

「αode」



# Analisi dei requisiti

2025-03-10

|              |               |
|--------------|---------------|
| Responsabile | Nicolò Bovo   |
| Redattori    | Romeo Calero  |
| Verificatori | Elia Leonetti |

## Indice

|   |          |
|---|----------|
| <b>I. Introduzione .....</b>                      | <b>4</b> |
| I - 1. Scopo del documento .....                  | 4        |
| I - 2. Scopo del prodotto .....                   | 4        |
| I - 3. Riferimenti .....                          | 4        |
| I - 3.1. Riferimenti tecnici .....                | 4        |
| I - 3.2. Riferimenti implementativi .....         | 4        |
| <b>II. Descrizione del prodotto .....</b>         | <b>4</b> |
| II - 1. Obiettivi del prodotto .....              | 4        |
| II - 1.1. Obiettivi primari: .....                | 5        |
| II - 1.2. Obiettivi secondari: .....              | 5        |
| II - 2. Funzionalità e vincoli del prodotto ..... | 5        |
| II - 2.1. Funzionalità implementate: .....        | 5        |
| II - 2.2. Vincoli tecnici: .....                  | 6        |
| II - 3. Utenti e loro caratteristiche .....       | 6        |
| II - 3.1. Tipologie di utenti: .....              | 6        |
| <b>III. Architettura del sistema .....</b>        | <b>6</b> |
| III - 1. Componenti Core .....                    | 6        |
| III - 2. Flusso Dati .....                        | 7        |
| <b>IV. Casi d'uso .....</b>                       | <b>7</b> |
| IV - 1. Introduzione .....                        | 7        |
| IV - 2. Attori .....                              | 7        |

**Lista delle figure**

**Lista delle tabelle**

## I. Introduzione

### I - 1. Scopo del documento

Il presente documento di Analisi dei Requisiti fornisce una descrizione dettagliata e completa del sistema **NearYou**, una piattaforma di notifiche personalizzate basate sulla posizione che utilizza intelligenza artificiale generativa (AI generativa) per creare messaggi contestuali e rilevanti per gli utenti.

Il documento costituisce la base per lo sviluppo del sistema, definendo chiaramente funzionalità, vincoli, casi d'uso e requisiti tecnici derivanti dall'analisi del codice sorgente e dell'architettura implementata.

### I - 2. Scopo del prodotto

**NearYou** è una piattaforma tecnologica che combina **tracking della posizione in tempo reale**, **database geospaziali**, **intelligenza artificiale generativa** e **analytics avanzate** per fornire notifiche personalizzate agli utenti quando si trovano in prossimità di negozi o punti di interesse.

Il sistema è progettato per:

- **Tracciare la posizione** degli utenti in tempo reale attraverso simulazione GPS
- **Generare messaggi personalizzati** utilizzando LLM (Large Language Models) quando l'utente è vicino a un negozio
- **Fornire una dashboard interattiva** con visualizzazione su mappa delle posizioni, percorsi e notifiche
- **Memorizzare e analizzare** tutti i dati per fornire insights attraverso dashboard analytics
- **Aggiornare automaticamente** il database dei negozi tramite ETL da OpenStreetMap

### I - 3. Riferimenti

#### I - 3.1. Riferimenti tecnici

- Docker Compose: `docker-compose.yml`
- Configurazione ambiente: `.env`
- Documentazione API: FastAPI autodocs ( `/docs` )
- Database schema: Script di inizializzazione ClickHouse e PostgreSQL

#### I - 3.2. Riferimenti implementativi

- Repository GitHub del progetto
- Dockerfile e script di deployment
- Configurazioni di monitoraggio (Grafana, Prometheus)

## II. Descrizione del prodotto

### II - 1. Obiettivi del prodotto

Il sistema **NearYou** ha come obiettivo principale la **creazione di un'esperienza utente personalizzata e contestuale** attraverso l'integrazione di diverse tecnologie avanzate:

## II - 1.1. Obiettivi primari:

1. **Personalizzazione intelligente:** Utilizzare profili utente (età, professione, interessi) per generare messaggi rilevanti tramite LLM
2. **Geolocalizzazione precisa:** Implementare tracking in tempo reale con calcolo di distanze attraverso PostGIS
3. **Scalabilità e performance:** Gestire flussi di dati in tempo reale attraverso Apache Kafka e Bytewax
4. **Analytics avanzate:** Fornire insights attraverso dashboard Grafana con metriche di business e sistema
5. **Automazione:** Mantenere aggiornato il database negozi attraverso ETL automatici da OpenStreetMap

## II - 1.2. Obiettivi secondari:

- **Caching intelligente:** Ridurre latenza e costi LLM attraverso cache Redis
- **Monitoraggio completo:** Garantire osservabilità del sistema con Prometheus, Loki e metriche custom
- **Facilità di deployment:** Fornire ambiente containerizzato completo con Docker Compose

## II - 2. Funzionalità e vincoli del prodotto

### II - 2.1. Funzionalità implementate:

#### Data Pipeline e Streaming:

- Producer Kafka che simula movimenti utenti su percorsi reali di Milano usando OSRM
- Consumer Bytewax per elaborazione stream dati con arricchimento in tempo reale
- Integrazione con PostGIS per calcolo distanze e identificazione negozi vicini

#### Generazione Messaggi Intelligente:

- Servizio microservice FastAPI per generazione messaggi personalizzati
- Integrazione con LLM (supporto Groq, OpenAI, provider multipli)
- Sistema di cache Redis per ottimizzazione performance e riduzione costi
- Template personalizzabili basati su profilo utente e tipologia negozio

