 15. Home page sito web (front-end) IEMAP

Tutti i maggiori browser più recenti sono supportati e non è stato segnalato alcun problema in tal senso da parte degli utenti. E' supportato sia il tema scuro (in figura) che il tema chiaro (vedi figura seguente slider in alto alla pagina), e sono fornite in tale pagina le informazioni di base con le finalità del progetto oltre ai link per il **login**, nel caso di utenti già registrati, ed il **signup** qualora si voglia creare un account IEMAP. Nella home page si trova anche il link al modulo Python iemap-mi per l'accesso programmatico alla piattaforma.

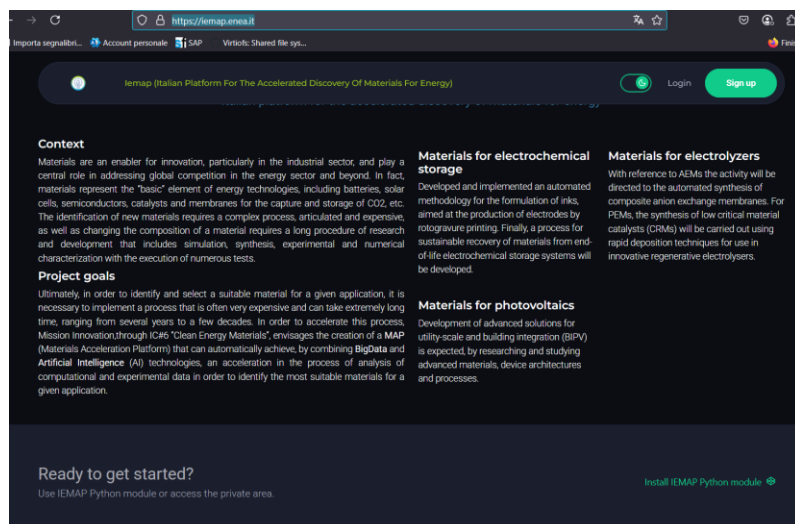


Figura 16. Home page front-end IEMAP: sezione inferiore

Il pulsante **Get Started**, consente di andare alla seguente pagina:

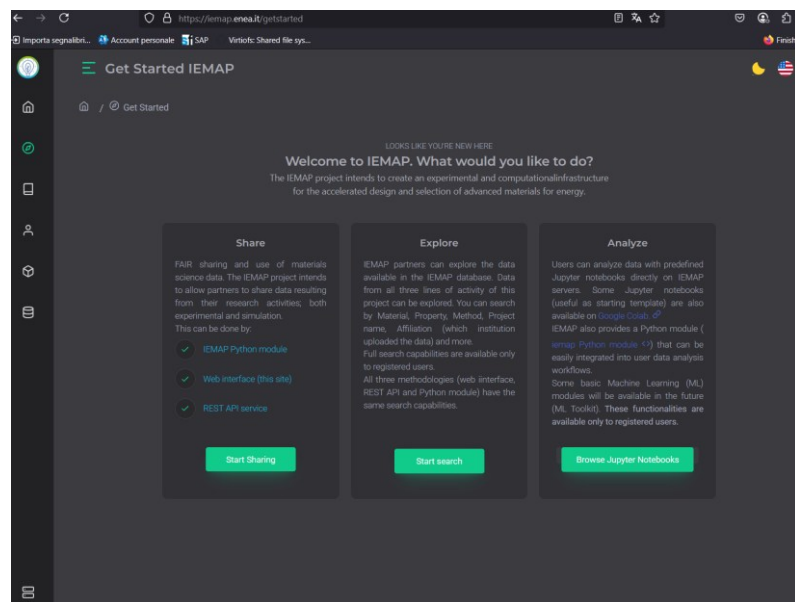
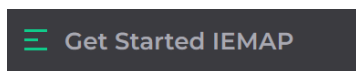


Figura 17. Front-end IEMAP: Get Started

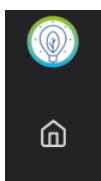
Tale pagina riporta altri dettagli in merito ai servizi forniti dalla piattaforma, inoltre quella qui descritta costituisce il layout di base del front-end dove si possono identificare i seguenti elementi:



“breadcrumb” indica in quale sezione del sito ci si trova



Selettore lingua, default Inglese, il sito è progettato in modo da consentire la gestione dinamica di più lingue ma solo pochissime sezioni sono indicate anche in Italiano



Re-indirizza alla Home Page



Link alla pagina Get Started, quella al momento selezionata, da cui il colore verde



Link alla pagina di documentazione



Link alla pagina di Login



Link alla pagina dei progetti



Link alla pagina di ricerca dei dati presenti in IEMAP



Link alla pagina di descrizione dell'architettura IEMAP

Ora che dovrebbe essere chiaro il layout e le modalità di navigazione nel sito mostrerò di seguito prima le pagine che non richiedono autenticazione per poi passare ad illustrare la fase di creazione e validazione account ed infine quella di inserimento dati e/o file di progetto.

La pagina relativa alla documentazione è la seguente (<https://iemap.enea.it/documentation>)

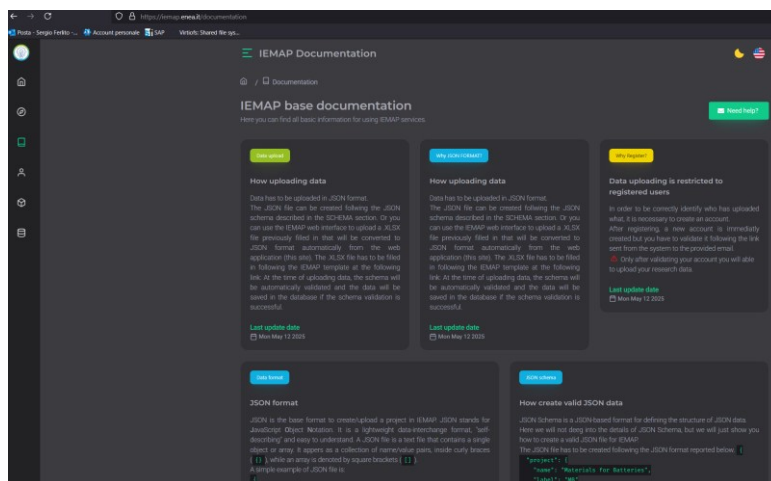


Figura 18. Front-end IEMAP: pagina documentazione

La pagina di Login è mostrata di seguito:

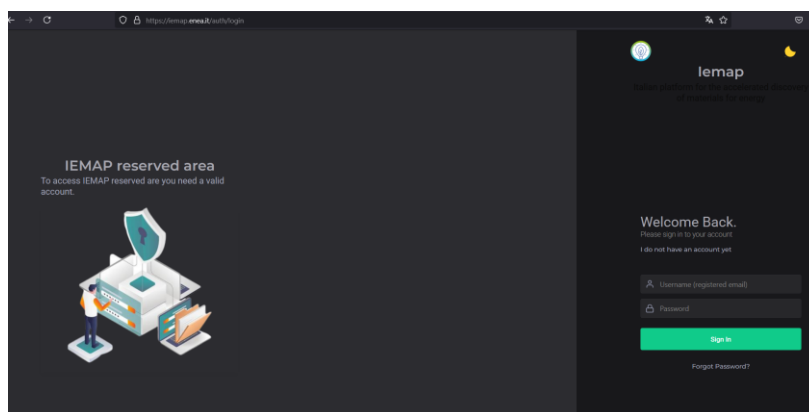


Figura 19. Front-end IEMAP: pagina login

Da qui è possibile:

- Effettuare il login qualora si disponga già di un account validato
- Recuperare la password (il sistema invia un link per il reset)
- Effettuare la registrazione (cliccando sulla scritta "I do not have an account yet")

La registrazione di un nuovo account avviene tramite la seguente procedura di seguito spiegata nel dettaglio:

1. **Sign up** tramite e-mail, affiliazione e password (minimo 8 caratteri). L'affiliazione è una qualsiasi tra le opzioni valide (elenco a discesa). Viene anche fatto un controllo sulla mail che deve essere formalmente valida
2. Subito dopo appare la pagina seguente

