



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Informatica

A.a. 2023/2024

Sessione di laurea Ottobre 2024

**Un approccio low-code e
conversazionale per l'interrogazione
di documenti aziendali**

Relatori:

Luigi De Russis

Giovanni Campolo

Candidato:

Fabio Mangani

Sommario

Questo progetto di tesi affronta il tema dell'uso di tecnologie low-code e intelligenza artificiale per migliorare l'interrogazione e la gestione dei documenti aziendali confidenziali. L'obiettivo principale è sviluppare un chatbot avanzato che permetta agli utenti di accedere facilmente a informazioni archiviate su piattaforme aziendali come SharePoint, utilizzando un'interfaccia conversazionale e tecniche di elaborazione del linguaggio naturale.

Il lavoro analizza le tecnologie impiegate, tra cui Microsoft Power Platform, che include strumenti quali Power Apps e Power Automate, integrati con servizi di Azure per potenziare le capacità di ricerca e risposta del chatbot. Si esplora come queste tecnologie consentano di costruire soluzioni scalabili, sicure e facili da implementare, riducendo la necessità di sviluppo tradizionale grazie all'approccio low-code.

Vengono approfondite le sfide legate alla gestione dei documenti aziendali, evidenziando i limiti delle attuali soluzioni di ricerca e le difficoltà nell'accedere rapidamente a informazioni distribuite su diverse piattaforme. La tesi dimostra come il chatbot, sfruttando l'intelligenza artificiale, possa superare queste problematiche fornendo risposte precise e contestuali, migliorando l'efficienza e supportando la presa di decisioni.

L'implementazione del chatbot si basa su un'architettura integrata che combina la potenza di Microsoft Copilot Studio con flussi automatizzati per la gestione e l'accesso ai dati. La soluzione è progettata per garantire la sicurezza e la conformità alle normative, rispettando criteri rigorosi di accesso e protezione delle informazioni sensibili. La tesi si conclude con una valutazione dei risultati ottenuti, evidenziando i benefici pratici, l'efficacia del sistema e le potenzialità future di miglioramento e applicazione commerciale della tecnologia sviluppata.

Ringraziamenti

Desidero ringraziare voi, i miei genitori, per il vostro supporto instancabile durante questi anni. Il vostro amore e la vostra presenza costante sono stati il fondamento di ogni mio passo avanti. Mi avete insegnato a credere in me stesso, ad affrontare le sfide con coraggio e a non arrendermi mai. Senza il vostro esempio e la vostra forza, non sarei riuscito a raggiungere questo traguardo. Vi sono immensamente grato per tutto quello che avete fatto per me, per ogni sacrificio, ogni incoraggiamento e ogni gesto d'amore che mi ha permesso di arrivare fin qui.

Ringrazio tutta la mia famiglia: nonni, zii e cugini, che mi avete fatto sentire sempre amato e sostenuto. Il vostro affetto e la vostra vicinanza mi hanno dato forza e serenità in ogni momento, e vi sono grato per aver sempre creduto in me. Un pensiero speciale va a te, mia cugina Maria Rita, con cui sono praticamente cresciuto. Non sei stata solo una cugina, ma una vera e propria sorella. Sei stata la prima a credere in me, nelle mie capacità, anche quando io stesso avevo forti dubbi. La tua fiducia, il tuo sostegno e il tuo affetto incondizionato sono stati una luce nei momenti più difficili, e la tua presenza ha reso tutto più sopportabile. Sei una parte fondamentale della mia vita, e lo sarai sempre.

Desidero ringraziare in modo speciale te, Alice (o Bobbie, come mi piace chiamarti), la mia migliore amica da ormai dieci anni. Sei una delle persone più importanti della mia vita e la tua amicizia è per me un dono prezioso. Nonostante la distanza, sei sempre stata presente, ascoltandomi, incoraggiandomi e sostenendomi ogni volta che ne avevo bisogno. Il tuo affetto ha superato ogni chilometro e mi ha fatto sentire meno solo, ogni giorno. Non posso immaginare questi anni senza il tuo costante sostegno e la tua capacità di portare luce anche nei momenti più bui.

Desidero ringraziare i miei colleghi Francesco, Gabriele, Michele, Edoardo, Jacopo e Andrea: siete stati molto più di semplici compagni di studio. Anche se alle lezioni eravate più assenti che presenti, la vostra amicizia ha reso questi anni indimenticabili e pieni di risate.

Desidero ringraziare i miei amici, tutti voi della Federazione: la vostra compagnia mi ha dato energia e gioia, e il pensiero di tornare a casa per stare insieme a voi mi ha sempre scaldato il cuore. Un ringraziamento particolare va ai pizzitani, che ad ogni occasione e festa mi hanno reso immensamente felice di tornare a casa per

passare del tempo con loro, rendendo ogni ritorno un momento speciale.

Vorrei anche ringraziare i miei compagni di DnD: Emanuele, Enrico, Filippo, Michele e Davide. Con voi ho passato serate piene di risate e spensieratezza, momenti che hanno portato leggerezza nelle mie giornate e che ricorderò sempre con affetto, soprattutto ricordando che non proverò mai più a costruire una zattera in una tempesta!

A tutti voi, grazie di cuore. Senza di voi, questo percorso non sarebbe stato lo stesso.

Indice

Elenco delle figure	VIII
1 Introduzione	1
1.1 Obiettivo della Tesi	4
1.2 Struttura della tesi	6
2 Panoramica delle Tecnologie Impiegate	7
2.1 Low-code	7
2.2 Power Platform	10
2.2.1 Dataverse	12
2.3 Power Apps	13
2.3.1 Tipologie di Applicazioni	14
2.3.2 Interfaccia Grafica e Componenti delle Model-Driven App	14
2.3.3 Vantaggi, Casi d'uso e Limit	15
2.4 Power Automate	16
2.4.1 Il Concetto di Flow	17
2.4.2 Creazione di un Flow	17
2.4.3 Esecuzione e Monitoraggio	18
2.4.4 Casi d'Uso e Scenari	19
2.4.5 Vantaggi e Svantaggi	19
2.5 Microsoft Copilot Studio	20
2.5.1 Il Ruolo dell'Intelligenza Artificiale	21
2.5.2 Topic	22
2.5.3 Plugin Actions	23
2.5.4 Canali di Comunicazione e Analisi delle Prestazioni	24
2.6 Servizi Azure	25
2.6.1 Azure AI Search	26
2.6.2 Azure OpenAI	27
2.6.3 Direct Line	28
2.7 SharePoint	28

3	Progettazione del Chatbot	30
3.1	Descrizione Informale	31
3.2	Requisiti	32
3.3	Architettura Monolitica	33
3.3.1	Descrizione Tecnica	33
3.3.2	Vantaggi dell'Architettura	34
3.3.3	Limitazioni dell'Architettura	35
3.4	Architettura Ibrida	36
3.4.1	Analisi Economica	36
3.4.2	Descrizione Tecnica	37
3.4.3	Vantaggi e Limitazioni	38
3.5	Progettazione dell'Interfaccia di Configurazione	39
3.5.1	Scelta dell'approccio	39
3.5.2	Progettazione della Struttura Dati	40
4	Implementazione del Chatbot	42
4.1	Configurazione dell'Indicizzazione dei Documenti su SharePoint	42
4.1.1	Connessione a SharePoint	43
4.1.2	Creazione della Sorgente dei Dati	44
4.1.3	Creazione dell'Indice	44
4.1.4	Configurazione dell'Indicizzatore	46
4.1.5	Aggiornamento della Sorgente dei Dati	46
4.2	Configurazione del Chatbot	47
4.2.1	Scelta del modello	48
4.2.2	Configurazione della Ricerca	49
4.3	Integrazione tramite Power Automate	50
4.3.1	Configurazione di Microsoft Copilot Studio	50
4.3.2	Flusso di Lavoro su Power Automate	51
4.3.3	Elaborazione della Risposta	53
4.4	Gestione degli Accessi e Sicurezza	53
4.4.1	Introduzione a OData	54
4.4.2	Controlli di Accesso tramite Dataverse	54
4.4.3	Controllo degli Accessi con OData	56
4.5	Distribuzione del Chatbot	57
4.5.1	Distribuzione su Microsoft Teams	58
4.5.2	Distribuzione su WhatsApp	59
5	Risultati e Valutazione	65
5.1	Ambiente di Test e Metodologia	65
5.1.1	Ambiente di Test	65
5.1.2	Metriche valutate	66

5.1.3	Tipologia di test eseguiti	67
5.2	Risultati dei Test	68
5.2.1	Precisione delle risposte	69
5.2.2	Velocità di risposta	70
5.2.3	Scalabilità	71
5.3	Valutazione Economica e Competitività Commerciale	71
5.4	Sicurezza e Conformità alla Normativa	72
5.5	Miglioramenti Futuri	74
6	Conclusione	77
	Bibliografia	79