#### Dashboard Utente:

- Interfaccia web responsive con mappa Leaflet interattiva
- Visualizzazione real-time di posizione utente, percorsi e negozi
- Sistema di notifiche in-app per messaggi personalizzati
- Autenticazione JWT con gestione sessioni sicura
- WebSocket per aggiornamenti posizione in tempo reale

#### Storage e Persistenza:

- ClickHouse per analytics e storage eventi temporali ad alta performance
- PostgreSQL/PostGIS per gestione dati geospaziali dei negozi
- Schema ottimizzato per query geospaziali e analytics

#### ETL e Data Management:

- Apache Airflow per orchestrazione ETL automatici
- ETL adattivo che si auto-configura in base ai cambiamenti nei dati

- Integrazione con OpenStreetMap per aggiornamento negozi Milano
- Sistema di tracking cambiamenti con logging completo

#### Monitoring e Observability:

- Dashboard Grafana con pannelli personalizzati per metriche business e sistema
- Prometheus per raccolta metriche con esportatori per tutti i componenti
- Loki per gestione log centralizzata
- Metriche custom per tracking performance LLM, cache hit rate, eventi utente

#### II - 2.2. Vincoli tecnici:

- **Geografici:** Sistema focalizzato su Milano (coordinate definite in configurazione)
- **Performance:** Target <200ms per generazione messaggi (con cache)
- **Scalabilità:** Supporto multi-worker Bytewax per elaborazione parallela
- **Sicurezza:** Kafka con SSL/TLS, JWT per autenticazione, password protette
- **Disponibilità:** Health checks e restart automatici per tutti i servizi

### II - 3. Utenti e loro caratteristiche

#### II - 3.1. Tipologie di utenti:

##### Utente finale:

- **Descrizione:** Persona che utilizza l'applicazione per ricevere notifiche personalizzate
- **Caratteristiche:** Ha un profilo definito (età, professione, interessi) memorizzato nel sistema
- **Interazioni:** Accede alla dashboard, visualizza la mappa, riceve notifiche, naviga lo storico
- **Competenze richieste:** Utilizzo base di applicazioni web e mobile

##### Amministratore di sistema:

- **Descrizione:** Operatore responsabile del monitoraggio e manutenzione del sistema
- **Caratteristiche:** Competenze tecniche per interpretare metriche e log
- **Interazioni:** Accesso a Grafana, Prometheus, log Loki, configurazione Airflow
- **Competenze richieste:** Conoscenza sistemi distribuiti, database, monitoring

##### Sviluppatore/DevOps:

- **Descrizione:** Persona che sviluppa, deploy e mantiene il sistema
- **Caratteristiche:** Competenze tecniche avanzate su architetture microservizi
- **Interazioni:** Deploy con Docker Compose, configurazione environment, debug
- **Competenze richieste:** Docker, Kafka, database, cloud native technologies

## III. Architettura del sistema

Il sistema NearYou implementa un'**architettura microservizi event-driven** con i seguenti componenti principali:

### III - 1. Componenti Core

#### Streaming Data Pipeline:

- **Apache Kafka:** Message broker per streaming eventi GPS con SSL
- **Bytewax:** Stream processor per elaborazione real-time con supporto multi-worker

- **Producer:** Simulatore movimenti utenti su percorsi OSRM Milano
- **Consumer:** Processore eventi con arricchimento dati e generazione messaggi

**API Services:**

- **Message Generator:** Microservizio FastAPI per generazione messaggi LLM
- **Dashboard Service:** Servizio FastAPI per dashboard utente con WebSocket
- **Authentication:** Sistema JWT per gestione autenticazione e autorizzazione

**Storage Layer:**

- **ClickHouse:** Database colonnare per analytics e storage eventi temporali
- **PostgreSQL/PostGIS:** Database relazionale con estensioni geospaziali per negozi
- **Redis:** Cache distribuita per messaggi LLM e sessioni

**ETL e Data Management:**

- **Apache Airflow:** Orchestratore per ETL automatici da OpenStreetMap
- **OSRM:** Servizio routing per calcolo percorsi realistici su Milano

**Monitoring Stack:**

- **Grafana:** Dashboard per visualizzazione metriche business e sistema
- **Prometheus:** Raccolta metriche con esportatori specializzati
- **Loki:** Gestione log centralizzata
- **Exporters:** Componenti per esposizione metriche database e servizi

## III - 2. Flusso Dati

1. **Generazione Eventi:** Producer simula movimenti utenti su percorsi OSRM
2. **Stream Processing:** Bytewax consumer elabora eventi, arricchisce con dati negozi
3. **Proximity Detection:** Calcolo distanze con PostGIS, trigger per messaggi
4. **Message Generation:** Chiamata servizio LLM con cache Redis
5. **Storage:** Persistenza eventi arricchiti in ClickHouse
6. **Real-time Updates:** Notifica utenti via WebSocket
7. **Analytics:** Visualizzazione dati in dashboard Grafana

## IV. Casi d'uso

### IV - 1. Introduzione

I casi d'uso del sistema NearYou sono strutturati attorno alle funzionalità core implementate nel codice sorgente. Ogni caso d'uso è identificato da un codice univoco UC[numero] e descrive le interazioni tra attori e sistema.

### IV - 2. Attori

**Utente Autenticato:** Persona che utilizza la dashboard per visualizzare la propria posizione e ricevere notifiche personalizzate, avendo eseguito l'accesso.

**Utente non Autenticato:** Utente che non ha ancora eseguito l'accesso.

**Amministratore:** Persona che usa la dashboard per visualizzare le analisi di marketing e di monitoraggio del sistema.