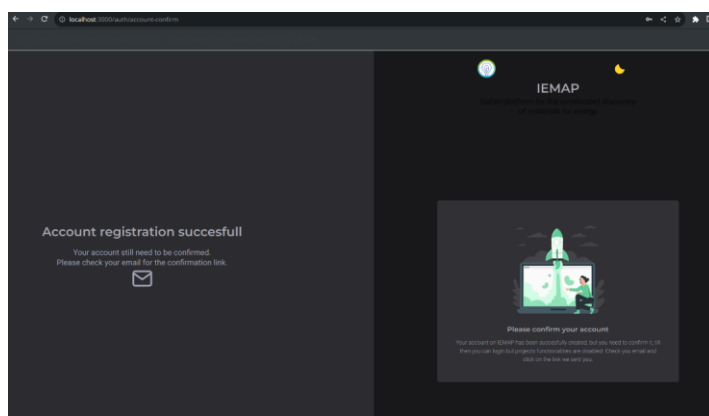


Figura 20. Front-end IEMAP: pagina validazione account

3. A questo punto il server invia una mail all'indirizzo fornito al passo 1 che appare come segue

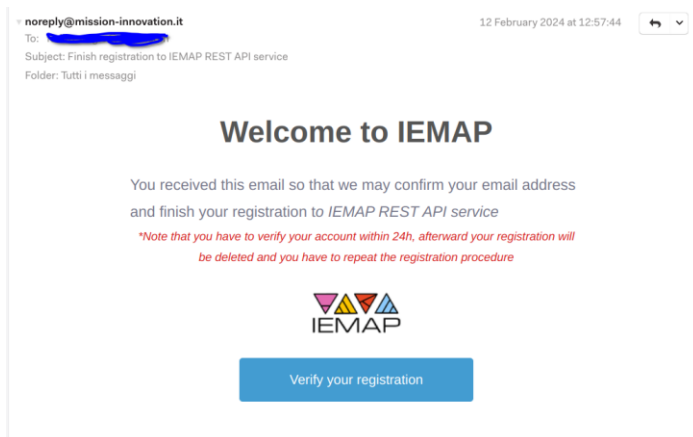


Figura 21. Esempio e-mail per attivazione account IEMAP

4. Facendo click su **Verify your registration** si apre la pagina web seguente

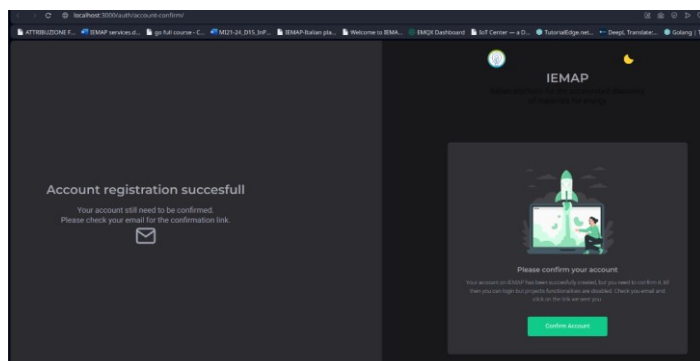


Figura 22. Front-end IEMAP: conferma attivazione account

5. molto simile a quella del punto 2, salvo la presenza del pulsante **Confirm Account**
6. Con un click su tale pulsante si invia al server un JWT per l'attivazione dell'account
7. Compare un popup che informa dell'attivazione dell'account e si è indirizzati alla pagina di login
8. Solamente dopo l'attivazione dell'account è possibile accedere all'area riservata

La pagina di ricerca dei dati in IEMAP è la seguente:

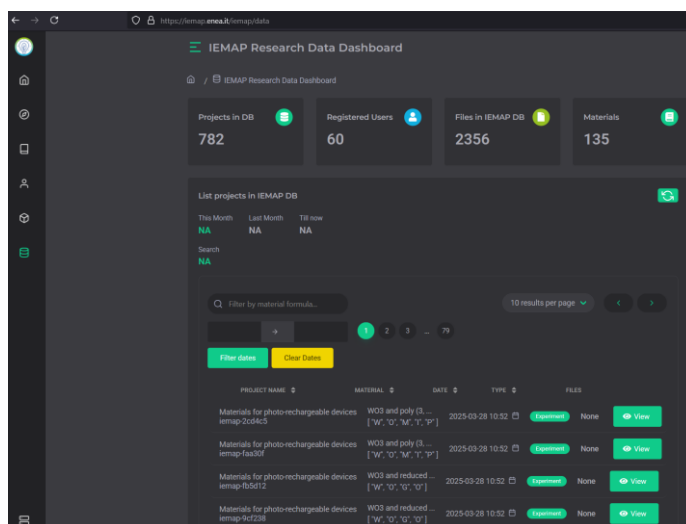


Figura 23. Front-end IEMAP: pagina ricerca progetti

Non richiede login essendo aperta a tutti, mostra alcuni dati statistici di base nella prima riga in alto, ovvero:

- Progetti inseriti in IEMAP
- Utenti registrati
- File presenti in IEMAP
- Materiali distinti inseriti in IEMAP

Inoltre, consente di visualizzare i progetti tramite una tabella con paginazione dove sono mostrate le informazioni di base (per i dettagli basta fare click su **View**), tale tabella è filtrabile per data inizio e fine, come evidenziato di seguito:

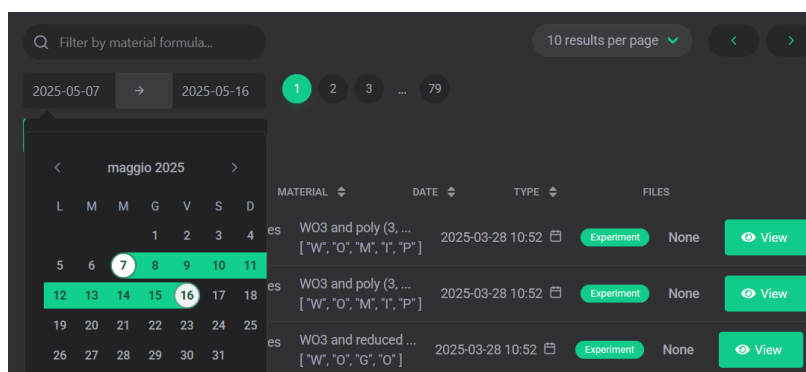


Figura 24. Front-end IEMAP: esempio filtro progetti per range date

Oppure è possibile filtrare per materiale (vedi di seguito), man mano che si inserisce la formula del materiale i dati mostrati sono filtrati.

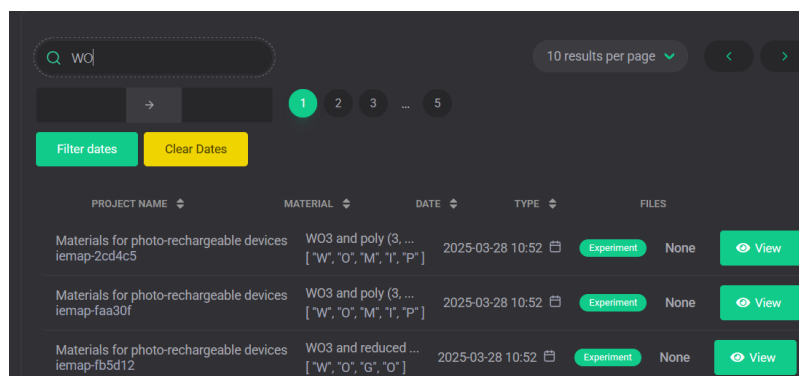


Figura 25. Front-end IEMAP: esempio filtro progetti per Materiale

Questi sono i soli criteri di ricerca/filtro disponibili da front-end, tramite modulo Python iemap-mi è possibile una ricerca semantica avanzata (dettagliata nella sezione 11.7 più avanti).

La pagina per l'inserimento dei dati di progetto è dettagliata di seguito (si è effettuato il login)

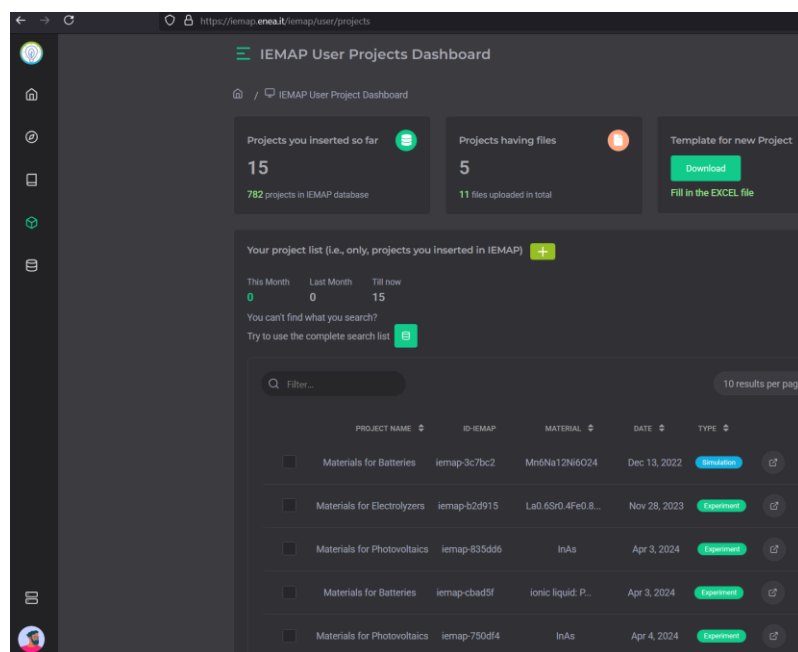


Figura 26. Front-end IEMAP: pagina progetti utente

Da notare l'avatar in basso a sinistra a testimonianza che si è effettuato il login, cliccando su tale avatar si apre un menu che consente di:

- Visualizzare le informazioni del profilo registrato
- Cambiare password ed eventualmente richiedere la cancellazione dell'account
- Effettuare il logout

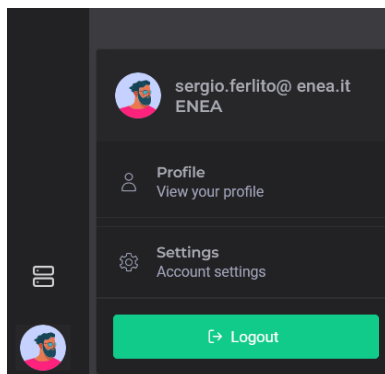


Figura 27. Dettaglio Front-end IEMAP per utente che ha effettuato login

Tale pagina (<https://iemap.enea.it/iemap/user/projects>) mostra i propri progetti inseriti, consente di scaricare il file Excel di template (https://iemap.enea.it/metadata_template.xlsx) da compilare per l'inserimento dei dati di progetto. Tale file Excel compilato va caricato tramite il pulsante **Browse metadata file** posto in basso alla pagina.

