



Specifica Tecnica

NearYou
Smart custom advertising platform

sevenbits.swe.unipd@gmail.com



Registro modifiche

Versione	Data	Autore	Verificatore	Descrizione
0.1.2	2025-03-05	Leonardo Trolese	Manuel Gusella	Conclusione redazione sottosezione Panoramica Linguaggi della sezione Tecnologie
0.1.1	2025-03-02	Leonardo Trolese	Manuel Gusella	Redazione sottosezione Panoramica Linguaggi della sezione Tecnologie
0.1.0	2025-02-26	Leonardo Trolese	Manuel Gusella	Inizio redazione del documento

Indice

1	Introduzione	5
1.1	Scopo del documento	5
1.2	Glossario	5
1.3	Riferimenti	5
1.3.1	Riferimenti normativi	5
1.3.2	Riferimenti informativi	5
2	Tecnologie	6
2.1	Panoramica dei linguaggi	6
2.1.1	Python	6
2.1.1.1	Versione	6
2.1.1.2	Documentazione	6
2.1.1.3	Utilizzo operato nel progetto	6
2.1.1.4	Dipendenze	6
2.1.2	SQL (Structured Query Language)	8
2.1.2.1	Versione	8
2.1.2.2	Documentazione	8
2.1.2.3	Utilizzo operato nel progetto	8
2.1.3	YAML (YAML Ain't Markup Language)	8
2.1.3.1	Versione	8
2.1.3.2	Documentazione	8
2.1.3.3	Utilizzo operato nel progetto	8
2.1.3.4	Framework e tools correlati	8
2.1.4	JSON (JavaScript Object Notation)	8
2.1.4.1	Versione	8
2.1.4.2	Documentazione	9
2.1.4.3	Utilizzo operato nel progetto	9
2.1.4.4	Framework e tools correlati	9
2.2	Strumenti e servizi	9
2.2.1	Docker	9
3	API	9
4	Architettura di sistema	9
5	Architettura di deployment	9
6	Stato dei requisiti funzionali	9

Elenco delle figure

Elenco delle tabelle

1 Introduzione

1.1 Scopo del documento

Il presente documento si propone come una risorsa completa per la comprensione degli aspetti tecnici e progettuali della piattaforma "NearYou", dedicata alla creazione di soluzioni di advertising personalizzato tramite intelligenza artificiale. L'obiettivo principale è fornire una descrizione dettagliata dell'architettura implementativa e di deployment, illustrando le tecnologie adottate e le motivazioni alla base delle scelte progettuali.

Nel contesto dell'architettura implementativa, il documento analizza nel dettaglio i moduli principali del sistema, i design pattern utilizzati. Saranno inclusi diagrammi delle classi, e una spiegazione dettagliata dei design pattern utilizzati e delle motivazioni di queste scelte.

Gli obiettivi di questo documento sono: motivare le decisioni architetturali, fungere da guida per lo sviluppo della piattaforma, e garantire la piena tracciabilità e copertura dei requisiti definiti nel documento di *Analisi dei Requisiti v1.0.0*.

In sintesi, il documento intende essere un punto di riferimento essenziale per tutti gli attori coinvolti nel ciclo di vita del progetto, offrendo una visione chiara e strutturata delle fondamenta tecniche che sorreggono NearYou e delle logiche che ne determinano il funzionamento.

1.2 Glossario

Con l'intento di evitare ambiguità interpretative del linguaggio utilizzato, viene fornito un Glossario che si occupa di esplicitare il significato dei termini che riguardano il contesto del Progetto_G. I termini presenti nel glossario sono contrassegnati con una *G* a pedice : Termine_G.

I termini composti, oltre alla *G* a pedice, saranno uniti da un "-" come segue: termine-composto_G.

Le definizioni sono presenti nell'apposito documento *Glossario v1.0.0.pdf*.