Elenco delle figure

2.1	Le caratteristiche dello sviluppo Low-Code	8
2.2	Struttura della Microsoft Power Platform	11
2.3	Flusso in Power Automate	18
2.4	Topic di Microsoft Copilot Studio	23
2.5	Azure AI Services	26
3.1	Architettura Monolitica basata su Microsoft Copilot Studio	34
3.2	Architettura Ibrida con integrazione di Power Automate e Azure AI Search	38
3.3	Diagramma UML	40
3.4	Schermata di Power Apps per configurare i Contatti	41
4.1	Richiesta HTTP al chatbot	52
4.2	Chat su Microsoft Teams	58
4.3	Chat su WhatsApp	60
5.1	Grafico della Precisione delle risposte	69
5.2	Grafico della velocità di Risposta	70

Capitolo 1

Introduzione

Nel panorama contemporaneo, caratterizzato da un rapido progresso tecnologico e da un incremento esponenziale nella digitalizzazione dei processi aziendali, la gestione delle informazioni confidenziali e l'accesso efficiente a queste informazioni sono diventati aspetti fondamentali per la competitività delle imprese. Come evidenziato nel report di Gartner sui dieci trend tecnologici strategici per il 2023, l'aumento del volume e della complessità dei dati è una delle sfide principali che le organizzazioni devono affrontare: le aziende oggi si trovano a gestire una quantità senza precedenti di dati, provenienti da fonti diverse come dispositivi IoT, applicazioni e social media, il che porta a una crescita esponenziale dei dati da elaborare e proteggere [1]. La complessità nella gestione dei dati è amplificata da vari fattori: le organizzazioni devono affrontare difficoltà significative nell'integrare dati distribuiti su piattaforme eterogenee, spesso caratterizzati da diversi livelli di riservatezza e sicurezza. Questo può creare silos informativi e rendere difficile ottenere una visione unificata e coerente delle informazioni, compromettendo l'efficienza operativa e la capacità di prendere decisioni informate. La sicurezza dei dati emerge come una preoccupazione cruciale, soprattutto alla luce delle recenti evoluzioni tecnologiche e delle normative sempre più stringenti: le violazioni dei dati e le non conformità alle regolazioni come il GDPR possono comportare gravi conseguenze, tra cui danni reputazionali e sanzioni economiche, pertanto, le aziende devono adottare tecnologie avanzate per garantire che le informazioni sensibili siano protette da accessi non autorizzati e siano conformi alle normative vigenti. Inoltre, la crescente complessità dei dati richiede l'adozione di soluzioni tecnologiche innovative e le tecnologie emergenti, come l'intelligenza artificiale e i sistemi di gestione dei dati avanzati, sono essenziali per affrontare le sfide legate all'integrazione, alla sicurezza e all'accessibilità dei dati. L'implementazione di queste tecnologie può migliorare significativamente la capacità delle organizzazioni di gestire e utilizzare i dati in modo efficace, mantenendo così un vantaggio competitivo nel mercato in continua evoluzione [2].

Sfide della Gestione Documentale

Le sfide legate alla gestione delle informazioni aziendali sono complesse e multifattoriali, toccano non solo la sicurezza dei dati, ma anche la loro accessibilità e fruibilità da parte degli utenti. In un contesto aziendale sempre più digitale e data-driven, la quantità di informazioni generate e archiviate cresce esponenzialmente, creando un ambiente dove la frammentazione dei dati è all'ordine del giorno. Questa frammentazione deriva dal fatto che le informazioni sono spesso distribuite su diverse piattaforme, archivi di documenti e strumenti, ciascuno con le proprie logiche di organizzazione e accesso: un esempio comune di questa situazione si verifica con l'uso di piattaforme di gestione documentale, come SharePoint, la quale, sebbene offra potenti strumenti per l'archiviazione e la condivisione di documenti all'interno di un'organizzazione, la ricerca e l'accesso alle informazioni specifiche possono risultare estremamente complicati. Quanto detto è particolarmente vero quando i dati sono sparsi su repository diversi, strutturati in modi non coerenti, o quando le informazioni pertinenti si trovano all'interno di documenti lunghi e complessi; di conseguenza, i dipendenti spesso spendono una quantità significativa di tempo cercando di localizzare le informazioni di cui hanno bisogno, riducendo l'efficienza operativa e aumentando il rischio di errori [3].

Una delle principali difficoltà in questo contesto è la capacità di identificare con precisione dove si trova una specifica informazione all'interno di un documento, o persino di determinare quale documento contenga l'informazione desiderata [4]. Questa problematica diventa ancora più rilevante quando si tratta di documenti aziendali confidenziali o privati, ovvero tutti quei dati che, se divulgati, potrebbero causare danni all'azienda, come informazioni finanziarie riservate, dati personali dei clienti, segreti industriali o proprietà intellettuale, i quali non possono ovviamente essere indicizzati o ricercati tramite motori di ricerca pubblici o tramite intelligenze artificiali progettate per lavorare su dati accessibili pubblicamente. A differenza delle informazioni disponibili pubblicamente sul web, che possono essere facilmente indicizzate e recuperate da strumenti di ricerca generici, i documenti aziendali confidenziali richiedono metodi di ricerca che garantiscono non solo l'efficacia e la precisione, ma anche la sicurezza e la riservatezza. Inoltre, molti sistemi di gestione documentale tradizionali, come il già citato SharePoint, si limitano a funzioni di ricerca basate su parole chiave, che spesso non sono sufficienti in contesti aziendali complessi; le informazioni rilevanti per l'utente potrebbero non corrispondere esattamente ai termini cercati, specialmente se quest'ultime sono state redatte in un linguaggio tecnico o specialistico, o se si tratta di dettagli sepolti in lunghe discussioni o report. Questo limita l'efficacia della ricerca e può portare a situazioni in cui informazioni critiche non vengono individuate o utilizzate correttamente, con potenziali conseguenze negative per le decisioni aziendali.

Queste sfide evidenziano la necessità di soluzioni più avanzate e specifiche, che non solo migliorino l'efficacia dell'accesso alle informazioni, ma che lo facciano

in un modo sicuro e conforme alle necessità aziendali. In particolare, vi è un crescente bisogno di strumenti che siano in grado di interrogare documenti aziendali riservati, comprendere il contesto delle domande poste dagli utenti e restituire risposte pertinenti e accurate, indipendentemente da dove tali informazioni siano memorizzate. In questo contesto, l'adozione di chatbot basati su intelligenza artificiale rappresenta una risposta innovativa e potenzialmente risolutiva.

Perchè un Chatbot?

La scelta di adottare un chatbot come soluzione per la gestione delle informazioni aziendali risponde a diverse esigenze chiave che le imprese moderne devono affrontare. In primo luogo, i chatbot offrono un'interfaccia di comunicazione intuitiva e accessibile, che può essere utilizzata da utenti con diversi livelli di competenza tecnica. Questo è particolarmente importante in un contesto aziendale, dove l'accesso rapido e preciso alle informazioni è cruciale per mantenere l'efficienza operativa, i chatbot, infatti, sono in grado di automatizzare le risposte alle domande più comuni, riducendo il carico di lavoro dei team di supporto e migliorando l'efficacia del servizio [5]. Un aspetto fondamentale che rende i chatbot particolarmente efficaci è la loro capacità di operare in tempo reale: in un ambiente aziendale, la velocità con cui le informazioni possono essere recuperate e utilizzate spesso fa la differenza tra il successo e il fallimento di una decisione strategica. I chatbot possono rispondere immediatamente a richieste complesse, accedendo a database o archivi di documenti per fornire le informazioni richieste in pochi secondi. Questo livello di reattività è difficile da ottenere con altri strumenti tradizionali di gestione documentale, che spesso richiedono tempi di ricerca più lunghi e competenze specifiche per navigare tra i vari repository. Un'altra ragione per l'adozione di chatbot è la loro flessibilità e scalabilità. I chatbot possono essere facilmente integrati con diverse piattaforme aziendali, come: sistemi di gestione documentale, CRM (Customer Relationship Management), ovvero sistemi per la gestione delle relazioni con i clienti, e anche per piattaforme di comunicazione. Questo significa che possono essere adattati per rispondere a una vasta gamma di esigenze aziendali, migliorando la coesione e l'integrazione dei vari sistemi informativi. Inoltre, i chatbot sono scalabili: possono essere implementati per servire una piccola squadra o un'intera organizzazione, con la possibilità di espandersi man mano che le esigenze dell'azienda crescono.