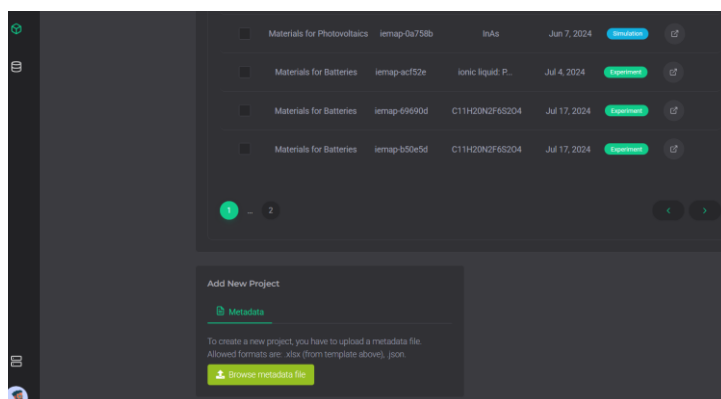


Figura 28. Front-end IEMAP: Upload metadati tramite file Excel o JSON

Una volta caricato il file Excel compilato (accetta anche un file JSON ma dallo schema valido), lo stesso controllo mostra una anteprima dei dati JSON estratti, a questo punto basta fare click sul pulsante upload per l'effettiva creazione del progetto, pertanto la creazione di un nuovo progetto, per la sola sezione dei metadati, avviene tramite due step:

1. Download file Excel di template e compilazione con i dati di progetto
2. Upload del JSON estratto dal file Excel

La figura seguente evidenzia come una volta caricato il file Excel compilato oppure un JSON valido si ha un'anteprima dei dati estratti così come verranno inseriti nel DB MongoDB.

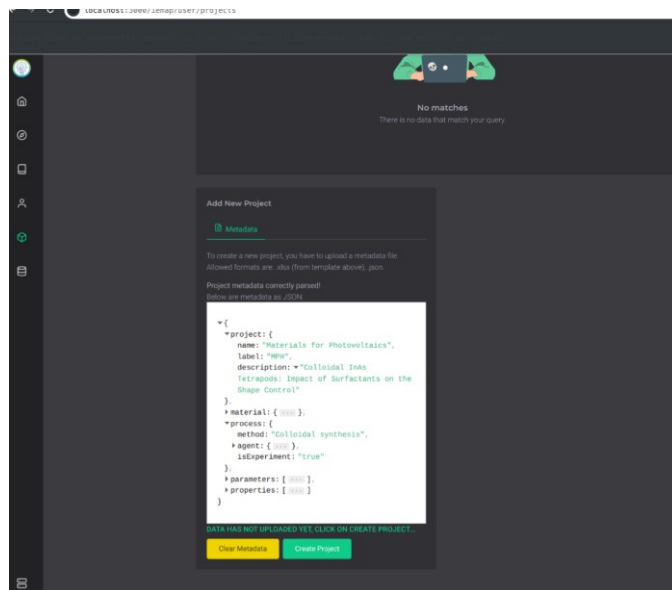


Figura 29. Front-end IEMAP: metadati caricati

Come appare in figura sopra, è possibile azzerare i metadati, pulsante **Clear Metadata**, oppure creare il progetto con il pulsante **Create Project** (solo dopo aver fatto click su tale pulsante i dati sono effettivamente inviati al server per la memorizzazione su DB).

Una volta creato il progetto si è re-indirizzati in automatico alla pagina di dettagli del progetto appena creato, qui è possibile caricare uno o più file associati al progetto stesso tramite il pulsante in alto a destra nella pagina alla voce **ADD FILES**, vedi figura seguente.

Da notare che la pagina di progetto mostra con un elemento grafico a forma di "tachimetro" il numero di file associati al progetto correntemente visualizzato e tutti i dettagli del progetto stesso, nonché (non mostrato in figura ma presente nella parte inferiore della pagina di progetto) i file correntemente associati con la possibilità di scaricarli e/o cancellarli.

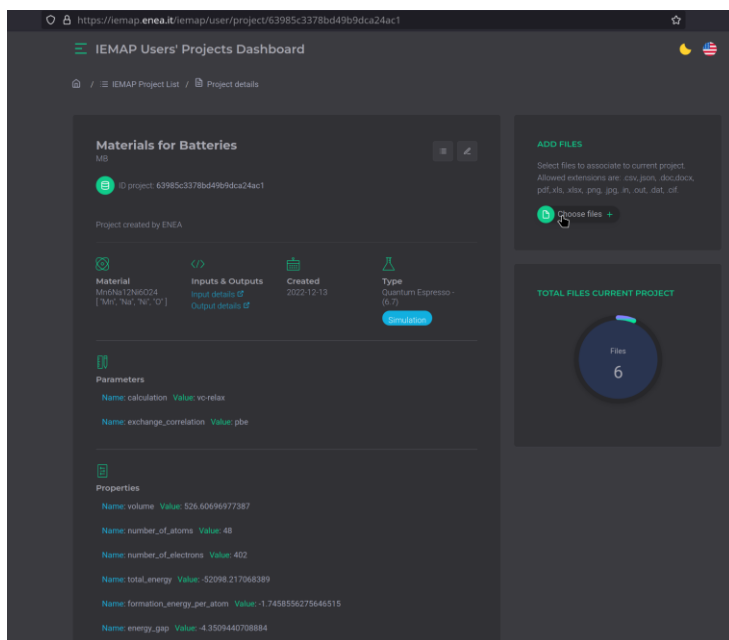


Figura 30. Front-end IEMAP: pagina progetto utente con file associati

11.4 Utilizzo REST API

Il servizio che consente di accedere ai dati in IEMAP è un servizio REST API (Representational State Transfer – Application Programming Interface). In breve, una REST API è un'interfaccia che permette a utenti o applicazioni di interagire con un sistema informatico attraverso richieste HTTP, usando un insieme limitato e standardizzato di operazioni.

Nel contesto REST:

- Una **risorsa** è qualsiasi entità gestita dal sistema, ad esempio un progetto, un esperimento, un materiale. Ogni risorsa è accessibile tramite un identificatore univoco.
- Un **URI** (*Uniform Resource Identifier*) è l'indirizzo univoco che identifica una risorsa sul web. Ad esempio, `/materials/abc123` è un URI che potrebbe identificare un materiale specifico all'interno del database IEMAP.
- Il protocollo REST è **stateless**, ovvero ogni richiesta HTTP inviata al server deve contenere tutte le informazioni necessarie per essere compresa e gestita: il server **non mantiene memoria dello stato della sessione** tra una richiesta e l'altra.

Le operazioni disponibili sulle risorse sono mappate sui metodi HTTP standard:

- GET → recupera una risorsa,
- POST → crea una nuova risorsa,
- PUT o PATCH → modifica una risorsa esistente,
- DELETE → rimuove una risorsa.

La seguente figura fornisce una spiegazione sintetica di quanto detto:

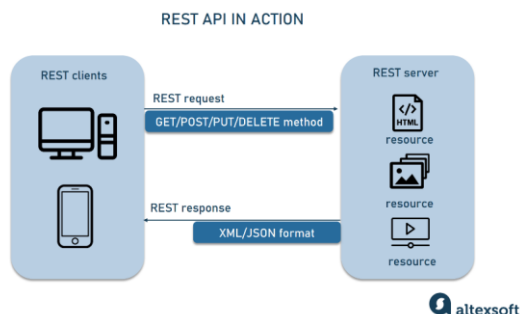


Figura 31. Schema funzionamento REST API

Autenticazione con JWT

Per accedere ad alcune funzionalità della piattaforma, è necessario essere autenticati. IEMAP utilizza **JWT** (*JSON Web Token*) come meccanismo di autenticazione. Quando un utente effettua il login (tramite endpoint di autenticazione), il sistema restituisce un **token JWT**, ovvero una stringa codificata che rappresenta l'identità dell'utente e i suoi permessi. Questo token deve essere incluso in ogni richiesta successiva, nell'**header HTTP Authorization**, secondo la sintassi:

Authorization: Bearer <token>

Qui:

- Authorization è il nome dell'header HTTP,
- Bearer è un tipo di autorizzazione che indica che il client sta inviando un token,
- <token> è la stringa JWT ricevuta dopo l'autenticazione.

Il server verifica il token a ogni richiesta per assicurarsi che l'utente sia autenticato e autorizzato a eseguire l'azione richiesta. Questo approccio **mantiene la natura stateless del protocollo**: non serve mantenere una sessione lato server, perché ogni richiesta è autosufficiente.

La documentazione tecnica delle API è accessibile tramite un'interfaccia interattiva generata automaticamente (Swagger UI), che consente:

- di esplorare gli endpoint disponibili,
- di vedere i parametri richiesti e i formati accettati,
- di testare direttamente le API con un token JWT fornito.

Per gli utenti più esperti è disponibile anche il **modulo Python iemap-mi** (su PyPI), che facilita la connessione e l'uso delle API all'interno di notebook o script Python.

Di seguito si mostreranno alcuni dettagli su come utilizzare le REST API tramite la semplice utility CURL e con codice Java.

curl (abbreviazione di *Client for URLs*) è un **comando da terminale** usato per **inviare richieste HTTP** (e altri protocolli) a server web.

Viene comunemente usato per:

- testare API REST,
- scaricare file o dati da un URL,
- inviare dati in formato JSON, form o multipart,
- effettuare autenticazioni (con token, credenziali, ecc.).

È disponibile **di default su Linux e macOS**, e installabile su Windows (via Git Bash, WSL o direttamente).

Ad esempio, per effettuare il login tramite REST API si può utilizzare:

```
curl -X POST https://iemap.enea.it/rest/auth/jwt/login \
-H "Content-Type: application/x-www-form-urlencoded" \
-d "username=tuo_username&password=tua_password"
```

La risposta sarà simile a:

```
{
  "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
}
```

Per accedere ad un endpoint protetto si utilizzerà il token acquisito tramite l'endpoint per il login tramite:

```
curl -X GET https://iemap.enea.it/rest/api/materials/iemap-2cd4c5 \
-H "Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9..."
```

Si ricorda che la documentazione Swagger per le REST API è disponibile alla URL

<https://iemap.enea.it/rest/docs>.

Di seguito come appare la documentazione generata automaticamente, da notare che gli endpoint che richiedono una autenticazione sono contrassegnati da un lucchetto e che è possibile tramite l'interfaccia stessa utilizzare curl per eseguire/invocare l'endpoint dalla stessa interfaccia della documentazione (pulsante **Try it out**).

