

# Implementazione di un Assistente AI per il Supporto Tecnico Konsolex

---

Università degli Studi di Genova

Corso di Laurea Informatica ordinamento tecnologico

---

## Tesi di Laurea

### Candidato:

Sebastiano Gastaldi

### Relatore

Professor Stefano Rovetta

### Anno Accademico:

2024/2025

---

## Codice Sorgente del Progetto

Il codice sorgente completo del progetto, insieme ai file citati nella tesi, è disponibile nel repository GitHub:

<https://github.com/Gaseb86/tirocinio.git>

---

## Indice

### Capitolo 1: Introduzione e Contesto

- 1.1 Abstract del Progetto
- 1.2 Contesto Aziendale: OnTheCloud e Konsolex
- 1.3 Sfide del Supporto Tecnico nel Cloud B2B
- 1.4 Vantaggi dell'Automazione tramite AI
- 1.5 Obiettivi del Progetto

### Capitolo 2: Requisiti di Sistema

- 2.1 Requisiti Funzionali
  - 2.1.1 Gestione Automatica delle Richieste di Supporto Tecnico
  - 2.1.2 Interfaccia Conversazionale Multicanale
  - 2.1.3 Sistema di Knowledge Base per Risposte Tecniche
  - 2.1.4 Meccanismo di Escalation verso Operatori Umani attraverso sistema di ticket
  - 2.1.5 Operazioni Tecniche Eseguibili tramite AI
- 2.2 Requisiti Non Funzionali
  - 2.2.1 Performance
  - 2.2.2 Scalabilità
  - 2.2.3 Affidabilità

- 2.2.4 Sicurezza
- 2.2.5 Usabilità
- 2.2.6 Manutenibilità
- 2.2.7 Integrazione con Sistemi Esistenti

## Capitolo 3: Casi d'Uso

- InterazioneBaseUtenteChatBot
- CreaNuovoUtente
- GenerazioneRispostaAI
- EsecuzioneFunzioniSpecializzate
- InoltroMessaggioTicketAperto
- AperturaTicketSupporto
- RispostaAmministratoreATicket
- ChiusuraTicketSupporto

## Capitolo 4: Scelte Tecnologiche

- 4.1 OpenAI GPT-4: Cuore dell'Intelligenza Conversazionale
- 4.2 Stack tecnologico preesistente: TypeScript, Node.js, Express
  - 4.2.2 Node.js ed Express.js
- 4.3 Database e ORM Sequelize
- 4.4 Integrazione con API esterne
  - 4.4.1 Telegram Bot API
  - 4.4.2 Konsolex API
- 4.5 Considerazioni finali

## Capitolo 5: Architettura del Sistema

- 5.1 Overview dell'Architettura Complessiva
- 5.2 Moduli Principali e Loro Interazioni
- 5.3 Diagrammi Architetture
  - 5.3.1 Class Diagram
  - 5.3.2 Component Diagram
  - 5.3.3 Deployment Diagram
  - 5.3.4 Diagramma ER del Database
- 5.4 Flussi di Dati e Comandi
- 5.5 Conclusioni Architetture

## Capitolo 6: Componenti Principali

- 6.1 Web Server & REST API
  - 6.1.1 Architettura Generale
  - 6.1.2 Endpoint per Utenti Finali
  - 6.1.3 Endpoint Amministrativi
  - 6.1.4 Integrazione con Konsolex
  - 6.1.5 Gestione Errori e Resilienza
- 6.2 OpenAI Handler

- 6.2.1 Gestione Thread e Assistenti
  - 6.2.2 Function Calling
  - 6.2.3 Ottimizzazione delle Risposte
  - 6.2.4 Resilienza e Gestione Errori
- 6.3 Telegram Bot
  - 6.3.1 Struttura e Inizializzazione
  - 6.3.2 Gestione delle Conversazioni
- 6.4 Endpoint Handler per Konsolex
  - 6.4.1 Architettura dell'Integrazione
  - 6.4.2 Operazioni Server
  - 6.4.3 Gestione Domini e DNS
  - 6.4.4 Container e Database
  - 6.4.5 Autenticazione e Verifica Utenti
- 6.5 Sistema di Ticketing
  - 6.5.1 Architettura del Sistema
  - 6.5.2 Creazione e Gestione Ticket
  - 6.5.3 Interfaccia per Operatori
  - 6.5.4 Notifiche e Aggiornamenti
- 6.6 Repository Pattern per Persistenza Dati
  - 6.6.1 Struttura dei Repository