Il Linguaggio Naturale

L'adozione del linguaggio naturale (Natural Language Processing, NLP) nei chatbot rappresenta un'importante evoluzione rispetto agli strumenti tradizionali di ricerca e gestione documentale. Il linguaggio naturale è il modo più diretto e intuitivo per gli esseri umani di comunicare, ed è per questo che l'integrazione dell'NLP in un chatbot rende l'interazione con i sistemi informativi aziendali più fluida e user-friendly. A differenza dei sistemi di ricerca basati su parole chiave, che richiedono

all'utente di sapere esattamente cosa cercare, i chatbot dotati di NLP possono comprendere richieste formulate in modo naturale, come farebbe un essere umano, e interpretarle per restituire risposte pertinenti [5]. Questo è particolarmente utile nelle aziende dove le informazioni potrebbero non essere chiaramente etichettate o dove gli utenti non sono sempre familiari con la terminologia specifica utilizzata nei documenti. Ad esempio, un dipendente potrebbe chiedere al chatbot "Dove posso trovare il report annuale sulle vendite?" senza conoscere l'esatta denominazione del documento o il percorso di archiviazione. Grazie alle capacità di NLP, il chatbot può interpretare questa richiesta, cercare nei documenti pertinenti e fornire la risposta corretta, facilitando così l'accesso alle informazioni necessarie. Un ulteriore vantaggio cruciale dell'NLP è la sua capacità di gestire l'ambiguità e il contesto delle richieste degli utenti. Spesso, i sistemi di ricerca tradizionali faticano a distinguere tra richieste simili o a comprendere il significato esatto di una domanda quando il contesto non è chiaramente definito. I chatbot, invece, sono in grado di analizzare il contesto della conversazione, disambiguare i termini e fornire risposte che siano non solo corrette, ma anche rilevanti per la situazione specifica dell'utente. Questo migliora significativamente l'esperienza dell'utente, riducendo i tempi di ricerca e aumentando la precisione delle risposte [6].

1.1 Obiettivo della Tesi

L'obiettivo principale di questo lavoro di tesi è la progettazione, implementazione e validazione di un chatbot destinato a interrogare documenti confidenziali archiviati online, utilizzando un approccio low-code per accelerare il processo e garantire maggiore flessibilità. Il lavoro di tesi, sviluppato presso l'azienda Cluster Reply DCX, presenta un chatbot che si distingue per la sua capacità di comprendere il linguaggio naturale, permettendo agli utenti autorizzati di effettuare interrogazioni complesse e ricevere risposte precise sulla base del contenuto dei documenti. Esso è progettato non solo per operare efficacemente con un insieme di dati privati e confidenziali, spesso non indicizzati e caratterizzati da notevole eterogeneità, ma anche per soddisfare specifici criteri di efficacia e a valutare la competitività commerciale della soluzione nel mercato attuale.

Il successo del chatbot si basa fortemente sulla capacità di soddisfare criteri ben definiti di efficacia. Questi criteri non solo assicurano la funzionalità tecnica del sistema, ma ne garantiscono anche il valore aggiunto nell'ambiente aziendale. Di seguito, vengono illustrati i principali requisiti tecnici di efficacia che abbiamo mirato a garantire nella progettazione del chatbot:

- *Precisione delle Risposte*: Il chatbot utilizza avanzati algoritmi di elaborazione del linguaggio naturale per interpretare accuratamente le query degli utenti e fornire risposte pertinenti. Questo livello di accuratezza è cruciale per garantire

che gli utenti ricevano informazioni rilevanti e corrette, riducendo il rischio di errori e migliorando la qualità complessiva del servizio offerto.

- *Velocità di Risposta*: La rapidità con cui il chatbot è in grado di elaborare e rispondere alle richieste degli utenti è un fattore determinante per migliorare la produttività aziendale. In un contesto dove le decisioni devono essere prese rapidamente e sulla base di informazioni precise, un chatbot che fornisce risposte immediate permette agli utenti di accedere ai dati necessari senza ritardi, ottimizzando così il tempo speso in attività critiche. Questa velocità contribuisce a ridurre significativamente i tempi di attesa e permette ai dipendenti di completare le loro operazioni con maggiore efficienza.
- *Scalabilità*: La capacità del chatbot di adattarsi a un crescente volume di dati e richieste è fondamentale in un ambiente aziendale in continua evoluzione. Il sistema è stato progettato per supportare un'espansione progressiva sia in termini di utenti che di carico di lavoro, garantendo prestazioni costanti anche in situazioni di utilizzo intensivo. Ciò permette all'azienda di scalare il servizio in modo efficiente, senza dover rivedere l'architettura di base del sistema o sostenere costi elevati di implementazione.

Un aspetto cruciale del progetto è la valutazione preliminare del potenziale commerciale della soluzione proposta. Durante le fasi iniziali, sono stati esaminati in dettaglio i costi di attivazione e manutenzione, con l'obiettivo di garantire che il chatbot non solo risponda ai requisiti tecnici di efficacia, ma si riveli anche economicamente vantaggioso rispetto ad altre soluzioni già presenti sul mercato. Questa valutazione aiuterà a determinare la competitività della soluzione sul mercato, identificando i punti di forza e le aree di miglioramento potenziale.

La privacy e la sicurezza delle informazioni trattate rappresentano un altro pilastro fondamentale del progetto. È imperativo che il chatbot incorpori meccanismi di sicurezza avanzati per autenticare gli utenti e limitare l'accesso alle informazioni sensibili, assicurando che solo il personale autorizzato possa consultare i dati. Ciò è particolarmente rilevante in un ambiente aziendale dove la protezione dei dati sensibili è soggetta a rigorose normative, come il GDPR. Inoltre, per prevenire l'utilizzo improprio dei dati, le interazioni con il chatbot non saranno utilizzate per alimentare ulteriori modelli di machine learning, proteggendo così la riservatezza e l'integrità dei dati aziendali.

L'approccio scelto per lo sviluppo del chatbot è il low-Code, sfruttando le tecnologie della Power Platform di Microsoft e i servizi Azure. Questo approccio è finalizzato a semplificare il processo di sviluppo, riducendo il tempo e i costi associati, e a facilitare la creazione di soluzioni robuste e personalizzabili. L'utilizzo di tali tecnologie mira a garantire che il chatbot possa essere integrato agevolmente in ambienti aziendali esistenti, aumentando la flessibilità e l'efficacia della soluzione.

1.2 Struttura della tesi

Nei capitoli che seguono, verrà presentata un'esposizione dettagliata delle fasi di sviluppo del progetto, articolata come segue:

- *Capitolo 2 “Panoramica delle Tecnologie Impiegate”*: Saranno illustrate le tecnologie impiegate per la realizzazione del progetto, con particolare riferimento alla metodologia low-code. Verranno analizzati i principali strumenti della Power Platform e alcuni servizi Azure utilizzati per implementare le soluzioni.
- *Capitolo 3 “Progettazione del Chatbot”*: Si procederà alla fase di progettazione, con una dettagliata definizione dei requisiti funzionali e non funzionali del progetto. Successivamente, saranno esaminate e confrontate diverse proposte architettoniche per individuare la soluzione più idonea.
- *Capitolo 4 “Implementazione del Chatbot”*: Questo capitolo descrive i principali passi compiuti per l'implementazione del chatbot. Si parte dalla configurazione e indicizzazione dei documenti su SharePoint, passando per l'integrazione con Azure AI Search e Azure OpenAI Studio per la gestione dell'intelligenza artificiale. Vengono poi trattate le soluzioni adottate per interfacciare il sistema con Microsoft Copilot Studio e per garantire la sicurezza e la gestione degli accessi. Infine, il capitolo si conclude con la descrizione della distribuzione del chatbot su piattaforme di comunicazione come Microsoft Teams e WhatsApp.
- *Capitolo 5 “Risultati e Valutazione”*: In questo capitolo vengono presentati i risultati ottenuti dal progetto. Saranno illustrati i test eseguiti per validare il corretto funzionamento del chatbot, con particolare attenzione alla precisione delle risposte, velocità di elaborazione e scalabilità del sistema in ambienti aziendali complessi. Inoltre, verrà valutato il potenziale commerciale della soluzione, insieme alle misure di sicurezza adottate per garantire la protezione dei dati. Infine, saranno discusse le possibili aree di miglioramento e le future evoluzioni del sistema.
- *Capitolo 6 “Conclusione”*: In questo capitolo vengono presentate le conclusioni finali del lavoro, discutendo le sfide affrontate, le opportunità di miglioramento e il potenziale futuro del sistema.