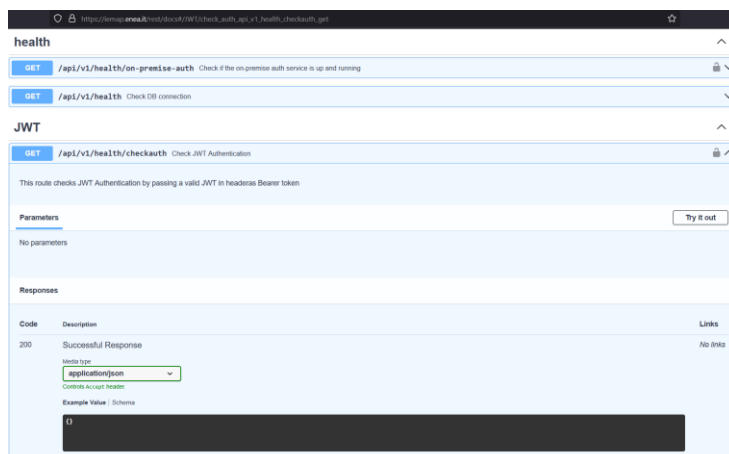


Figura 32. Pagina Swagger di documentazione e test REST API IEMAP

È possibile utilizzare un qualsivoglia linguaggio per utilizzare gli endpoint REST API, ad esempio in JAVA:

```
IEMAPClient/
├── ApiAuthenticator.java
├── ProjectAdder.java
├── lib/
│   ├── jackson-core-2.15.2.jar
│   ├── jackson-databind-2.15.2.jar
│   └── jackson-annotations-2.15.2.jar
```

<https://repo1.maven.org/maven2/com/fasterxml/jackson/core/>

ApiAuthenticator.java

```
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.http.*;
import java.net.URLEncoder;
import java.nio.charset.StandardCharsets;
import java.util.Map;
import com.fasterxml.jackson.core.type.TypeReference;
import com.fasterxml.jackson.databind.ObjectMapper;

public class ApiAuthenticator {
    private static final String AUTH_URL = "https://iemap.enea.it/rest/api/v1/auth/jwt/login";
```

```

    public static String getJwtToken(String username, String password) throws IOException,
    InterruptedException, URISyntaxException {
        HttpClient client = HttpClient.newHttpClient();
        String formData = String.format("username=%s&password=%s",
            URLEncoder.encode(username, StandardCharsets.UTF_8),
            URLEncoder.encode(password, StandardCharsets.UTF_8));

        HttpRequest request = HttpRequest.newBuilder()
            .uri(new URI(AUTH_URL))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString(formData))
            .build();

        HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());

        if (response.statusCode() == 200) {
            ObjectMapper objectMapper = new ObjectMapper();
            Map<String, String> json = objectMapper.readValue(response.body(), new
            TypeReference<>() {});
            return json.get("access_token");
        } else {
            System.err.println("Login fallito. Codice: " + response.statusCode());
            return null;
        }
    }
}

```

ProjectAdder.java

```

import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.net.http.*;
import java.util.HashMap;
import java.util.Map;
import com.fasterxml.jackson.databind.ObjectMapper;

public class ProjectAdder {
    private static final String ADD_PROJECT_URL = "https://iemap.enea.it/rest/api/v1/project/add";

    public static void main(String[] args) {
        String username = "your_username";
        String password = "your_password";

        try {
            String token = ApiAuthenticator.getJwtToken(username, password);
            if (token == null) return;

            HttpClient client = HttpClient.newHttpClient();

```

```
Map<String, Object> project = new HashMap<>();
project.put("name", "Demo Project Java");
project.put("label", "JAVA01");
project.put("description", "A test project created from Java");
```

```
Map<String, String> provenance = new HashMap<>();
provenance.put("email", "tuo_email@example.com");
provenance.put("affiliation", "Tua Affiliazione");
```

```
project.put("provenance", provenance);
```

```
ObjectMapper mapper = new ObjectMapper();
String json = mapper.writeValueAsString(project);
```

```
HttpRequest request = HttpRequest.newBuilder()
    .uri(new URI(ADD_PROJECT_URL))
    .header("Authorization", "Bearer " + token)
    .header("Content-Type", "application/json")
    .POST(HttpRequest.BodyPublishers.ofString(json))
    .build();
```

```
HttpResponse<String> response = client.send(request,
    HttpResponse.BodyHandlers.ofString());
System.out.println("Status: " + response.statusCode());
System.out.println("Body: " + response.body());
```

```
    } catch (IOException | InterruptedException | URISyntaxException e) {
        e.printStackTrace();
    }
}
```

Occorre compilare il codice dall'interno della cartella tramite:

```
javac -cp ".:lib/*" ApiAuthenticator.java ProjectAdder.java
```

ed infine eseguirlo con

```
java -cp ".:lib/*" ProjectAdder
```

11.5 Utilizzo modulo Python iemap-mi

Per facilitare l'interazione programmatica con la piattaforma IEMAP, è disponibile un modulo Python ufficiale denominato iemap-mi. Questo modulo fornisce un'interfaccia ad alto livello per autenticarsi, creare progetti, caricare file, effettuare interrogazioni e utilizzare funzionalità di intelligenza artificiale come la predizione del potenziale redox e dell'energia di formazione tramite modelli geoCGNN. Tale modulo consentirà anche le query semantiche, funzionalità implementata ma attualmente in fase di test e non ancora in produzione.

Installazione

È possibile installare il modulo direttamente da PyPI con:

```
pip install iemap-mi
```

Si consiglia, al fine di isolare l'ambiente di esecuzione, di usare un virtual environment. In alternativa ai classici venv o conda, è possibile usare **uv**, un gestore di ambienti Python moderno e molto performante:

```
pip install uv # solo la prima volta
```

```
uv venv .venv
```

```
source .venv/bin/activate # Linux/macOS
```

```
.venv\Scripts\activate # Windows
```

```
uv pip install iemap-mi
```

Il modulo iemap-mi espone classi e metodi asincroni che permettono di interagire con il database IEMAP. Di seguito è riportato un esempio completo di script (file example.py disponibile nella directory di installazione) che:

- autentica l'utente (con password mascherata),
- recupera statistiche,
- interroga i progetti esistenti,
- esegue predizioni AI (con file .cif),
- crea un nuovo progetto con dati validati,
- aggiunge file associati a un progetto esistente.

Tutti i dettagli, modelli Pydantic e API handler sono documentati sul sito ufficiale del modulo:
<https://iemap-mi-module.readthedocs.io>

Di seguito il codice d'esempio:

```
# This is an example script that demonstrates how to use the iemap-mi Python module.  
# If not yet install then you can install the module by running the following command:  
# pip install iemap-mi
```

```
# Import the required modules  
# asyncio is used to run the main function asynchronously  
import asyncio
```

```
# stdiomask is used to hide the password input
import stdiomask
# TypeAdapter is used to validate the project data
from pydantic import TypeAdapter
# Import the IemapMI class from the iemap_mi module
from iemap_mi import IemapMI
# typing is used to define the type hints
from typing import List
# Import the required models from the iemap_mi module
from iemap_mi.models import (IEMAPProject, Project, Material, Process, Agent,
                             Parameter, Property, FlattenedProjectBase, FlattenedProjectHashEmail,
                             FileInfo)

# Import the ProjectHandler class from the iemap_mi module to handle project data tasks
as:
# - Create a new project
# - Add a file to a project
from iemap_mi.project_handler import ProjectHandler

# Import the flatten_project_data function from the iemap_mi.utils module to easily flatten
project data for display
from iemap_mi.utils import flatten_project_data

# Import the PredictionType enumeration to use it in the get_prediction function
from iemap_mi.ai_handler import PredictionType

# to pretty print the output of JSON responses
from pprint import pprint

# Check if pandas and tqdm are available,
# and set the corresponding flags
# pandas is used to display the project data in a DataFrame (if pandas is available)
# tqdm is used to display a progress bar while fetching projects (if tqdm is available)
try:
    import pandas as pd

    PANDAS_AVAILABLE = True
except ImportError:
    PANDAS_AVAILABLE = False

try:
    from tqdm import tqdm

    TQDM_AVAILABLE = True
except ImportError:
    TQDM_AVAILABLE = False

# Define the iterate_projects function
```

```
# this function shows how to fetch projects in pages and display them in a pandas
DataFrame
# similar functions can be created to fetch other data, this is just an example
# executed later in the main function
async def iterate_projects(client: IemapMI, page_size: int = 40) -> None:
    """
        Iterates over projects in the Iemap Management Interface (IEMI) and prints them,
        or converts to a pandas DataFrame if pandas is available.

        This function fetches projects in pages, with each page containing up to `page_size`
        projects.
        It can optionally include email addresses in the output if `show_email` is True and the
        user has
        the necessary permissions.

        Parameters:
        - client (IemapMI): An authenticated instance of IemapMI used to fetch project data.
        - page_size (int): The number of projects to fetch per page. Defaults to 40.

        Returns:
        None. The function directly prints project information to the console or displays it in a
        pandas DataFrame.

        Usage Example:
        ```python
 import asyncio
 from Iemap_mi import IemapMI

 async def main():
 client = IemapMI()
 await client.authenticate(username="your_username", password="your_password")
 await iterate_projects(client, page_size=50)

 if __name__ == "__main__":
 asyncio.run(main())
 """

 page_number = 1
 all_projects: List[FlattenedProjectBase] = []
 total_projects = None # Initialize total_projects to None

 while True:
 projects_response = await client.project_handler.get_projects(page_size=page_size,
 page_number=page_number)
 if not projects_response.data:
 break

 adapter = TypeAdapter(FlattenedProjectHashEmail)
```

```

projects = [adapter.validate_python(project) for project in projects_response.data]

all_projects.extend(projects)
page_number += 1

if total_projects is None:
 total_projects = projects_response.number_docs

if TQDM_AVAILABLE:
 tqdm.write(f"Page {page_number - 1} fetched. Total projects so far:
{len(all_projects)}/{total_projects}")

if PANDAS_AVAILABLE:
 # Set the option to display all columns
 pd.set_option('display.max_columns', None)
 # Convert the projects to a pandas DataFrame
 # df = pd.DataFrame([project.model_dump() for project in all_projects])
 flat_projects = [flatten_project_data(project) for project in all_projects]
 df = pd.DataFrame(flat_projects)
 print(df)
else:
 # Print the projects as a list of dictionaries
 for project in all_projects:
 print(project.model_dump())

Define the main function
This function demonstrates how to use the IemapMI module to interact with the IEMAP
API
It includes examples of:
- Authenticating a user
- Creating a new project
- Adding a file to a project
- Fetching statistical data from the API
async def main():
 # Initialize the client
 client = IemapMI()

 # Print the module version
 IemapMI.print_version()

 # Get the prediction using AI functionalities from IEMAP Platform using a geoCGNN model
 # The geoCGNN model is inspired by the research paper published in Nature
 Communications:
 # "Crystal Graph Convolutional Neural Networks for Analyzing Materials Properties"
 # (https://www.nature.com/articles/s43246-021-00194-3).
 # This model has been further trained and optimized by ENEA on the High-Performance
 Computing (HPC)

