



# Specifica Tecnica

NearYou  
Smart custom advertising platform

[sevenbits.swe.unipd@gmail.com](mailto:sevenbits.swe.unipd@gmail.com)



### Registro modifiche

Versione	Data	Autore	Verificatore	Descrizione
0.1.3	2025-03-05	Alfredo Rubino	Manuel Gusella	Redazione sottosezione Strumenti e Servizi della sezione Tecnologie
0.1.2	2025-03-05	Leonardo Trolese	Manuel Gusella	Conclusione redazione sottosezione Panoramica dei Linguaggi della sezione Tecnologie
0.1.1	2025-03-02	Leonardo Trolese	Manuel Gusella	Redazione sottosezione Panoramica dei Linguaggi della sezione Tecnologie
0.1.0	2025-02-26	Leonardo Trolese	Manuel Gusella	Inizio redazione del documento

# Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del documento . . . . .	6
1.2	Glossario . . . . .	6
1.3	Riferimenti . . . . .	6
1.3.1	Riferimenti normativi . . . . .	6
1.3.2	Riferimenti informativi . . . . .	6
<b>2</b>	<b>Tecnologie</b>	<b>7</b>
2.1	Panoramica dei linguaggi . . . . .	7
2.1.1	Python . . . . .	7
2.1.1.1	Versione . . . . .	7
2.1.1.2	Documentazione . . . . .	7
2.1.1.3	Utilizzo operato nel progetto . . . . .	7
2.1.1.4	Dipendenze . . . . .	7
2.1.2	SQL (Structured Query Language) . . . . .	9
2.1.2.1	Versione . . . . .	9
2.1.2.2	Documentazione . . . . .	9
2.1.2.3	Utilizzo operato nel progetto . . . . .	9
2.1.3	YAML (YAML Ain't Markup Language) . . . . .	9
2.1.3.1	Versione . . . . .	9
2.1.3.2	Documentazione . . . . .	9
2.1.3.3	Utilizzo operato nel progetto . . . . .	9
2.1.3.4	Framework e tools correlati . . . . .	9
2.1.4	JSON (JavaScript Object Notation) . . . . .	10
2.1.4.1	Versione . . . . .	10
2.1.4.2	Documentazione . . . . .	10
2.1.4.3	Utilizzo operato nel progetto . . . . .	10
2.1.4.4	Framework e tools correlati . . . . .	10
2.2	Strumenti e servizi . . . . .	10
2.2.1	Docker . . . . .	10
2.2.1.1	Versione . . . . .	10
2.2.1.2	Documentazione . . . . .	10
2.2.1.3	Utilizzo nel progetto . . . . .	10
2.2.2	Grafana . . . . .	10
2.2.2.1	Versione . . . . .	10
2.2.2.2	Documentazione . . . . .	11
2.2.2.3	Utilizzo nel progetto . . . . .	11
2.2.3	ClickHouse . . . . .	11
2.2.3.1	Versione . . . . .	11
2.2.3.2	Documentazione . . . . .	11
2.2.3.3	Utilizzo nel progetto . . . . .	11
2.2.4	Apache Kafka . . . . .	11
2.2.4.1	Versione . . . . .	11
2.2.4.2	Documentazione . . . . .	11
2.2.4.3	Utilizzo nel progetto . . . . .	11
2.2.5	Apache ZooKeeper . . . . .	11
2.2.5.1	Versione . . . . .	11
2.2.5.2	Documentazione . . . . .	11
2.2.5.3	Utilizzo nel progetto . . . . .	12
2.2.6	Apache Flink . . . . .	12
2.2.6.1	Versione . . . . .	12
2.2.6.2	Documentazione . . . . .	12
2.2.6.3	Utilizzo nel progetto . . . . .	12
<b>3</b>	<b>API</b>	<b>12</b>
<b>4</b>	<b>Architettura di sistema</b>	<b>12</b>

<b>5</b>	<b>Architettura di deployment</b>	<b>12</b>
<b>6</b>	<b>Stato dei requisiti funzionali</b>	<b>12</b>

## 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<sub>G</sub>. I termini presenti nel glossario sono contrassegnati con una *G* a pedice : Termine<sub>G</sub>.

I termini composti, oltre alla *G* a pedice, saranno uniti da un "-" come segue: termine-composto<sub>G</sub>.

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

## 1.3 Riferimenti

### 1.3.1 Riferimenti normativi

- Regolamento del Progetto<sub>G</sub> didattico  
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/PD1.pdf>  
(Consultato: 2025-02-10).
- Capitolato<sub>G</sub> 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<sub>G</sub> - SWE 2024-25  
<https://www.math.unipd.it/~tullio/IS-1/2024/Dispense/T05.pdf>  
(Consultato: 2025-02-10).
- Dependency Injection - SWE 2024-25  
<https://www.math.unipd.it/~rcardin/swea/2022/Design%20Pattern%20Architetturali%20-%20Dependency%20Injection.pdf>  
(Consultato: 2025-02-26).
- Design Pattern Creazionali - SWE 2024-25  
<https://www.math.unipd.it/~rcardin/swea/2022/Design%20Pattern%20Creazionali.pdf>  
(Consultato: 2025-02-26).