Capitolo 2

Panoramica delle Tecnologie Impiegate

Nel seguente capitolo viene fornita un'esposizione dettagliata delle tecnologie e degli strumenti utilizzati durante lo sviluppo del progetto. In primo luogo, verranno analizzati i principi dell'approccio low-code, una metodologia che ha facilitato la realizzazione del chatbot attraverso l'uso di strumenti che riducono la necessità di scrivere codice complesso. Successivamente, verranno approfonditi i software specifici impiegati, tra cui Power Apps, Power Automate e Microsoft Copilot Studio, tutti componenti della Microsoft Power Platform, che hanno reso possibile la creazione di applicazioni e flussi di lavoro personalizzati. Oltre agli strumenti di sviluppo, il capitolo tratterà l'uso di SharePoint, una piattaforma centrale per la gestione e l'archiviazione dei documenti aziendali. La comprensione del funzionamento di SharePoint è fondamentale per comprendere come il chatbot accede e interroga i documenti riservati archiviati al suo interno. Infine, sarà presentata una panoramica dei servizi Azure, con un focus su Azure AI Search e OpenAI. Queste tecnologie di intelligenza artificiale sono state integrate nel sistema per migliorare le capacità di ricerca e risposta del chatbot, garantendo un'interazione più efficace e intelligente con gli utenti.

2.1 Low-code

Il low-code rappresenta un approccio allo sviluppo software che permette di creare applicazioni attraverso strumenti di modellazione visuale e configurazioni, riducendo significativamente la necessità di scrivere codice manuale. Le piattaforme low-code forniscono interfacce grafiche che semplificano il processo di sviluppo, rendendolo accessibile non solo ai programmatori esperti ma anche a coloro con competenze tecniche limitate. Questo approccio mira a democratizzare lo sviluppo software,

consentendo a un numero maggiore di persone di partecipare alla creazione di soluzioni digitali.

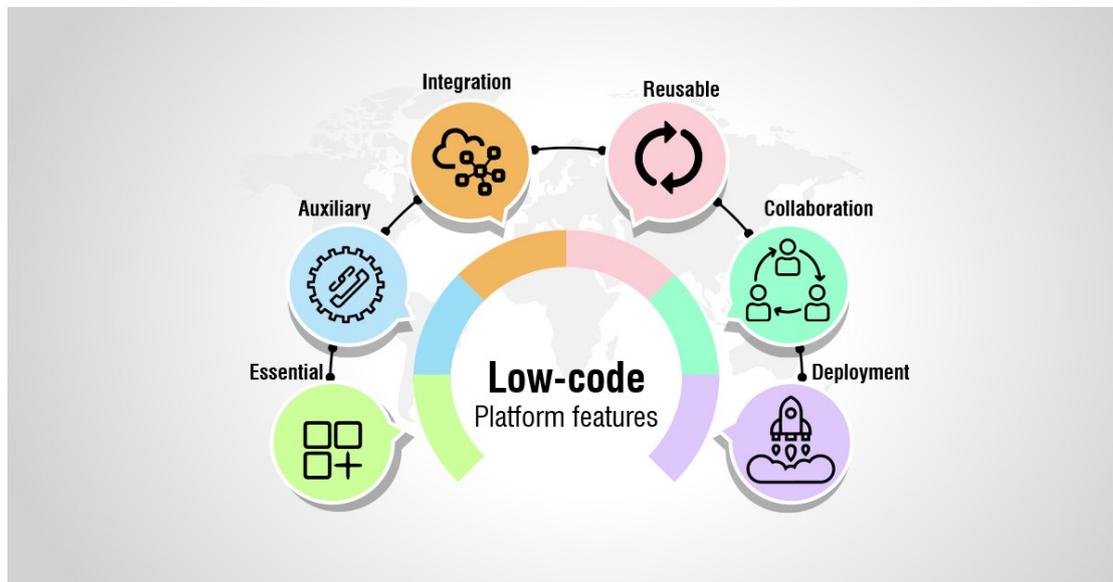


Figura 2.1: Le caratteristiche dello sviluppo Low-Code

L'emergere e la diffusione delle piattaforme low-code devono essere comprese nel più ampio contesto dello sviluppo software e delle dinamiche di mercato che ne hanno guidato la trasformazione. Tradizionalmente, lo sviluppo software ha richiesto competenze altamente specializzate, con processi che si articolavano in cicli lunghi e complessi. Questo modello, seppur efficace per la creazione di applicazioni su misura, ha dimostrato nel tempo significativi limiti in termini di velocità e adattabilità, specialmente in un contesto economico e tecnologico in rapida evoluzione. L'aumento esponenziale della domanda di soluzioni digitali è stato uno dei principali catalizzatori di questo cambiamento, infatti le imprese di ogni settore hanno iniziato a percepire la necessità di digitalizzare i propri processi per rimanere competitive, affrontando contemporaneamente un'accelerazione della domanda di nuove applicazioni. Tuttavia, questa esigenza si è scontrata con la disponibilità limitata di sviluppatori qualificati, generando un divario tra la domanda di soluzioni tecnologiche e la capacità di fornirle, ha messo in evidenza l'urgenza di strumenti che potessero semplificare e accelerare il processo di sviluppo, riducendo la dipendenza da competenze tecniche specialistiche [7].

Parallelamente, la crescente pressione per ridurre i tempi di sviluppo e contenere i costi associati ha reso evidenti le inefficienze dei modelli tradizionali, difatti le aziende si sono ritrovate a dover bilanciare la necessità di innovazione con quella di efficienza operativa, cercando soluzioni che permettesero di sviluppare applicazioni

in modo più rapido e, allo stesso tempo, economico in termini di risorse, senza però compromettere la qualità. In questo contesto, il low-code è emerso come risposta efficace e innovativa, consentendo lo sviluppo di applicazioni mediante strumenti di programmazione visuale e modelli predefiniti facilmente personalizzabili e riducendo significativamente la quantità di codice manuale necessaria. Se la possibilità di utilizzare componenti predefiniti e la facilità di personalizzazione hanno reso il low-code particolarmente appetibile per diverse aziende che necessitano di accelerare i cicli di sviluppo, senza compromettere funzionalità o la qualità delle soluzioni finali, la democratizzazione dello sviluppo software ne ha rappresentato un ulteriore motore di crescita. Il movimento dei *citizen developers*, ovvero utenti non tecnici che creano applicazioni per rispondere a specifiche esigenze aziendali, ha sfruttato queste piattaforme per realizzare soluzioni innovative senza dover necessariamente possedere competenze di programmazione avanzate. Questo ha effettivamente cambiato la cultura organizzativa, spostando lo sviluppo di applicazioni dalle sole mani dei team IT a una più ampia gamma di utenti all'interno del panorama aziendale, favorendo una maggiore partecipazione e collaborazione interfunzionale [8]. L'ascesa del low-code è stata inoltre favorita da fattori tecnologici e di mercato, come la diffusione del cloud computing, che ha reso più accessibili e scalabili le piattaforme low-code, consentendo alle aziende di implementare rapidamente nuove applicazioni senza doversi preoccupare dell'infrastruttura sottostante, ma allo stesso la maturazione delle tecnologie di modellazione visuale e l'integrazione con strumenti di intelligenza artificiale e machine learning hanno arricchito le capacità di queste piattaforme, rendendole sempre più potenti, versatili e pratiche da utilizzare. Infine, la pandemia COVID-19 ha anch'essa ulteriormente influenzato, o per meglio dire accelerato, l'adozione del low-code, poiché le aziende hanno dovuto affrontare la necessità di digitalizzare rapidamente i propri processi per rispondere alle nuove sfide imposte dal contesto globale, l'effettiva urgenza di sviluppare applicazioni in tempi brevi, spesso con risorse anche limitate, ha reso il low-code una scelta quasi obbligata per molte organizzazioni.

Tuttavia, sebbene il low-code offra indubbi vantaggi, è essenziale considerare anche i limiti e le sfide che questa tecnologia comporta. Un aspetto cruciale è rappresentato dalla scalabilità: mentre il low-code è ideale per sviluppare rapidamente applicazioni semplici, può diventare meno adeguato quando si tratta di progetti complessi che richiedono un alto grado di personalizzazione o integrazione con sistemi esistenti sofisticati, le piattaforme low-code, infatti, possono limitare la capacità di espandere e adattare le applicazioni nel tempo, portando a soluzioni subottimali che potrebbero non soddisfare completamente le esigenze aziendali in continua evoluzione. Un'altra sfida significativa riguarda la percezione e l'uso del low-code da parte di sviluppatori professionisti. Sebbene queste piattaforme democratizzino lo sviluppo software, rendendolo accessibile anche ai non esperti, come citato in precedenza, esse possono essere percepite come restrittive per coloro che

possiedono competenze tecniche avanzate. Gli sviluppatori possono sentirsi limitati nelle loro capacità creative, dato che il low-code impone un quadro predefinito di componenti e funzionalità, riducendo la flessibilità che si ottiene con la programmazione tradizionale. Questo fenomeno può essere paragonato al *Newspeak* descritta da George Orwell nel romanzo *1984*, dove la riduzione del linguaggio riduce anche la capacità di pensare ed esprimere idee complesse, analogamente, le piattaforme low-code possono restringere il potenziale creativo degli sviluppatori, vincolandoli a scelte già definite [9]. Infine, il rischio di “vendor lock-in” è un’altra preoccupazione rilevante. Le aziende che adottano una specifica piattaforma low-code possono diventare fortemente dipendenti da essa, limitando la loro capacità di migrare a soluzioni alternative o di adattarsi a nuove tecnologie in futuro. Questo rischio può diventare particolarmente problematico se il fornitore della piattaforma non riesce a mantenere il passo con le esigenze tecnologiche emergenti o con l’evoluzione del mercato, costringendo l’azienda a rimanere vincolata a una tecnologia che potrebbe non soddisfare più le proprie necessità.