```

```

infrastructure CRESCO (https://ict.enea.it/cresco/) to predict formation energy and
redox potential,
The prediction type can be either "formation-energy" or "redox-potential"
The prediction is based on the crystal structure provided in .cif format
The prediction is returned as a JSON response
example CIFs can be found in the "example_cif" folder

cif_file_for_energy_prediction = "/example_cif/mp-2064.cif"
cif_file_for_redox_potential = "/example_cif/mp-1003402.cif"

prediction_fe = await
client.ai_handler.get_prediction(cif_file_path=cif_file_for_energy_prediction,
 prediction_type=PredictionType.FORMATION_ENERGY)

print(f"Formation energy for file {cif_file_for_energy_prediction} predicted by AI model")
pprint(prediction_fe, indent=2)

prediction_rp = await
client.ai_handler.get_prediction(cif_file_path=cif_file_for_redox_potential,
 prediction_type=PredictionType.REDOX_POTENTIAL)

print(f"Redox potential for file {cif_file_for_redox_potential} predicted by AI model")
pprint(prediction_rp, indent=2)

Iterate over projects and print them or convert to pandas DataFrame if available
await iterate_projects(client, page_size=60)

Fetch statistics data
stats = await client.stat_handler.get_stats()
print(stats.model_dump())

query_response = await client.project_handler.query_projects(
 # project_name="Materials for Batteries",
 material_formula="C11H20N2F6S2O4",
 # isExperiment=True,
 limit=10
)

print([doc.model_dump() for doc in query_response])

Prompt for username and password
username = input("Enter your username (email address): ")
password = stdiomask.getpass(prompt="Enter your password: ")

Authenticate to get the JWT token to be used to invoke REST API endpoints
!! ATTENTION: you should register to the IEMAP platform to get your credentials !!
This credential has to be validated by email sent to the user email address
To do so, please visit: https://iemap.enea.it/auth/signup

```

```
This sets a JWT token in the client instance
await client.authenticate(username=username, password=password)

To create a new project, you need to provide the project metadata as a dictionary
(JSON-like format)
The project metadata should include the project name, label, description, material,
process, parameters,
and properties.
Parameters and properties are arrays of dictionaries containing the
name, value, and unit of each parameter/property.
Below an example of a valid project metadata dictionary is provided
#
data = {
 "project": {
 "name": "Materials for Batteries",
 "label": "MB",
 "description": "IEMAP - eco-sustainable synthesis of ionic liquids as innovative
solvents for lithium/sodium batteries"
 },
 "material": {
 "formula": "C11H20N2F6S2O4"
 },
 "process": {
 "method": "Karl-Fischer titration",
 "agent": {
 "name": "Karl-Fischer titrator Mettler Toledo",
 "version": None
 },
 "isExperiment": True
 },
 "parameters": [
 {
 "name": "time",
 "value": 20,
 "unit": "s"
 },
 {
 "name": "weight",
 "value": 0.5,
 "unit": "gr"
 }
],
 "properties": [
 {
 "name": "Moisture content",
 "value": "<2",
 "unit": "ppm"
 }
]
}
```

```

]
}

Build and validate the project payload
valid_payload_example_1 = ProjectHandler.build_project_payload(data)

if valid_payload_example_1:
 print("Payload is valid and ready to be submitted.")
else:
 print("Payload is invalid.")

Example of invalid payload
data_invalid = {"project": {
 "name": "Materials for Batteries",
 "label": "MB",
 # "description": Description is missing, this is a required field !!!
},
 # material is missing and this is a required field !!!!
 # "material": {
 # "formula": "C11H20N2F6S2O4"
 # },
 "process": {
 "method": "Karl-Fischer titration",
 "agent": {
 "name": "Karl-Fischer titrator Mettler Toledo",
 "version": None
 },
 "isExperiment": True
 }
}

Also missing are parameters and properties, which are required fields

Build and validate the project payload
as the payload is invalid, the function will return None and will
print the error message that caused the payload to be invalid.
In this case, the error message is:
#
Validation Error: The provided data is not valid.
Error in field 'project.description': Field required (type: missing)
Error in field 'material': Field required (type: missing)
Error in field 'parameters': Field required (type: missing)
Error in field 'properties': Field required (type: missing)

valid_payload_example_2 = ProjectHandler.build_project_payload(data_invalid)

if valid_payload_example_2:
 print("Payload from 'data_invalid' is valid and ready to be submitted.")
else:
 print("Payload from 'data_invalid' is invalid.")

```

```

for p in [valid_payload_example_1, valid_payload_example_2]:
 if p:
 # Create a new project
 current_proj = IEMAPProject(**p)
 print(f"Adding project: {current_proj.project.name}")
 new_project = await client.project_handler.create_project(current_proj)
 print(new_project)

Alternatively metadata can be defined using the Pydantic class CreateProjectRequest as
below

project_data = IEMAPProject(
 project=Project(
 name="Materials for Batteries",
 label="MB",
 description="IEMAP - eco-sustainable synthesis of ionic liquids as innovative solvents
for lithium/sodium batteries"
),
 material=Material(
 formula="C11H20N2F6S2O4"
),
 process=Process(
 method="Karl-Fischer titration",
 agent=Agent(
 name="Karl-Fischer titrator Mettler Toledo",
 version=None
),
 isExperiment=True
),
 parameters=[
 Parameter(
 name="time",
 value=20,
 unit="s"
),
 Parameter(
 name="weight",
 value=0.5,
 unit="gr"
)
],
 properties=[
 Property(
 name="Moisture content",
 value="<2",
 unit="ppm"
)
]
)

```

```

as previously add a new project
new_project = await client.project_handler.create_project(project_data)
print(new_project)

Now add a file to the newly created project
to do this:
1. input file name path
2. get the file name from full path
3. use method "add_file_to_project" from "project_handler" providing the id of the
newly created project (metadata only)
and the file to read
file_to_add_to_project = input(
 f"Please type in the full path of file to upload and associate to current project
{project_data.project.name}: ")
file_name = file_to_add_to_project.split("/")[-1]
print("Adding file to project...")

Add a file to the project.
To add a file to a project,
you need to provide the project ID (inserted_id) and the file path.
If you already have the project ID, you can use it directly (as string).
file_response = await client.project_handler.add_file_to_project(
 project_id=new_project.inserted_id,
 file_path=file_to_add_to_project,
 file_name=file_name
)
Check if the file was uploaded successfully
If the file was uploaded successfully, the 'uploaded' key in the response will be True
if file_response['uploaded']:
 file_info = FileInfo(**file_response)
 # Print the file information
 # The file is save onto the IEMAP FileSystem with a unique hash
 print(f"File {file_info.file_name} uploaded successfully to project (saved as
{file_info.file_hash})")
else:
 print("File upload failed.")

Finally an example of how to fetch statistics data from IEMAP DB
client.stat_handler.get_stats()

to view all functionalities consult documentation at
https://iemap-mi-module.readthedocs.io/en/latest/iemap_mi.html
Note that current module is in development and not a stable release.

Run the main function asynchronously
if __name__ == "__main__":
 asyncio.run(main())

```

### Predizione AI con geoCGNN

Tra le funzionalità più avanzate, il modulo **iemap-mi** permette di inviare file CIF contenenti la struttura cristallina di un materiale e ottenere, tramite modelli AI addestrati da ENEA su CRESCO:

- formation energy;
- il redox potential.

Questi modelli sono ispirati al paper:

Xie and Grossman, Crystal Graph Convolutional Neural Networks, Nature Communications Materials, 2021 (<https://www.nature.com/articles/s43246-021-00194-3> )

## 11.6 Utilizzo funzionalità AI: geoCGNN

Una Crystal Graph Convolutional Neural Network (CGCNN) è un tipo di rete neurale progettata per prevedere le proprietà dei materiali cristallini rappresentandoli come grafi, dove gli atomi sono nodi e i legami chimici sono archi. Il modello geoCGNN migliora ulteriormente questa architettura integrando informazioni geometriche dettagliate, come le distanze e le direzioni tra atomi, per catturare meglio la struttura tridimensionale del cristallo.

Secondo lo studio di Cheng et al. (2021) pubblicato su Communications Materials, il modello geoCGNN ha dimostrato una maggiore accuratezza nella previsione di proprietà come l'energia di formazione e il band gap rispetto ad altri modelli GNN, grazie all'incorporazione di informazioni geometriche dettagliate nella rappresentazione del grafo.

ENEA ha adattato e addestrato il modello geoCGNN utilizzando dati sperimentali per prevedere proprietà dei materiali come l'energia di formazione e il potenziale redox. Il modello è stato implementato in un container Docker dedicato, utilizzando PyTorch per l'inferenza, ed è esposto tramite un'API REST sviluppata con FastAPI. Questo servizio consente agli utenti di inviare file CIF (Crystallographic Information File) per ottenere predizioni sulle proprietà dei materiali.