- Design Pattern Strutturali - SWE 2024-25  
<https://www.math.unipd.it/~rcardin/swea/2022/Design%20Pattern%20Strutturali.pdf>  
(Consultato: 2025-02-26).
- Software Architecture Patterns - SWE 2024-25  
<https://www.math.unipd.it/~rcardin/swea/2022/Software%20Architecture%20Patterns.pdf>  
(Consultato: 2025-02-26).
- Verbali Interni
- Verbali Esterni

## 2 Tecnologie

Questa sezione descrive strumenti e tecnologie impiegate nello sviluppo del software del progetto nearYou. Saranno analizzate quindi le tecnologie utilizzate 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](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 consultata al seguente link: <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 consultata al seguente link: <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 consultata al seguente link: [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON](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. Docker è una piattaforma di containerizzazione che permette di eseguire applicazioni in ambienti isolati e riproducibili.

#### 2.2.1.1 Versione

24.0.7

#### 2.2.1.2 Documentazione

La documentazione può essere consultata al seguente link: <https://docs.docker.com/> (Consultato: 2025-03-05)

#### 2.2.1.3 Utilizzo nel progetto

- Creazione di container per il database, il backend e il servizio di elaborazione dati;
- Definizione di un ambiente standardizzato per il deployment.

### 2.2.2 Grafana

Grafana è una piattaforma open-source per la visualizzazione e il monitoraggio dei dati, utilizzata per creare dashboard interattive e grafici da fonti di dati eterogenee.

#### 2.2.2.1 Versione

11.5.2

#### 2.2.2.2 Documentazione

La documentazione può essere consultata al seguente link: <https://grafana.com/docs/> (Consultato: 2025-03-05)

#### 2.2.2.3 Utilizzo nel progetto

- Creazione di dashboard per il monitoraggio delle performance del sistema;
- Analisi delle metriche degli utenti e degli annunci pubblicitari.

### 2.2.3 ClickHouse

ClickHouse è un sistema di gestione di database (DBMS) di tipo column-oriented, progettato specificamente per l'analisi di grandi volumi di dati con prestazioni estremamente elevate.

#### 2.2.3.1 Versione

24.10

#### 2.2.3.2 Documentazione

La documentazione può essere consultata al seguente link: <https://clickhouse.com/docs/en/> (Consultato: 2025-03-05)

#### 2.2.3.3 Utilizzo nel progetto

- Archiviazione dei dati degli utenti e degli annunci pubblicitari;
- Query ad alte prestazioni per l'analisi dei dati.

### 2.2.4 Apache Kafka

Apache Kafka è una piattaforma di streaming distribuita. Progettata per gestire flussi di dati in tempo reale in modo scalabile e affidabile, è ampiamente utilizzata nel data streaming e nell'integrazione dei dati nelle applicazioni moderne.

#### 2.2.4.1 Versione

7.6.0

#### 2.2.4.2 Documentazione

La documentazione può essere consultata al seguente link: <https://kafka.apache.org/documentation/> (Consultato: 2025-03-05)

#### 2.2.4.3 Utilizzo nel progetto

- Pubblicazione e consumo di eventi relativi ai movimenti degli utenti;
- Integrazione con Flink per l'elaborazione dei dati in tempo reale.

### 2.2.5 Apache ZooKeeper

Apache ZooKeeper è un servizio centralizzato per il mantenimento delle informazioni di configurazione, la sincronizzazione distribuita e la gestione dei nodi nei sistemi distribuiti.

#### 2.2.5.1 Versione

7.6.0

#### 2.2.5.2 Documentazione

La documentazione può essere consultata al seguente link: <https://zookeeper.apache.org/documentation.html> (Consultato: 2025-03-05)

#### **2.2.5.3 Utilizzo nel progetto**

- Coordinamento e gestione dei broker Kafka;
- Sincronizzazione tra i nodi del sistema distribuito;
- Monitoraggio dello stato dei servizi distribuiti.

#### **2.2.6 Apache Flink**

Apache Flink è un framework per l'elaborazione di flussi di dati in tempo reale, utilizzato per la gestione dei dati generati dagli utenti.

##### **2.2.6.1 Versione**

1.20.0

##### **2.2.6.2 Documentazione**

La documentazione può essere consultata al seguente link: <https://nightlies.apache.org/flink/flink-docs-stable/> (Consultato: 2025-03-05)

##### **2.2.6.3 Utilizzo nel progetto**

- Elaborazione in tempo reale dei dati di posizione degli utenti;
- Analisi dei flussi di dati per la generazione di messaggi pubblicitari personalizzati.

## **3 API**

## **4 Architettura di sistema**

## **5 Architettura di deployment**

## **6 Stato dei requisiti funzionali**