In conclusione, il low-code ha rivoluzionato il panorama dello sviluppo software, abbassando le barriere all’ingresso e democratizzando la creazione di applicazioni. La sua capacità di accelerare i tempi di sviluppo, ridurre i costi e coinvolgere un pubblico più ampio di utenti ne ha fatto una tecnologia dirompente. Tuttavia, è fondamentale riconoscere i suoi limiti, come la scalabilità per progetti complessi e la potenziale rigidità per gli sviluppatori esperti. Nonostante queste sfide, il low-code rappresenta un pilastro fondamentale per la trasformazione digitale delle imprese.

2.2 Power Platform

La Power Platform di Microsoft è una suite di strumenti concepita per facilitare lo sviluppo di soluzioni aziendali attraverso un approccio low-code, rispondendo così alla crescente necessità di velocizzare la creazione di applicazioni senza richiedere avanzate competenze di programmazione. Lanciata ufficialmente nel 2018, la piattaforma si è posizionata come una risposta innovativa alla carenza di sviluppatori qualificati e alla crescente domanda di soluzioni software personalizzate all’interno delle aziende. Attraverso un’integrazione armoniosa di diversi componenti, la Power Platform permette agli utenti, che spaziano dai *citizen developers* ai professionisti IT, di creare applicazioni personalizzate, automatizzare flussi di lavoro e analizzare dati all’interno di un ecosistema unificato [10]. In questo contesto, la piattaforma si distingue per la capacità di democratizzare lo sviluppo software, facilitando un accesso più ampio al processo di creazione di applicazioni e favorendo l’innovazione distribuita all’interno delle organizzazioni.

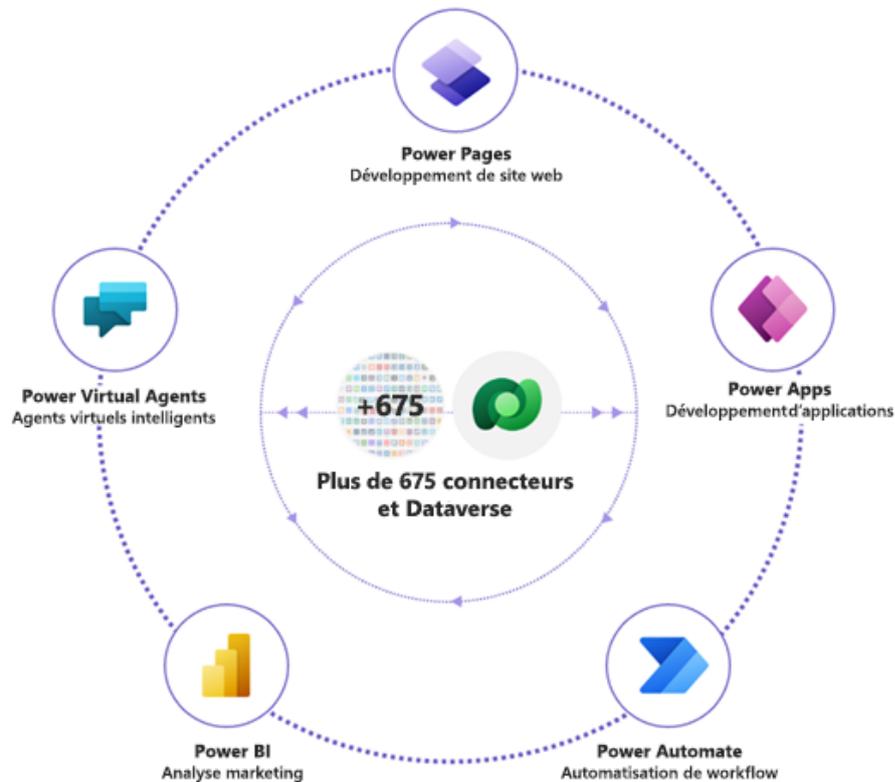


Figura 2.2: Struttura della Microsoft Power Platform

La Power Platform integra diverse tecnologie che permettono di gestire vari aspetti del lavoro azienda [10]:

- *Power Apps*: Strumento di sviluppo low-code che consente la creazione rapida di applicazioni aziendali personalizzate. Facilita l'integrazione con vari dati aziendali e offre un'interfaccia intuitiva per sviluppatori e non, riducendo i tempi di sviluppo e aumentando la produttività.
- *Power Automate*: Servizio di automazione che permette di creare flussi di lavoro automatizzati tra applicazioni e servizi, riducendo il tempo speso in attività ripetitive. È progettato per migliorare l'efficienza aziendale attraverso l'automazione di processi manuali.
- *Power Bi*: Strumento di business intelligence che consente di individuare e condividere dati approfonditi attraverso report interattivi e dashboard. Aiuta le aziende a prendere decisioni basate sui dati, trasformando grandi volumi di informazioni in insight utilizzabili.

- *Power Pages*: Piattaforma low-code per la creazione di siti web aziendali sicuri e professionali, collegati ai dati aziendali. Permette agli utenti di progettare e lanciare rapidamente portali personalizzati per clienti, partner o dipendenti.
- *Microsoft Copilot Studio*: Strumento avanzato che integra intelligenza artificiale nelle applicazioni aziendali, supportando gli utenti con suggerimenti automatici, generazione di contenuti e automazioni di task complessi, migliorando la produttività e l'efficacia delle soluzioni sviluppate.

Questi strumenti sono stati progettati per essere intuitivi e accessibili anche a chi non ha una formazione tecnica, pur mantenendo un elevato grado di personalizzazione e scalabilità grazie all'integrazione con il cloud di Microsoft Azure. Questa sinergia tra componenti diverse non solo rende possibile una rapida implementazione di nuove soluzioni, ma assicura anche che tali soluzioni possano evolvere e adattarsi alle crescenti esigenze aziendali.

Fin dalla sua introduzione, la Power Platform ha continuato ad arricchirsi con nuove funzionalità, tra cui l'integrazione di tecnologie di intelligenza artificiale e machine learning, che ne hanno esteso le capacità senza compromettere la facilità d'uso. In questo modo, la piattaforma rappresenta una risorsa fondamentale per le aziende che mirano a mantenere un vantaggio competitivo in un ambiente di mercato sempre più dinamico e basato sui dati.

2.2.1 Dataverse

Il Dataverse rappresenta una componente cruciale nell'ecosistema Microsoft e costituisce la base dati della Power Platform, fungendo da fondamento per la gestione e l'organizzazione dei dati aziendali all'interno di applicazioni low-code. Sebbene la sua struttura non esponga esplicitamente la complessità interna, "sotto" si appoggia effettivamente a un database relazionale, quasi certamente un'istanza di SQL Server, che garantisce robustezza e prestazioni elevate per la memorizzazione dei dati. Questo aspetto, combinato con l'approccio low-code, consente agli utenti di sfruttare i vantaggi di una configurazione semplificata e di una gestione integrata della sicurezza senza dover intervenire direttamente sulle complessità tipiche di un'infrastruttura relazionale [11].

Il Dataverse è stato originariamente introdotto come Common Data Service e successivamente evolutosi nell'attuale piattaforma, offrendo un ambiente sicuro e scalabile in cui i dati strutturati sono organizzati sotto forma di entità, analoghe a tabelle di un database relazionale. Questo livello di astrazione facilita l'interoperabilità tra le applicazioni della Power Platform, pur garantendo che alla base rimangano le solide fondamenta di un sistema relazionale [11].

Oltre a fornire una piattaforma di dati comune, il Dataverse include anche strumenti avanzati per la gestione della business logic, come i plugin, che possono

essere paragonati ai trigger nei database relazionali. Questi plugin consentono di eseguire logiche personalizzate o automatizzare processi in risposta a eventi sui dati, aggiungendo ulteriore flessibilità e potenzialità alle applicazioni costruite sulla piattaforma [11]. Inoltre, la sua capacità di standardizzare i dati attraverso modelli predefiniti, i Common Data Models, assicura coerenza e riduzione di duplicazioni, mentre la gestione granulare dei permessi d'accesso fornisce un elevato livello di sicurezza [12].