Le API REST per accedere al servizio di predizione basato su geoCGNN sono documentate e disponibili all'indirizzo: <https://iemap.enea.it/ai/docs>.

Di seguito è riportato un esempio di script Python che utilizza la libreria **httpx** per autenticarsi e inviare un file CIF al servizio di predizione AI di IEMAP:

```
import httpx
import asyncio

async def predict_property(file_path: str, prediction_type: str, username: str, password: str):
 base_url = "https://iemap.enea.it/ai"
 auth_url = f"{base_url}/auth/jwt/login"
 prediction_url = f"{base_url}/api/v1/predict"
```

```
async with httpx.AsyncClient() as client:
 # Autenticazione per ottenere il token JWT
 auth_response = await client.post(auth_url, json={"username": username, "password":
password})
 auth_response.raise_for_status()
 token = auth_response.json().get("token")

 headers = {
 "Authorization": f"Bearer {token}"
 }

 # Preparazione del file da inviare
 files = {
 "file": (file_path, open(file_path, "rb"), "application/octet-stream")
 }

 # Parametri della richiesta
 params = {
 "prediction_type": prediction_type # "formation-energy" o "redox-potential"
 }

 # Invio della richiesta di predizione
 response = await client.post(prediction_url, headers=headers, files=files, params=params)
 response.raise_for_status()
 prediction = response.json()
 print(f"Predizione per {prediction_type}: {prediction}")

Esecuzione dello script
if __name__ == "__main__":
 import getpass
 file_path = input("Inserisci il percorso del file CIF: ")
 prediction_type = input("Tipo di predizione ('formation-energy' o 'redox-potential'): ")
 username = input("Nome utente: ")
 password = getpass.getpass("Password: ")
 asyncio.run(predict_property(file_path, prediction_type, username, password))
```

#### Note aggiuntive

Il file CIF deve essere conforme agli standard cristallografici per garantire una predizione accurata.

Il servizio restituisce una risposta JSON contenente la predizione della proprietà richiesta.

Per ulteriori dettagli e parametri disponibili, consulta la documentazione ufficiale delle API all'indirizzo fornito sopra.

### 11.7 Query semantica (in fase di test, non disponibile in produzione)

In questa sezione si mostrano esempi pratici di utilizzo della funzionalità di ricerca semantica e ibrida implementata tramite l'integrazione tra MongoDB, Qdrant e modelli di embedding.

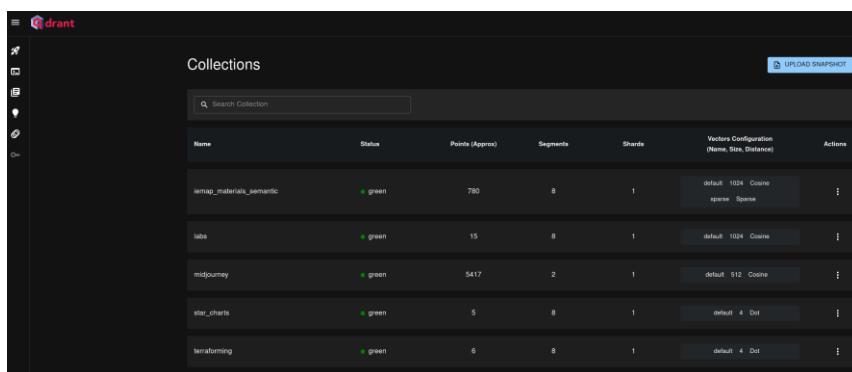
Le funzionalità saranno in un prossimo futuro esposte nel modulo iemap-mi, attualmente disponibile come pacchetto PyPI.

#### Flusso operativo

Il processo che consente la ricerca semantica in IEMAP si articola in tre fasi:

1. Estrazione dati da MongoDB e trasformazione dei campi rilevanti in un campo `combined_text`.  
Tale fase sarà implementata in automatico tramite la funzionalità di `change stream` di MongoDB in tal modo quello che attualmente ha richiesto del codice ad hoc per la sincronizzazione dei dati su MongoDB con quelli in Qdrant da avviare manualmente verrà eseguiti in produzione automaticamente, da micro-servizio in Docker dedicato, all'inerimento di nuovi progetti in MongoDB
2. Indicizzazione su Qdrant, con:
  - embedding denso (modello BAAI/bge-m3)
  - embedding sparso (modello SPLADE)
  - metadati strutturati (data, ente, progetto, ID)
3. Query semantica, ibrida o filtrata tramite API REST

Qdrant fornisce una dashboard all'indirizzo di default <http://localhost:6333/dashboard#/> che mostra quanto segue:

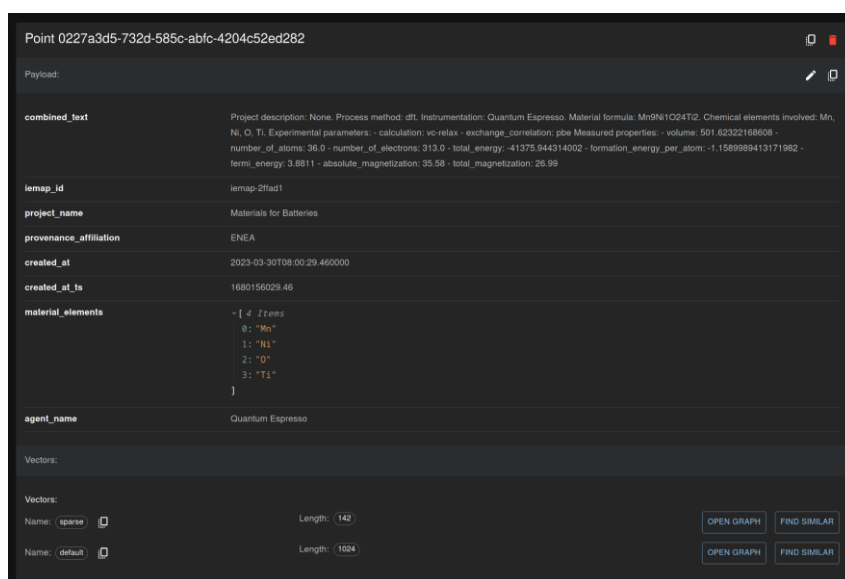


Name	Status	Points (Approx)	Segments	Shards	Vectors Configuration (Name, Size, Distance)	Actions
iemap_materials_semantic	green	750	8	1	default 1024 Cosine sparse Sparse	
lata	green	15	8	1	default 1024 Cosine	
midjourney	green	5417	2	1	default 512 Cosine	
star_charts	green	5	8	1	default 4 Dot	
terriforming	green	6	8	1	default 4 Dot	

Figura 33. Dashboard Qdrant

Si noti che la collezione **iemap\_materials\_semantic** ha due vettori **default** e **sparse** (penultima colonna) ed al momento include 780 punti.

Facendo click sulla collezione è possibile visualizzare tutti i punti presenti nella collezione, ad esempio si ha:

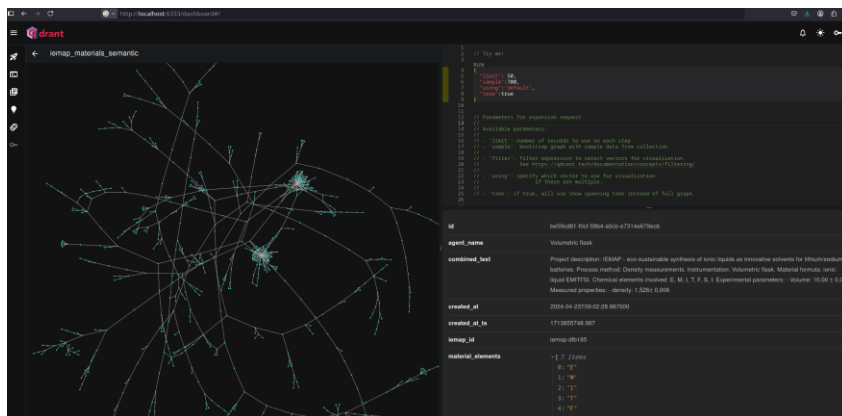


**Figura 34. Dashboard Qdrant: esempio punto inserito con payload costruito dai dati in MongoDB**

Da questo esempio si nota chiaramente il payload combinato (usato per creare il vettore denso memorizzato in default, Qdrant non mostra i vettori nella dashboard ma da notare che indica la dimensione).

E' anche possibile avere una visualizzazione interattiva ad albero per l'intero DB (o sue sezioni), basta fare click sul pulsante **GRAPH** in alto a destra nella dashboard e indicare nella console in alto a destra

```
{
 "limit": 50, # limita i risultati restituiti
 "sample": 700, # campiona un sottoinsieme dei punti totali
 "using": "default", # vettore da usare nel grafo
 "tree": true # mostra struttura ad albero
}
```



**Figura 35. Grafo ad albero dei dati IEMAP in Qdrant**

Per maggiori dettagli si consiglia il breve video esplicativo Visualizing Vector Embeddings: Qdrant's WebUI Graph Tool <https://www.youtube.com/watch?v=OzTHZ0Siulg>.

È stata sperimentata con successo tale funzionalità avanzata di interrogazione semantica e ibrida all'interno della piattaforma IEMAP. Tali funzionalità saranno esposte pubblicamente in produzione tramite API asincrone nonché tramite metodi aggiuntivi della libreria iemap-mi, simile a quanto segue:

```
query_results = await client.semantic_search(K=10, query="experiments with LiClO4 and capacitance retention")
```

La sperimentazione ha permesso di testare tre modalità di interrogazione:

#### **Ricerca semantica (solo dense embedding)**

La ricerca semantica permette di trovare contenuti semanticamente affini anche se la formulazione testuale nei documenti è diversa da quella usata nella query.  
Ad esempio, interrogando il sistema con:

"experiments with LiClO<sub>4</sub> and capacitance retention"

il sistema restituisce documenti anche se tali termini non compaiono esattamente nella forma interrogata.

