

Documentazione Tecnica del Progetto

Erasmus Help Desk

26 ottobre 2025

Indice

1 Architettura	2
1.1 Panoramica	2
1.2 Flussi principali	2
1.2.1 Step 1: Bando	2
1.2.2 Step 2: Destinazioni	2
1.2.3 Step 3: Esami	2
2 Setup e Configurazione	3
2.1 Prerequisiti	3
2.2 Variabili d'ambiente	3
2.3 Dipendenze principali	3
2.4 Avvio in sviluppo	3
3 Database	4
3.1 Schema logico	4
3.2 Tipi di documento	4
3.3 Esempio di query	4
4 Servizi applicativi	5
4.1 File app/services/rag_service.py	5
4.2 Gestione errori	5
5 API	6
5.1 Studenti	6
5.1.1 GET /api/students/universities	6
5.1.2 POST /api/students/step1	6
5.1.3 POST /api/students/departments	6
5.1.4 POST /api/students/step2	6
5.1.5 POST /api/students/step3	6
5.2 Università	6
5.2.1 POST /api/universities/upload/erasmus-call	6
5.2.2 POST /api/universities/upload/destinazioni	6
5.2.3 POST /api/universities/upload/erasmus-courses	6
6 Integrazione LLM	7
6.1 Google Generative AI	7
6.2 Prompt	7
6.2.1 Step 2	7
6.2.2 Step 3	7

6.3	Parsing output	7
7	Testing	8
7.1	Unit test suggeriti	8
7.2	Integrazione	8
8	Deployment	9
8.1	Sviluppo	9
8.2	Porta 8000	9
9	Sicurezza	10
9.1	Gestione segreti	10
9.2	Upload PDF	10
10	Troubleshooting	11
10.1	Step 2: Dipartimento non trovato	11
10.2	Step 3: Nessun file esami trovato	11
10.3	Porta 8000 occupata	11

Capitolo 1

Architettura

1.1 Panoramica

Il sistema è composto da:

- Backend Python (FastAPI) con moduli in `app/`
- Servizi applicativi (RAG, PDF, Vector DB) in `app/services/`
- Persistenza SQLite gestita da `app/core/database.py`
- Frontend statico in `frontend/`

1.2 Flussi principali

1.2.1 Step 1: Bando

Recupera bando attivo dall'archivio documenti, opzionale retrieval su vector store e riassunto con LLM.

1.2.2 Step 2: Destinazioni

1. Estrazione testo da PDF “destinazioni” (tabelle + testo)
2. Isolamento sezione di dipartimento
3. Prompt a LLM per estrarre un array JSON di destinazioni

1.2.3 Step 3: Esami

1. Ricerca PDF corsi Erasmus (`document_type = corsi_erasmus`)
2. Estrazione testo e prompting per matching con piano di studi

Capitolo 2

Setup e Configurazione

2.1 Prerequisiti

- Python 3.10+
- Virtualenv/Conda
- Latexmk (MiKTeX/TeX Live) per compilare la documentazione

2.2 Variabili d'ambiente

- GOOGLE_API_KEY: chiave API Google Generative AI

2.3 Dipendenze principali

- fastapi, uvicorn
- google-generativeai
- pdfplumber, pymupdf

2.4 Avvio in sviluppo

```
uvicorn app.main:app --host 0.0.0.0 --port 8000 --reload
```

Capitolo 3

Database

3.1 Schema logico

Tabelle principali:

- `universities`: anagrafica atenei (`id`, `university_name`, `email`, ...)
- `uploaded_documents`: documenti caricati (`id`, `university_id`, `document_type`, `file_path`, `stored_filename`, ...)