In questo contesto, Dynamics 365 rappresenta una suite di applicazioni aziendali che si fonda sulla Power Platform, beneficiando dell'integrazione con il Dataverse. Descrivere Dynamics 365 come semplicemente basato sul Dataverse sarebbe tuttavia riduttivo, poiché l'architettura complessiva sfrutta un'integrazione più profonda all'interno dell'ecosistema Microsoft. Questa integrazione consente una connessione fluida tra le diverse applicazioni aziendali e la gestione centralizzata dei dati, garantendo una sinergia ottimale tra gli strumenti della piattaforma [11].

In conclusione, il Dataverse non è soltanto un semplice sistema per la memorizzazione dei dati, ma un ambiente integrato che, grazie alla configurabilità low-code e alla gestione della sicurezza, supporta in modo efficiente e scalabile le operazioni aziendali, offrendo un'infrastruttura dinamica per la costruzione di applicazioni e flussi di lavoro [11].

2.3 Power Apps

Microsoft Power Apps è una piattaforma low-code all'interno della Power Platform, che consente agli utenti di sviluppare rapidamente applicazioni aziendali personalizzate [13]. Questa piattaforma si rivolge a un pubblico eterogeneo, dai *citizen developers* senza competenze tecniche avanzate, ai professionisti IT che necessitano di accelerare i cicli di sviluppo delle applicazioni. Power Apps è nato dall'esigenza crescente di fornire soluzioni digitali rapide ed efficienti che possano adattarsi rapidamente ai mutamenti del contesto aziendale o dei requisiti funzionali del progetto, riducendo, allo stesso tempo, la dipendenza dai team di sviluppo tradizionali.

Power Apps consente di costruire applicazioni partendo da modelli predefiniti o da una tela bianca, utilizzando strumenti visivi che permettono di creare interfacce utente e logiche di business senza scrivere codice. La sua integrazione inoltre con altri servizi Microsoft, come Microsoft 365, Dynamics 365 e Azure, ne amplifica l'efficacia, offrendo una soluzione scalabile e sicura che può essere adattata alle mutevoli esigenze aziendali. La piattaforma, inoltre, promuove la collaborazione tra team tecnici e non, democratizzando il processo di sviluppo software e migliorando la qualità delle soluzioni attraverso un approccio partecipativo.

2.3.1 Tipologie di Applicazioni

Power Apps offre due principali modalità di sviluppo di applicazioni: le Canvas App e le Model-Driven App.

- *Canvas App*: Queste applicazioni offrono un controllo totale sull'interfaccia utente, il processo di sviluppo di un'applicazione è come se partisse da una tela bianca. L'utente può trascinare e rilasciare componenti grafici per creare layout personalizzati, con la possibilità di connettere l'applicazione a numerose fonti dati. Questo approccio è particolarmente consigliato quando l'estetica e la user experience sono prioritarie [14].
- *Model-Driven App*: Esse, invece, adottano un approccio più strutturato, basato su modelli di dati predefiniti e fortemente integrato al Dataverse. L'interfaccia è in gran parte determinata dai dati e dalle relazioni tra di essi, garantendo coerenza e facilità d'uso per processi aziendali complessi. Questo tipo di applicazione è particolarmente adatto per la gestione di flussi di lavoro e la gestione di grandi quantità di dati strutturati [15].

2.3.2 Interfaccia Grafica e Componenti delle Model-Driven App

Le Model-Driven App di Microsoft Power Platform si caratterizzano, come già anticipato, per un'interfaccia grafica strutturata e orientata ai dati, pensata per offrire un'esperienza d'uso intuitiva e altamente personalizzabile. Questo tipo di applicazioni è organizzato attorno a due componenti chiave: View e Form, che sono fondamentali per la gestione e visualizzazione dei dati all'interno delle applicazioni aziendali [15].

- *View*: Le view sono essenziali per la presentazioni dei dati in forma tabellare o di lista. Permettono di visualizzare insieme di record in modo ordinato, consentendo di filtrare e ordinare le informazioni secondo criteri specifici, in base alle necessità dell'utente. Questo strumento è particolarmente utile per esplorare i dati in modo efficiente, rendendo possibile individuare rapidamente le informazioni rilevanti. Le view possono essere configurate per mostrare esattamente i campi di interesse, e possono essere ulteriormente personalizzate per riflettere le esigenze specifiche del business [16].
- *Form*: I form, d'altra parte, rappresentano l'elemento attraverso il quale si interagisce con i singoli record. Essi forniscono interfacce dettagliate per visualizzare e modificare i dati relativi a un record specifico. La potenza dei form risiede nella loro elevata personalizzabilità: Power Apps, infatti, consente agli utenti di configurare i form per includere layout complessi, campi

obbligatorie, regole di validazione e logiche di business. Sebbene i form siano flessibili dal punto di vista funzionale, presentano limitazioni per quanto riguarda la personalizzazione grafica e la user experience, risultando meno versatili rispetto alle canvas app. Tuttavia, grazie all'interfaccia visiva, è possibile apportare modifiche senza scrivere codice, permettendo agli utenti di creare esperienze ottimizzate e coerenti con le esigenze aziendali, assicurando al contempo una gestione dei dati efficace e sicura [17].

2.3.3 Vantaggi, Casi d'uso e Limiti

Microsoft Power Apps offre una serie di vantaggi distinti che rendono la piattaforma particolarmente attraente per le aziende che cercano soluzioni rapide e flessibili. Uno dei principali punti di forza è la rapidità di sviluppo: grazie alla natura low-code della piattaforma, i tempi di sviluppo sono notevolmente ridotti rispetto ai metodi tradizionali, portando a una significativa riduzione dei costi. Questo si traduce in un ciclo di vita del prodotto più veloce, consentendo alle aziende di rispondere tempestivamente alle esigenze del mercato e ai cambiamenti nelle condizioni operative. La flessibilità è un altro vantaggio cruciale, la piattaforma è altamente adattabile a diverse esigenze aziendali, permettendo lo sviluppo di applicazioni personalizzate che possono essere rapidamente modificate o ampliate per rispondere a nuove esigenze, ciò risulta particolarmente utile in settori dinamici dove le esigenze aziendali possono evolvere molto rapidamente. Un ulteriore vantaggio di Power Apps è la sua integrazione senza soluzione di continuità con il mondo Microsoft, come già anticipato precedentemente. Questa integrazione consente alle applicazioni sviluppate di sfruttare appieno le funzionalità e i dati provenienti da altre soluzioni Microsoft, creando un ecosistema coeso e potente. I casi d'uso specifici di Power Apps sono numerosi e variegati, dimostrando la versatilità della piattaforma. Nel settore CRM, Power Apps consente la creazione di soluzioni personalizzate che migliorano l'interazione con i clienti e ottimizzano i processi di vendita; in ambito HR, invece, la piattaforma può essere utilizzata per sviluppare applicazioni che gestiscono il ciclo di vita dei dipendenti, dall'assunzione alla gestione delle performance [13].

Nonostante i numerosi vantaggi, Power Apps presenta alcune limitazioni che vale la pena considerare. Da non trascurare è il fatto che la piattaforma può presentare una certa curva di apprendimento per gli utenti non tecnici, sebbene essa sia progettata per essere accessibile anche ai non sviluppatori, la necessità di comprendere concetti fondamentali, quali View e Form, e la logica di base della piattaforma stessa può richiedere una formazione iniziale per essere utilizzata in maniera efficiente. Un'altra problematica radicata è la forte dipendenza dal Dataverse, che può avere implicazioni per la scalabilità. Sebbene il Dataverse sia un potente strumento per la gestione dei dati, applicazioni molto grandi o complesse

potrebbero richiedere una gestione più sofisticata e potrebbero incontrare limitazioni in termini di prestazioni. Infine, la gestione e la manutenzione di applicazioni di grandi dimensioni e complessità possono presentare delle sfide significative. In questi casi, è necessaria una pianificazione più attenta e un approccio strutturato sin dalle fasi iniziali di progettazione; l'organizzazione delle risorse, la gestione delle relazioni tra i dati e l'implementazione di logiche di business articolate richiedono una cura particolare per garantire che l'applicazione rimanga scalabile, coerente ed efficiente nel tempo.

A seguito di questa analisi, la possibilità di personalizzare in modo così dettagliato sia le View che i Form, insieme alla capacità di integrare queste applicazioni con altre soluzioni e servizi all'interno dell'ecosistema Microsoft, come ad esempio Dataverse per la gestione dei dati, rende la Model-Driven App uno strumento estremamente potente e versatile [15]. Questa combinazione di efficienza e versatilità risponde perfettamente alle esigenze delle aziende moderne, che necessitano di soluzioni su misura per rimanere competitive in un mercato in continua evoluzione.