Risultati ricerca semantica:

--- Semantic Query ---

ID: c0c0c4fe-31ee-5c52-b2bc-d2cf3a63aa1e, Score: 0.5000, Affiliation: ISSMC-CNR

ID: fdd70ae2-afa8-5f7f-946c-83948b52ef87, Score: 0.3333, Affiliation: CNR

ID: a7f54563-79db-5a92-9d2d-7dc8716491c1, Score: 0.2500, Affiliation: ISSMC-CNR

ID: a6b855d4-a626-5de1-9709-54a6d2f56381, Score: 0.2000, Affiliation: ISSMC-CNR

ID: 02393ad9-fc78-50c1-b03d-77d97cf1bf32, Score: 0.1667, Affiliation: ISSMC-CNR

### Query 2 – Ricerca ibrida (dense + sparse embedding)

La query ibrida combina embedding densi (per la semantica) e sparsi (per la corrispondenza precisa su keyword). Questo approccio migliora la qualità dei risultati in presenza di simboli tecnici o formule chimiche.

Esempio di query utente:

“experiments with LiClO<sub>4</sub> and capacitance retention”

Risultati query ibrida:

--- Hybrid Query ---

ID: 0e901cad-aaa2-5b26-9c0d-b1d808968f60, Score: 0.5000, Elements: ['C', 'H', 'N', 'O', 'S']

ID: c0c0c4fe-31ee-5c52-b2bc-d2cf3a63aa1e, Score: 0.5000, Elements: ['W', 'O', 'G', 'O']

ID: fdd70ae2-afa8-5f7f-946c-83948b52ef87, Score: 0.3333, Elements: ['Li', 'Fe', 'P', 'O']

ID: 77ef0459-6179-57d7-931f-abf0a24a7e87, Score: 0.3333, Elements: ['C', 'H', 'N', 'O', 'S']

ID: a7f54563-79db-5a92-9d2d-7dc8716491c1, Score: 0.2500, Elements: ['W', 'O', 'G', 'O']

Nota: La presenza del simbolo “Li” ha favorito il matching nei documenti tecnici anche se l’espressione completa “LiClO<sub>4</sub>” non era esplicitata.

### Query 3 – Ricerca semantica con filtro su metadati

È possibile combinare la semantica con filtri sui metadati, ad esempio per:

ente di provenienza (es. ISSMC-CNR)

data di creazione (es. documenti dopo il 2025-01-01)

Risultati query con filtro metadati:

--- Filtered Semantic Query (affiliation + date\_gte) ---

ID: c0c0c4fe-31ee-5c52-b2bc-d2cf3a63aa1e, Score: 0.5000, Created: 2025-03-28T10:52:37.414000,

Affiliation: ISSMC-CNR

ID: a7f54563-79db-5a92-9d2d-7dc8716491c1, Score: 0.3333, Created: 2025-03-

28T10:52:26.547000, Affiliation: ISSMC-CNR

ID: a6b855d4-a626-5de1-9709-54a6d2f56381, Score: 0.2500, Created: 2025-03-

28T10:52:07.970000, Affiliation: ISSMC-CNR

ID: 02393ad9-fc78-50c1-b03d-77d97cf1bf32, Score: 0.2000, Created: 2025-03-28T10:52:17.117000,

Affiliation: ISSMC-CNR

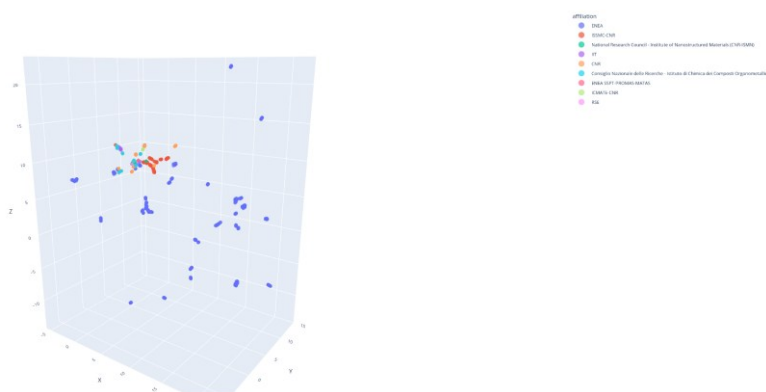
ID: dbdbc797-b950-56d0-bea4-b8f3b82df216, Score: 0.1667, Created: 2025-03-28T10:52:47.867000, Affiliation: ISSMC-CNR

È possibile estrarre i vettori densi dalla collezione Qdrant e visualizzarli tramite UMAP.

**UMAP** (Uniform Manifold Approximation and Projection) è una tecnica di riduzione dimensionale utile per visualizzare dati ad alta dimensione, come gli embedding densi estratti da Qdrant, in uno spazio 2D o 3D. Applicato ai documenti scientifici della piattaforma IEMAP, consente di esplorare similarità semantiche tra contenuti.

Da un sottoinsieme di 500 punti si ha:

3D UMAP of Qdrant Embeddings



**Figura 36. Grafico 3D proiezione UMAP vettori densi dei dati IEMAP in Qdrant**

Il grafico 3D generato mostra:

- Cluster distinti: documenti con affiliazioni simili (es. ENEA, ISSMC-CNR) tendono a raggrupparsi, segnalando coerenza semantica tra contenuti. Alcuni gruppi di punti (es. a sinistra e al centro) si addensano e mostrano colori coerenti, ad es. molti punti ENEA in blu e CNR o ISSMC-CNR in altri colori tendono a posizionarsi vicini nello spazio semantico. Questo è un segno che l'embedding BGE-m3 ha catturato correttamente similitudini concettuali tra documenti provenienti dallo stesso ente o con descrizioni tecniche simili.
- Punti isolati: alcuni documenti ENEA sono distanti, possibile segno di outlier o dati da verificare (es. testo scarso, affiliazione mancante). Possibili cause:
  - Documenti poco informativi o molto brevi.
  - Terminologia inconsistente.
  - Errori nei metadati (es. affiliazione mancante o errata).
  - Esperimenti molto diversi per natura.

- Sovrapposizione tra enti: affiliazioni vicine (es. ICMATE-CNR e CNR) suggeriscono collaborazioni o temi comuni.
  - Collaborazioni frequenti.
  - Uso di terminologie o protocolli comuni.
  - Sovrapposizione nei temi di ricerca (es. ICMATE-CNR e CNR).

Questa visualizzazione è utile per il debugging del sistema di embedding, la validazione semantica e l'individuazione di potenziali anomalie nei metadati.

### *11.8 Utilizzo applicazione Python Panel per Data Analysis dati da processo deposizione Perovskite per celle fotovoltaiche*

La soluzione implementata è stata realizzata come una “web application” interattiva utilizzando il framework Panel (<https://panel.holoviz.org/>) e integrata all'interno della piattaforma IEMAP. Per eseguire l'applicazione resa disponibile attraverso immagine Docker basta scaricarla da Docker hub ed eseguirla tramite:

**`docker run -p 5006:5006 sergioeanx/iemap-perovskite-ui:latest`**

Ovviamente occorre avere Docker installato sul proprio PC, per i passi necessari si rimanda alle seguenti risorse:

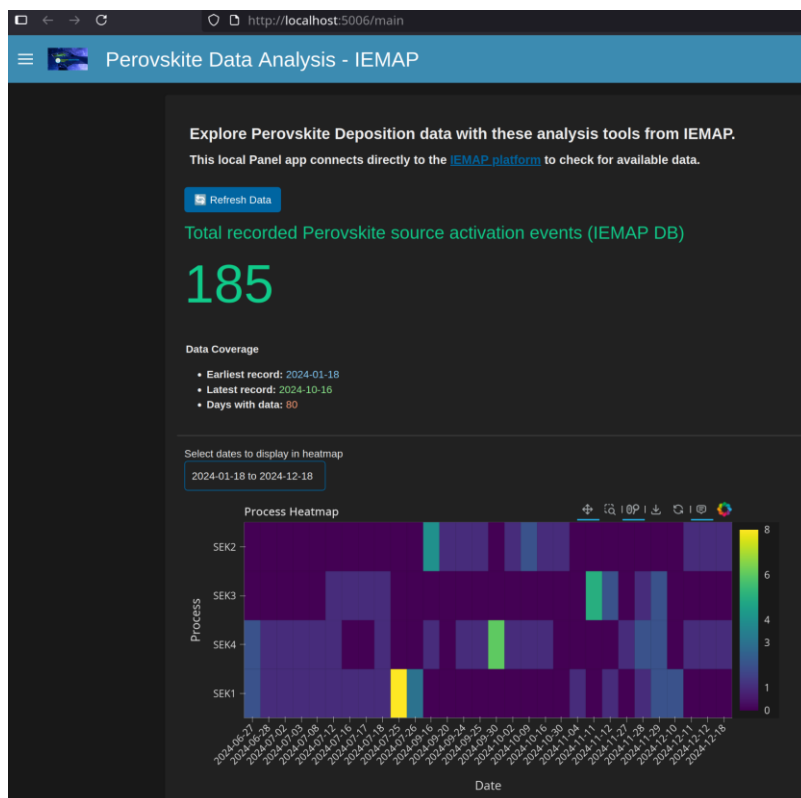
<https://docs.docker.com/get-started/>

<https://www.docker.com/101-tutorial/>

L'applicazione consente di esplorare visivamente i dati, selezionando intervalli temporali specifici e visualizzando i profili evolutivi dei principali parametri monitorati, come temperature, pressioni e velocità di evaporazione dei precursori.