3.2 Tipi di documento

- `erasmus_call`
- `destinazioni`
- `corsi_erasmus`

3.3 Esempio di query

Ricerca corsi per università di destinazione:

```
1 SELECT d.*  
2 FROM uploaded_documents d  
3 JOIN universities u ON d.university_id = u.id  
4 WHERE d.document_type = 'corsi_erasmus'  
5 AND d.is_active = 1  
6 AND (  
7     LOWER(u.university_name) = LOWER(?)  
8     OR LOWER(u.university_name) LIKE LOWER(?)  
9     OR LOWER(?) LIKE '%' || LOWER(u.university_name) || '%'  
10    )  
11 ORDER BY d.upload_date DESC  
12 LIMIT 1;
```

Capitolo 4

Servizi applicativi

4.1 File app/services/rag_service.py

Funzioni principali:

- `clean_and_parse_json_response`: parsing robusto dell'output LLM
- `markdown_to_html`: rendering Markdown per frontend
- `extract_department_section`: isolamento sezione dipartimento
- `get_call_summary`: riassunto bando
- `get_available_departments`: elenco dipartimenti
- `analyze_destinations_for_department`: estrazione destinazioni dal PDF
- `analyze_exams_compatibility`: matching esami

4.2 Gestione errori

Propagazione eccezioni specifiche e log diagnostici (lunghezze testo, header trovati, parsing JSON).

Capitolo 5

API

5.1 Studenti

5.1.1 GET /api/students/universities

Elenco università con bandi attivi.

5.1.2 POST /api/students/step1

Input: `home_university`. Output: riassunto bando + `session_id`.

5.1.3 POST /api/students/departments

Input: `session_id`. Output: lista dipartimenti per l'ateneo selezionato.

5.1.4 POST /api/students/step2

Input: `session_id`, `department`, `period`. Output: array destinazioni (JSON).

5.1.5 POST /api/students/step3

Input (multipart): `session_id`, `destination_university_name`, `study_plan_file`. Output: analisi compatibilità + link PDF.

5.2 Università

5.2.1 POST /api/universities/upload/erasmus-call

Carica PDF bando.

5.2.2 POST /api/universities/upload/destinazioni

Carica PDF destinazioni.

5.2.3 POST /api/universities/upload/erasmus-courses

Carica PDF corsi (`document_type=corsi_erasmus`).

Capitolo 6

Integrazione LLM

6.1 Google Generative AI

- Configurazione: `genai.configure(api_key=GOOGLE_API_KEY)`
- Modello: `gemini-2.0-flash`
- Uso asincrono: `generate_content_async(template)`

6.2 Prompt

6.2.1 Step 2

Vincolo: restituzione esclusiva di array JSON con campi fedeli al PDF.

6.2.2 Step 3

Oggetto JSON con: corrispondenze, suggerimenti, punteggio compatibilità, note su periodo.

6.3 Parsing output

`clean_and_parse_json_response`: rimozione code fences, parsing diretto e fallback regex.

Capitolo 7

Testing

7.1 Unit test suggeriti

- `clean_and_parse_json_response`: casi con code fences, testo extra, errori JSON
- `extract_department_section`: header varianti, concatenazioni, assenza header
- `extract_text_from_pdf`: gestione PDF vuoti o illeggibili

7.2 Integrazione

- Step 2 end-to-end con PDF reale di destinazioni
- Step 3 con DB popolato e PDF corsi reale

Capitolo 8

Deployment

8.1 Sviluppo

Esecuzione locale:

```
uvicorn app.main:app --host 0.0.0.0 --port 8000 --reload
```

8.2 Porta 8000

Per liberare la porta su Windows (PowerShell):

```
Get-NetTCPConnection -LocalPort 8000 -State Listen |  
Select -ExpandProperty OwningProcess -Unique |  
% { Stop-Process -Id $_ -Force }
```

Capitolo 9

Sicurezza

9.1 Gestione segreti

- GOOGLE_API_KEY in variabile d'ambiente/.env non versionato

9.2 Upload PDF

- Validazione tipo (solo PDF)
- Path salvati in uploaded_documents.file_path

Capitolo 10

Troubleshooting

10.1 Step 2: Dipartimento non trovato

- Verificare testo estratto e presenza header “Dipartimento”
- Usare estrazione più tollerante per header e normalizzazione input

10.2 Step 3: Nessun file esami trovato

- Verificare che il documento esista con `document_type='corsi_erasmus'`
- Controllare coerenza tra `destination_university_name` e `universities.university_name`

10.3 Porta 8000 occupata

- Kill processi come indicato in Deployment