2.4 Power Automate

Power Automate è uno strumento di automazione dei flussi di lavoro sviluppato da Microsoft, concepito per semplificare e ottimizzare i processi aziendali e avente come prerogativa il consentire anche agli utenti con limitate competenze tecniche di creare automazioni complesse [18]. Con Power Automate, le aziende possono collegare applicazioni e servizi diversi per automatizzare operazioni quotidiane, riducendo così il lavoro manuale e incrementando, di conseguenza, l'efficienza operativa.

L'introduzione dell'automazione con Power Automate porta vantaggi significativi alle organizzazioni. In primo luogo, la riduzione del lavoro manuale non solo aumenta l'efficienza, ma allo stesso tempo, diminuisce la probabilità di errori umani, migliorando così l'accuratezza e l'affidabilità dei processi aziendali. Inoltre, liberando tempo prezioso per i dipendenti, l'automazione consente loro di concentrarsi su attività più strategiche e di alto valore, contribuendo quindi al miglioramento della produttività complessiva [18]. Power Automate si distingue nel panorama delle soluzioni di automazione per diversi motivi, primi fra tutti la sua facilità d'uso che lo rende accessibile ad un'ampia gamma di utenti e per l'integrazione con altri prodotti Microsoft, che lo rende particolarmente appetibile per le aziende che operano già nel suddetto ecosistema. Questa capacità di interagire senza soluzione di continuità con altre applicazioni rappresenta un valore aggiunto, favorendo un'implementazione rapida e una gestione integrata dei flussi di lavoro, inoltre, la scalabilità di Power Automate consente di adattarlo facilmente alle diverse esigenze aziendali, rendendolo uno strumento flessibile sia per piccole automazioni che per processi più complessi.

2.4.1 Il Concetto di Flow

Il cuore di Power Automate è rappresentato dal concetto di *Flow*, ovvero una sequenza predefinita di operazioni automatizzate che si attivano al verificarsi di specifici eventi, chiamati trigger. Un Flow, quindi, è un flusso di lavoro automatizzato, che elimina la necessità di intervento umano, eseguendo automaticamente le operazioni necessarie al verificarsi di determinate condizioni [19].

Ogni Flow è composto da elementi fondamentali che ne definiscono il funzionamento, il trigger, come già citato in precedenza, è quello iniziale, ovvero l'evento che avvia il Flow. Questo può essere, ad esempio, l'arrivo di un'email, l'aggiornamento di un file in un repository o l'inserimento di nuovi dati in un sistema. Una volta attivato il trigger, il Flow prosegue con una serie di azioni predefinite, che possono includere operazioni come l'invio di notifiche, l'aggiornamento di record in un database o persino la generazione di report. La presenza di condizioni all'interno di un Flow permette di gestire percorsi decisionali, rendendo, difatti, l'automazione più sofisticata e adattabile a situazioni specifiche [19].

Power Automate offre diverse tipologie di Flow, ciascuna progettata per rispondere a esigenze specifiche.

- *Flow Automatici*: si avviano automaticamente in risposta a un evento, garantendo un'automazione continua e senza intervento manuale.
- *Flow Istantanei*: vengono avviati manualmente dall'utente, offrendo un controllo diretto sull'esecuzione dell'automazione.
- *Flow Pianificati*: si attivano a intervalli regolari o in base a una programmazione temporale, sono ideali per attività ricorrenti che richiedono una regolarità d'esecuzione.

2.4.2 Creazione di un Flow

L'interfaccia di Power Automate è concepita per essere intuitiva, permettendo agli utenti di creare Flow attraverso un processo visivo e guidato. all'interno dell'interfaccia, è possibile visualizzare tutti i Flow esistenti, monitorarne lo stato e crearne di nuovi utilizzando un editor che semplifica la configurazione delle automazioni attraverso un sistema di trascinamento e rilascio.

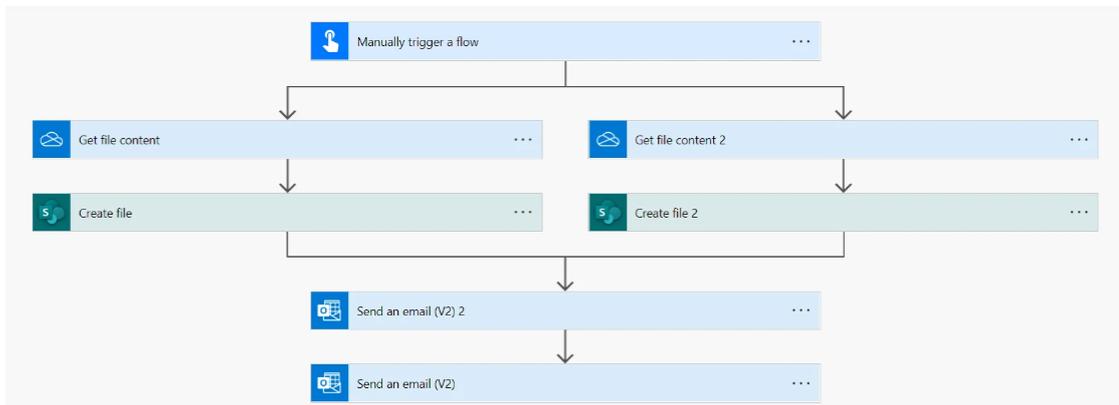


Figura 2.3: Flusso in Power Automate

La creazione di un Flow inizia con la scelta del trigger, che determina l'evento che attiverà l'automazione, si tratta di un passaggio cruciale, in quanto definisce il punto di partenza del flusso di lavoro. Successivamente, l'utente può aggiungere una o più azioni che il Flow dovrà eseguire, come l'invio di email, l'aggiornamento di database o la creazione di un file. L'editor di Power Automate permette di costruire queste sequenze in modo logico e intuitivo, verificando che tutte le condizioni e le azioni siano correttamente configurate prima dell'attivazione del Flow. Un aspetto distintivo di Power Automate è l'ampia disponibilità di connettori, che permettono di integrare il Flow con un vasto numero di servizi e applicazioni, sia all'interno dell'ecosistema Microsoft che di terze parti. Questi connettori sono fondamentali per estendere le capacità di automazione, consentendo la comunicazione e l'interazione tra diversi sistemi, grazie ad essi, gli utenti sono in grado di creare flussi di lavoro che integrano servizi come SharePoint o OneDrive, rendendo possibile l'automazione di processi che coinvolgono più piattaforme [18].

2.4.3 Esecuzione e Monitoraggio

Una volta creato, un Flow può essere avviato manualmente o automaticamente, in base al tipo di trigger configurato. I flow automatici si attivano in maniera autonoma, seguendo l'evento definito dal trigger, mentre quelli istantanei richiedono un'azione diretta dell'utente per essere eseguiti. Questa flessibilità nell'esecuzione permette di adattare i Flow a diversi scenari operativi. Inoltre, Power Automate supporta trigger basati su eventi specifici, che si attivano in risposta a cambiamenti in un'applicazione o servizio. Questi trigger consentono ai Flow di reagire immediatamente a situazioni dinamiche, rendendo le automazioni più reattive e adattabili a un'ampia gamma di scenari aziendali.

Power Automate offre anche strumenti avanzati per il monitoraggio e la gestione dei Flow. Attraverso la dashboard di monitoraggio, gli utenti possono visualizzare

lo stato di ogni flusso di lavoro, accedere ai log delle esecuzioni e identificare eventuali errori o problemi. Questo livello di controllo è essenziale per garantire che i Flow operino in maniera efficiente e affidabile, consentendo di intervenire tempestivamente in caso di malfunzionamenti; tale capacità di monitorarne in tempo reale l'esecuzione, analizzando i log e diagnosticando eventuali problemi, contribuisce a migliorare l'affidabilità e la sicurezza delle automazioni dei processi aziendali [18].

2.4.4 Casi d'Uso e Scenari

Power Automate si presta a una vasta gamma di applicazioni in contesti aziendali diversi, dimostrando una notevole versatilità. Un esempio pratico è l'automazione dei processi di approvazione, dove Power Automate può gestire l'intero ciclo di richiesta, invio e ricezione delle approvazioni, riducendo, quindi, i tempi di attesa e garantendo che le approvazioni siano indirizzate ai destinatari corretti in modo tempestivo.

La sincronizzazione dei dati è un altro ambito in cui Power Automate può fare la differenza. Ad esempio, può essere configurato per allineare automaticamente i dati tra un sistema CRM e una piattaforma di gestione delle risorse umane, eliminando, difatti, la necessità di inserimenti manuali e riducendo il rischio di errori. Inoltre, l'invio di notifiche automatiche è un'altra funzione comune, utile per informare i team di eventi importanti, come ad esempio, il superamento di una soglia di vendita, garantendo che tutte le parti interessate siano aggiornate in tempo reale.