Il tool si presenta come uno strumento pratico per ricercatori e tecnologi impegnati nella deposizione di materiali tipo perovskite, facilitando l'identificazione di condizioni operative ottimali e contribuendo alla ripetibilità e stabilizzazione dei processi. L'interfaccia grafica è semplice e intuitiva, con grafici dinamici e possibilità di esportazione dei risultati analizzati.

Di seguito alcune schermate esemplificative delle funzionalità disponibili tramite tale applicazione:



**Figura 37. Applicazione Python Panel per data analysis processo deposizione Perovskite per PV**

Tramite il framework Panel si crea una applicazione web totalmente sviluppata in Python che:

- Mostra quanti “processi” di deposizione sono disponibili sulla piattaforma IEMAP
- Numero totale di giorni in cui si è registrato almeno un processo
- Data primo processo inizio e data ultimo processo disponibile in DB
- Una heatmap interattiva che mostra quanti processi per ogni giorno sono stati registrati e quali sorgenti sono coinvolte (distinte in SEK1, SEK2, SEK3 e SEK4)

Selezionando una determinata data o un intervallo di tali è possibile generare grafici del tipo seguente:



Figura 38. Esempio grafici Data Analysis processo deposizione Perovskite

### 11.9 Risoluzione problemi e FAQ

#### Problemi comuni

Problema	Causa possibile	Soluzione
Errore di autenticazione (401) durante l'uso delle REST API	Token JWT assente o scaduto	Verificare di aver effettuato il login correttamente. Includere il token nell'header Authorization: Bearer <token>
Upload file fallito	Formato file non supportato o dimensione > 1 GB	Verificare che il file sia nei formati consentiti e di dimensione inferiore a 1GB
Dati non visualizzati nella dashboard Panel	Mancata connessione al DB o range temporale non corretto	Verificare la connessione a MongoDB, controllare i filtri applicati
Errore di validazione metadati	Mancanza di campi obbligatori o valori non validi	Usare il template Excel aggiornato o il modulo Python iemap-mi per una validazione assistita
L'account creato non è attivo	Email non verificata entro 24h	Ripetere la procedura di registrazione e verificare l'email in tempi brevi

### Domande frequenti (FAQ)

Q: È obbligatorio registrarsi per visualizzare i dati in IEMAP?

A: No, la consultazione dei progetti è libera. Solo la creazione di progetti e upload file richiede un account.

Q: Posso caricare dati da script Python?

A: Sì, è disponibile il modulo Python iemap-mi su PyPI.

Q: Come posso predire energia di formazione e potenziale redox?

A: Tramite file .CIF e REST API o modulo Python. Vedi sezioni 11.5 e 11.6.

Q: Quali formati di file sono accettati?

A: PDF, DOCX, XLSX, PNG, JPG, CIF, TXT e CSV, fino a 1GB per file.

Q: Posso richiedere la cancellazione del mio account?

A: Sì, dalla sezione "Profilo" dopo il login.

Q: Dove trovo il file Excel per l'inserimento dei dati?

A: Nella sezione "Progetti" del portale: metadata\_template.xlsx

Q: Quale documentazione tecnica è disponibile?

A: REST API Swagger UI (<https://iemap.enea.it/rest/docs>), documentazione Python (<https://iemap-mi-module.readthedocs.io>).

### Supporto tecnico

Per problemi non risolti:

 Email: [support@iemap.enea.it](mailto:support@iemap.enea.it)

 Teams (su appuntamento, [sergio.ferlito@enea.it](mailto:sergio.ferlito@enea.it))

## **11.10 Glossario Tecnico Piattaforma IEMAP (suddiviso per aree tematiche)**

### Architettura software e web

#### **SPA (Single Page Application)**

Architettura di applicazioni web in cui l'intera pagina viene caricata una sola volta e le interazioni successive avvengono tramite richieste asincrone al server, evitando il ricaricamento completo. In IEMAP, il front-end è implementato come SPA con VueJS v3, offrendo interfacce rapide e responsive.

#### **VueJS**

Framework JavaScript moderno e progressivo utilizzato per costruire interfacce utente. Nella piattaforma IEMAP, è stato scelto per la sua leggerezza, modularità e capacità di gestire in modo dinamico form, grafici e visualizzazioni interattive.

#### **VueRouter**

Libreria ufficiale di routing per VueJS, utilizzata per gestire la navigazione client-side nelle applicazioni SPA di IEMAP.

**Axios**

Libreria JavaScript per la gestione di richieste HTTP asincrone. È utilizzata da VueJS per comunicare con le REST API della piattaforma.

**Pinia**

State management library per VueJS, usata in IEMAP per mantenere lo stato delle componenti dell'interfaccia utente (UI) in modo reattivo.

**FastAPI**

Framework Python moderno, performante e asincrono per la realizzazione di REST API. Utilizzato per tutto il backend di IEMAP per via del supporto nativo a OpenAPI, Pydantic e JWT.

**Swagger UI**

Strumento per la generazione automatica della documentazione interattiva delle API REST, incluso nella piattaforma FastAPI.

**NGINX**

Web server e reverse proxy impiegato per servire l'interfaccia web IEMAP e reindirizzare le richieste HTTP verso i container Docker delle REST API.

**Certbot + Let's Encrypt**

Strumenti utilizzati per generare e rinnovare automaticamente certificati TLS, garantendo la connessione sicura HTTPS al sito IEMAP.

**Database e gestione dati****MongoDB Enterprise**

Database NoSQL orientato ai documenti, installato on-premise nella piattaforma IEMAP. Utilizzato per archiviare metadati dei progetti e dati sperimentali (collezioni standard e Time Series).

**Time Series (TS) Collection**

Tipo di collezione ottimizzata per il salvataggio efficiente di dati temporali, usata per la registrazione dei segnali dei sensori nei processi di deposizione perovskite.

**Ceph**

Sistema distribuito di object storage utilizzato in IEMAP per il salvataggio dei file fisici (documenti, immagini, file .CIF), con metadati associati archiviati su MongoDB.

**TTL Index**

Indice MongoDB con scadenza temporale, impiegato per cancellare automaticamente documenti (es. account temporanei) dopo un intervallo definito.

**Template Excel per metadati**

File Excel con convalida dati e fogli strutturati (Project, Process, Material, Parameters, Properties) per l'inserimento semi-guidato di progetti sulla piattaforma.

**Hash del file**

Codice identificativo univoco (es. SHA-256) calcolato sul contenuto del file per verificarne integrità e unicità prima dell'upload.

Ricerca semantica e vettoriale**Qdrant**

Motore di ricerca vettoriale open-source, scritto in Rust, usato in IEMAP per la gestione degli embedding semantici e per abilitare query semantiche e ibride.

**Embedding denso (Dense Vector)**

Rappresentazione numerica continua di un testo, ottenuta tramite modelli come BGE-m3, utile per ricerche per significato anziché per keyword esatte.

**Embedding sparso (Sparse Vector)**

Rappresentazione testuale basata su pesi attribuiti a token specifici (TF-IDF, BM25 o SPLADE), usata per garantire match esatti con termini specialistici.

**Modello BAAI/bge-m3**

Modello Transformer multilingua per la generazione di dense embedding ottimizzati per il semantic search. Utilizzato per l'indicizzazione del testo aggregato ("combined\_text") in IEMAP.

**Modello SPLADE (naver/splade-cocondenser)**

Modello per la generazione di sparse embedding altamente performanti nel recupero di contenuti scientifici con termini tecnici specifici.

**Hybrid Search**

Tecnica che combina dense e sparse embedding (bilanciati da un parametro  $\alpha$ ) per ottimizzare il recupero documentale in scenari tecnici avanzati.

**Payload vettoriale**

Struttura JSON contenente vettori di embedding e metadati, che viene inviata a Qdrant per l'indicizzazione e la successiva interrogazione semantica.

Intelligenza Artificiale e predizione proprietà materiali**geoCGNN**

Modello AI basato su reti neurali convoluzionali su grafi cristallini arricchite con informazioni geometriche. In IEMAP è usato per predire energia di formazione e potenziale redox da file CIF.

**CIF (Crystallographic Information File)**

Formato standard usato in cristallografia per rappresentare la struttura tridimensionale dei materiali.

**Formation Energy**

Quantità di energia necessaria per formare un composto a partire dagli elementi in stato standard. Indica stabilità termodinamica.

**Redox Potential**

Valore che indica la tendenza di un materiale a perdere o acquisire elettroni (ossidazione o riduzione).

**ONNX**

Formato interoperabile per la rappresentazione di modelli AI, usato per l'inferenza efficiente (es. via ONNX Runtime o TensorRT).

Sicurezza e autenticazione**JWT (JSON Web Token)**

Token crittografico contenente le credenziali dell'utente, usato per l'autenticazione nelle REST API IEMAP.

**FastAPIUsers**

Libreria Python per la gestione di autenticazione e utenti in applicazioni FastAPI, impiegata per registrazione, login, recupero password.

**HttpOnly Cookie**

Cookie non accessibile via JavaScript, usato per aumentare la sicurezza del token JWT contro attacchi XSS.

**Frontegg / Auth0**

Soluzioni SaaS per autenticazione e gestione utenti, citate come alternative più avanzate ma non adottate in IEMAP.

Ontologie, FAIR e interoperabilità**Principi FAIR**

Linee guida per garantire che i dati scientifici siano Findable, Accessible, Interoperable, Reusable. In IEMAP, si punta alla conformità tramite metadati, ontologie OWL e licenze aperte.

**OWL (Web Ontology Language)**

Linguaggio per la definizione formale di concetti e relazioni tra entità, utilizzato per modellare l'ontologia IEMAP (es. Project, Material, Process).

**JSON-LD / Turtle**

Formati di serializzazione usati per rappresentare ontologie in modo compatibile con il web semantico.

**Zenodo**

Repository open per la pubblicazione di dataset FAIR e ontologie. L'ontologia IEMAP è pubblicata con DOI su Zenodo.