1.3 Riferimenti

1.3.1 Riferimenti normativi

- Regolamento del Progetto_G didattico
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>
(Consultato: 2025-02-10).
- Capitolato_G C4 - NearYou - Smart custom advertising platform
<https://www.math.unipd.it/~tullio/IS-1/2024/Progetto/C4p.pdf>
(Consultato: 2025-02-10).
- *Norme di Progetto v1.0.0*

1.3.2 Riferimenti informativi

- *Glossario v1.0.0*
- *Analisi dei Requisiti v1.0.0*
- Analisi-dei-Requisiti_G - SWE 2024-25
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T05.pdf>
(Consultato: 2025-02-10).
- Dependency Injection - SWE 2024-25
- Design Pattern Creazionali - SWE 2024-25
- Design Pattern Strutturali - SWE 2024-25
- Software Architecture Patterns - SWE 2024-25

- Verbali Interni
- Verbali Esterni

2 Tecnologie

Questa sezione descrive strumenti e tecnologie impiegate nello sviluppo del software del progetto nearYou. Descriveremo quindi a seguire le tecnologie usate per lo sviluppo del software, le librerie e i framework utilizzati, e le motivazioni di queste scelte.

2.1 Panoramica dei linguaggi

2.1.1 Python

Si tratta di un linguaggio di programmazione ad alto livello, interpretato e orientato agli oggetti, noto per la sua sintassi chiara e leggibile. Python dispone anche di una vasta libreria standard ed è caratterizzato dalla grande quantità di framework disponibili.

2.1.1.1 Versione

3.12.2

2.1.1.2 Documentazione

La documentazione può essere trovata qui: <https://docs.python.org/> (consultato: 2025-03-02)

2.1.1.3 Utilizzo operato nel progetto

- Creazione dei sensori e simulazione degli spostamenti;
- Generazione dei punti di interesse oggetto del messaggio pubblicitario;
- Interazione con il database per la persistenza dei dati;
- Interazione con l'LLM mediante API;
- Logica di selezione dei punti di interesse rilevanti per l'utente;
- Testing.

2.1.1.4 Dipendenze

- ClickHouse Connect
 - . Versione: 0.6.8
 - . Documentazione: <https://clickhouse.com/docs/integrations/python> (consultato: 2025-03-02)
 - . Descrizione: ClickHouse Connect è una libreria Python che consente di connettersi al database ClickHouse, eseguire query SQL in modo veloce ed efficiente.
- PyFlink
 - . Versione: 1.18.1
 - . Documentazione: https://pyflink.readthedocs.io/en/main/getting_started/index.html (consultato: 2025-03-02)
 - . Descrizione: PyFlink è l'API Python di Apache Flink, che permette di scrivere e gestire applicazioni per l'elaborazione di flussi e batch di dati distribuiti in tempo reale, sfruttando la potenza e la scalabilità di Flink direttamente con Python.
- LangChain