## Capitolo 7: Gestione delle Conversazioni con il chatbot AI

- 7.1 OpenAI Thread Management
  - 7.1.1 Architettura dei Thread
  - 7.1.2 Ciclo di Vita dei Thread
- 7.2 Contestualizzazione delle Conversazioni
  - 7.2.1 Strategie di Contestualizzazione
  - 7.2.2 Persistenza del Contesto
- 7.3 Function Calling con OpenAI
  - 7.3.1 Architettura del Function Calling
  - 7.3.2 Funzioni Implementate
  - 7.3.3 Sicurezza e Validazione
- 7.4 Diagrammi Comportamentali
  - 7.4.1 Sequence Diagram per Flusso Conversazionale
  - 7.4.2 Activity Diagram per Elaborazione Messaggi
- 7.5 Ottimizzazione Parametri per Qualità Risposte
  - 7.5.1 Configurazione del prompt
  - 7.5.2 Configurazione dei Parametri di GPT-4
  - 7.5.3 Configurazione dei parametri dei file nel vectorStore
  - 7.5.4 Post-Elaborazione delle Risposte
  - 7.5.5 Score e Miglioramento Continuo
- 7.6 Meccanismi di Timeout e Recovery
  - 7.6.1 Rilevamento e Gestione Timeout
  - 7.6.2 Strategie di Recovery
- 7.7 Conclusioni sulla Gestione delle Conversazioni

## Capitolo 8: Il sistema di Function Calling di OpenAI

- 8.1 Architettura del Function Calling
  - 8.1.1 Principi di Funzionamento
  - 8.1.2 Gestione del Ciclo di Richiesta-Esecuzione
  - 8.1.3 Dichiarazione e Definizione delle Funzioni
- 8.2 Implementazione Tecnica delle Function Calls
  - 8.2.1 Gestione dell'Esecuzione
  - 8.2.2 Integrazione con il Flusso Conversazionale
- 8.3 Estensibilità del Sistema di Funzioni
  - 8.3.1 Aggiunta di Nuove Funzioni
  - 8.3.2 Versioning delle Funzioni
- 8.4 Sicurezza e Validazione nelle Function Calls
  - 8.4.1 Validazione dei Parametri
  - 8.4.2 Autorizzazione delle Operazioni
- 8.5 Panoramica delle Principali Funzionalità Implementate
- 8.6 Conclusioni

## Capitolo 9: Sistema di Ticketing

- 9.1 Rilevamento di Problematiche Non Risolvibili dall'AI
  - 9.1.1 Valutazione della Qualità delle Risposte
  - 9.1.2 Processo di Escalation
- 9.2 Creazione e Routing dei Ticket
  - 9.2.1 Struttura e Creazione del Ticket
  - 9.2.2 Meccanismo di Caching per Performance
  - 9.2.3 Routing verso gli Amministratori
- 9.3 Interfaccia Amministrativa
  - 9.3.1 Interfaccia Telegram
  - 9.3.2 Dashboard Web Konsolex
  - 9.3.3 Visualizzazione del Contesto
- 9.4 Diagrammi dei Processi
  - 9.4.1 Activity Diagram creazione del Ticket
  - 9.4.2 Sequence Diagram per Gestione Ticket tra AI e Operatori
- 9.5 Meccanismi di Chiusura
- 9.6 Conclusioni sul Sistema di Ticketing

## Capitolo 10: Conclusioni e Sviluppi Futuri

- 10.1 Riepilogo e Risultati Raggiunti
  - 10.1.1 Obiettivi Conseguiti
  - 10.1.2 Innovazioni Introdotte
- 10.2 Analisi Critica e Limitazioni
  - 10.2.1 Punti di Forza
  - 10.2.2 Limitazioni Attuali
- 10.3 Direzioni di Sviluppo Futuro
  - 10.3.1 Evoluzione Tecnica
  - 10.3.2 Miglioramenti dell'Esperienza Utente

- 10.4 Impatto Aziendale
- 10.5 Riflessioni Conclusive
- 10.6 Ringraziamenti