- . Versione: 0.1.12
- . Documentazione: <https://python.langchain.com/docs/introduction/> (consultato: 2025-03-02)
- . Descrizione: LangChain è una libreria Python che semplifica la creazione di applicazioni basate su modelli linguistici, consentendo di orchestrare prompt, gestire la memoria della conversazione e integrare fonti di dati esterne.
- Groq
 - . Versione: 0.4.2
 - . Documentazione: <https://console.groq.com/docs/libraries> (consultato: 2025-03-02)
 - . Descrizione: Groq è una libreria Python che permette di interagire con modelli di linguaggio tramite API, facilitando la generazione di testo, risposte conversazionali e completamenti.
- Confluent Kafka
 - . Versione: 2.8.0
 - . Documentazione: <https://docs.confluent.io/kafka/overview.html> (consultato: 2025-03-02)
 - . Descrizione: Confluent Kafka è una libreria Python che semplifica l'interazione con Apache Kafka, permettendo di produrre, consumare e gestire stream di dati in tempo reale, sfruttando la piattaforma Confluent.
- GeoPy
 - . Versione: 2.4.1
 - . Documentazione: <https://geopy.readthedocs.io/en/stable/index.html> (consultato: 2025-03-02)
 - . Descrizione: GeoPy è una libreria Python che facilita la geocodifica, il calcolo delle distanze e l'interazione con servizi di localizzazione.
- OSMnx
 - . Versione: 1.9.1
 - . Documentazione: <https://osmnx.readthedocs.io/en/stable/> (consultato: 2025-03-02)
 - . Descrizione: OSMnx è una libreria Python per scaricare, analizzare e visualizzare reti stradali e dati geografici da OpenStreetMap, utile per creare grafi, calcolare percorsi e studiare l'urbanistica.
- Faker
 - . Versione: 24.1.0
 - . Documentazione: <https://faker.readthedocs.io/en/master/> (consultato: 2025-03-02)
 - . Descrizione: Faker è una libreria Python che genera dati falsi realistici, come nomi, indirizzi, email o testi casuali, utile per test, mock di database e prototipazione.
- Pylint
 - . Versione: 3.0.3
 - . Documentazione: <https://pylint.pycqa.org/en/latest/index.html> (consultato: 2025-03-03)
 - . Pylint è uno strumento di analisi statica del codice Python per rilevare errori, violazioni di stile e migliorare la qualità del codice.
- pytest
 - . Versione: 7.4.3
 - . Documentazione: <https://docs.pytest.org/en/stable/> (consultato: 2025-03-03)
 - . Descrizione: Pytest è una libreria Python potente e flessibile che consente di scrivere ed eseguire test automatizzati, con funzionalità avanzate come asserzioni intuitive, fixture e supporto per test parametrizzati.

2.1.2 SQL (Structured Query Language)

Usato come dialetto SQL specifico per l'interazione con il database ClickHouse, un DBMS colonnare ad alte prestazioni.

2.1.2.1 Versione

ClickHouse 24.10

2.1.2.2 Documentazione

La documentazione può essere trovata qui: <https://clickhouse.com/docs/sql-reference> (consultato: 2025-03-05)

2.1.2.3 Utilizzo operato nel progetto

- Definizione schema del database;
- Query per il recupero dei dati per la visualizzazione;
- Query per il recupero dei dati per la generazione dei messaggi pubblicitari.

2.1.3 YAML (YAML Ain't Markup Language)

YAML è un linguaggio di serializzazione dei dati human-readable progettato per essere facilmente leggibile dall'uomo. La sua sintassi è basata sull'indentazione e ha una struttura chiave-valore. YAML è spesso usato in file di configurazione.

2.1.3.1 Versione

YAML 1.2

2.1.3.2 Documentazione

La documentazione può essere trovata qui: <https://yaml.org/spec/1.2.2/> (consultato: 2025-03-05)

2.1.3.3 Utilizzo operato nel progetto

- *docker-compose.yml* per la configurazione dei servizi;
- Workflow GitHub Actions per CI/CD;
- File di configurazione per Grafana;
- File di configurazione per i test.

2.1.3.4 Framework e tools correlati

- Docker Compose;
- GitHub Actions;
- Grafana.

2.1.4 JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) è un formato di interscambio dati leggero, text-based e indipendente dal linguaggio, che utilizza una struttura di coppie chiave-valore e array. Progettato per essere facilmente leggibile sia dall'uomo che dalle macchine.

2.1.4.1 Versione

JSON 2.0

2.1.4.2 Documentazione

La documentazione può essere trovata qui: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON (consultato: 2025-03-02)

2.1.4.3 Utilizzo operato nel progetto

- Configurazione dashboard Grafana;
- Invio e ricezione dati mediante il broker Kafka; [-] Risposta API del servizio Groq;

2.1.4.4 Framework e tools correlati

- Groq;
- Kafka;
- Grafana.

2.2 Strumenti e servizi

2.2.1 Docker

Per lo sviluppo, il testing e il rilascio dell'applicativo sono stati usati dei container Docker, come suggerito dalla proponente, al fine di garantire ambienti di sviluppo e di test isolati, consistenti e riproducibili.

3 API

4 Architettura di sistema

5 Architettura di deployment

6 Stato dei requisiti funzionali