2.4.5 Vantaggi e Svantaggi

Power Automate offre una serie di vantaggi che lo rendono uno strumento prezioso per l'automazione dei processi aziendali. La sua facilità d'uso è uno dei punti di forza principali, indubbiamente, consentendo agli utenti di creare automazioni senza bisogno di competenze tecniche avanzate. Inoltre, l'integrazione con l'ecosistema Microsoft lo rende particolarmente efficace per le organizzazioni che già utilizzano questi strumenti, favorendo una gestione coesa e integrata dei processi [18]. La flessibilità e scalabilità di Power Automate permettono di adattare facilmente lo strumento a diverse esigenze aziendali, rendendolo una soluzione adatta sia a piccole automazioni che a progetti più complessi.

Tuttavia, Power Automate presenta alcune limitazioni. Sebbene sia concepito come uno strumento low-code, la creazione di Flow complessi che coinvolgono integrazioni con sistemi di terze parti può richiedere competenze tecniche avanzate, rappresentando una sfida per alcuni utenti, inoltre alcune azioni che operano sui dati in modo avanzato, come l'uso delle espressioni regolari, possono risultare particolarmente complesse da implementare e gestire all'interno di Power Automate,

soprattutto rispetto alla programmazione tradizionale. Infine, man mano che la complessità dell'automazione aumenta, i Flow tendono a diventare più lunghi, ciò può comportare non solo un aumento del tempo di esecuzione e dei problemi di latenza, ma anche una maggiore difficoltà nella visualizzazione e gestione del flusso di lavoro, che potrebbe risultare ingombrante e difficile da seguire a causa dei numerosi blocchi decisionali o azioni coinvolte. Pertanto, è essenziale mantenere una documentazione accurata e ben organizzata dei processi per facilitare la gestione e prevenire inefficienze [19].

Power Automate è in continua evoluzione, con Microsoft che investe costantemente nello sviluppo di nuove funzionalità per rispondere alle esigenze sempre crescenti nelle aziende. Tra le tendenze future, si prevede un'integrazione sempre più profonda con l'intelligenza artificiale e l'apprendimento automatico, che permetterà di automatizzare decisioni più complesse e migliorare ulteriormente l'efficienza operativa. Inoltre, con la crescente diffusione del lavoro remoto, Power Automate è destinato a diventare un componente chiave nella gestione e automazione dei processi in ambienti di lavoro distribuiti.

2.5 Microsoft Copilot Studio

Microsoft Copilot Studio rappresenta una delle più recenti innovazioni nell'ambito delle piattaforme low-code, concepita per facilitare la creazione di chatbot e assistenti virtuali. La sua definizione si radica nella capacità di offrire strumenti potenti, ma accessibili, progettati per consentire anche a utenti non tecnici di sviluppare soluzioni conversazionali personalizzate, accelerando così i processi di automazione delle interazioni aziendali. Nel contesto odierno, in cui le aziende si affidano sempre più a tecnologie digitali per migliorare l'efficienza operativa e l'esperienza del cliente, Copilot Studio emerge come una risposta alle esigenze di flessibilità e personalizzazione [20].

L'evoluzione di Microsoft Copilot Studio ha le sue radici in Power Virtual Agents, una piattaforma che ha segnato un'importante svolta nell'accessibilità degli strumenti di sviluppo per chatbot. Inizialmente concepito come uno strumento per semplificare la creazione di agenti virtuali, Power Virtual Agents ha progressivamente integrato funzionalità sempre più sofisticate di intelligenza artificiale, e questo processo evolutivo ha culminato nella creazione di Copilot Studio, che non solo eredita le capacità del suo predecessore, ma le amplia, adottando un approccio più centrato sull'IA e fortemente integrato con la suite Microsoft 365. La trasformazione da Power Virtual Agents a Copilot Studio evidenzia il passaggio da una semplice piattaforma di sviluppo a uno strumento più avanzato e versatile, capace di sfruttare al meglio le potenzialità dell'intelligenza artificiale per rispondere alle sfide della comunicazione moderna [20].

Nel panorama competitivo delle soluzioni per chatbot e assistenti virtuali, Microsoft Copilot Studio si erge come una piattaforma leader, che si distingue da altre soluzioni concorrenti, come Google Dialogflow o IBM Watson Assistant, grazie alla facilità d'uso, che la rende accessibile anche a chi non ha conoscenze tecniche approfondite, e alla sua forte integrazione con l'ecosistema Microsoft, offrendo così una soluzione completa che va oltre la semplice automazione delle conversazioni, per abbracciare l'intero ecosistema aziendale [20]. Questo posizionamento strategico lo rende non solo un concorrente valido, ma spesso una scelta preferenziale per le aziende che già utilizzano prodotti Microsoft e cercano una soluzione che si integri perfettamente nei loro processi esistenti.

2.5.1 Il Ruolo dell'Intelligenza Artificiale

L'intelligenza artificiale gioca un ruolo cruciale nel funzionamento di Microsoft Copilot Studio, costituendo il motore che alimenta la sua capacità di gestire interazioni complesse e personalizzate. Al centro di questa capacità c'è l'utilizzo dei modelli di linguaggio naturale, che permettono alla piattaforma di comprendere e interpretare il linguaggio umano con un grado di precisione e coerenza notevoli [21]. L'IA, attraverso l'NLP, consente a Copilot Studio di gestire conversazioni con un livello di comprensione contestuale che simula in modo efficace le interazioni umane. Questo non solo rende il chatbot capace di rispondere alle domande degli utenti in maniera coerente, ma gli permette anche di gestire una vasta gamma di scenari, adattandosi dinamicamente alle richieste specifiche e offrendo risposte che non solo risolvono il problema immediato, ma che anche arricchiscono l'esperienza complessiva dell'utente.

Un altro aspetto fondamentale dell'IA in Copilot Studio è la capacità di personalizzare delle esperienze utente. Grazie agli avanzamenti nel machine learning, la piattaforma è in grado di adattarsi alle preferenze individuali degli utenti e al contesto specifico in cui avviene l'interazione [21]. Questo significa che ogni conversazione può essere modulata in base a parametri che tengono conto del comportamento passato degli utenti, delle loro preferenze esplicite o implicite, nonché dal contesto in cui la conversazione si svolge. Tale livello di personalizzazione non solo migliora l'efficacia delle interazioni, ma contribuisce anche a creare un'esperienza utente che è percepita come più naturale e intuitiva, rafforzando, in un certo senso, il legame tra l'utente e la piattaforma.

Infine, uno degli elementi più distintivi e innovativi di Microsoft Copilot Studio è la sua capacità di apprendimento continuo; l'IA alla base della piattaforma non si limita a eseguire compiti predefiniti, ma evolve nel tempo grazie all'apprendimento automatico. Ogni interazione con gli utenti diventa un'opportunità per affinare e migliorare le capacità del chatbot, rendendolo progressivamente più efficace e reattivo. Questo processo di *continuous learning* è fondamentale per garantire che la

piattaforma possa adattarsi, in maniera dinamica, a nuove esigenze, cambiamenti nel linguaggio o nelle aspettative degli utenti, mantenendo sempre un alto livello di performance.

2.5.2 Topic

Nel contesto di Microsoft Copilot Studio, i Topics costituiscono l'ossatura delle conversazioni, definendo la struttura attraverso cui l'interazione tra utente e assistente virtuale prende forma. Un Topic, in termini semplici, è un flusso di conversazione predefinito, progettato per coprire uno specifico argomento o compito. Questo concetto è fondamentale per comprendere come Copilot Studio organizzi e gestisca le interazioni, offrendo una base solida su cui costruire conversazioni complesse e ben orchestrate [22].

La struttura di un Topic è composta da diversi elementi chiave, ciascuno con una funzione specifica:

- *Trigger*: Sono le condizioni che attivano un determinato Topic, basandosi su input specifici da parte dell'utente, come una domanda o una richiesta.
- *Nodi*: Rappresentano i punti di snodo all'interno della conversazione, determinando come o quando un certo flusso di dialogo progredisce.
- *Risposte*: Sono i messaggi effettivi che l'assistente virtuale comunica all'utente, progettati per fornire informazioni, risolvere problemi o guidare l'utente attraverso una serie di azioni.

Questa struttura modulare non solo facilita la gestione delle conversazioni, ma consente anche un alto grado di personalizzazione, permettendo agli sviluppatori di adattare i flussi di conversazione alle esigenze specifiche del contesto operativo.

La flessibilità dei Topics è uno degli aspetti che rende Microsoft Copilot Studio una piattaforma potente e versatile. I Topics, non sono entità rigide, ma possono essere combinati, modificati e personalizzati per creare esperienze conversazionali che vanno ben oltre i semplici flussi lineari. Ad esempio, è possibile collegare diversi Topics tra loro per gestire conversazioni più complesse, che possono variare dinamicamente in base alle risposte dell'utente. Queste funzionalità permettono di costruire soluzioni che rispondono in modo preciso alle esigenze dell'utente, creando interazioni che sono al tempo stesso fluide e adattabili.

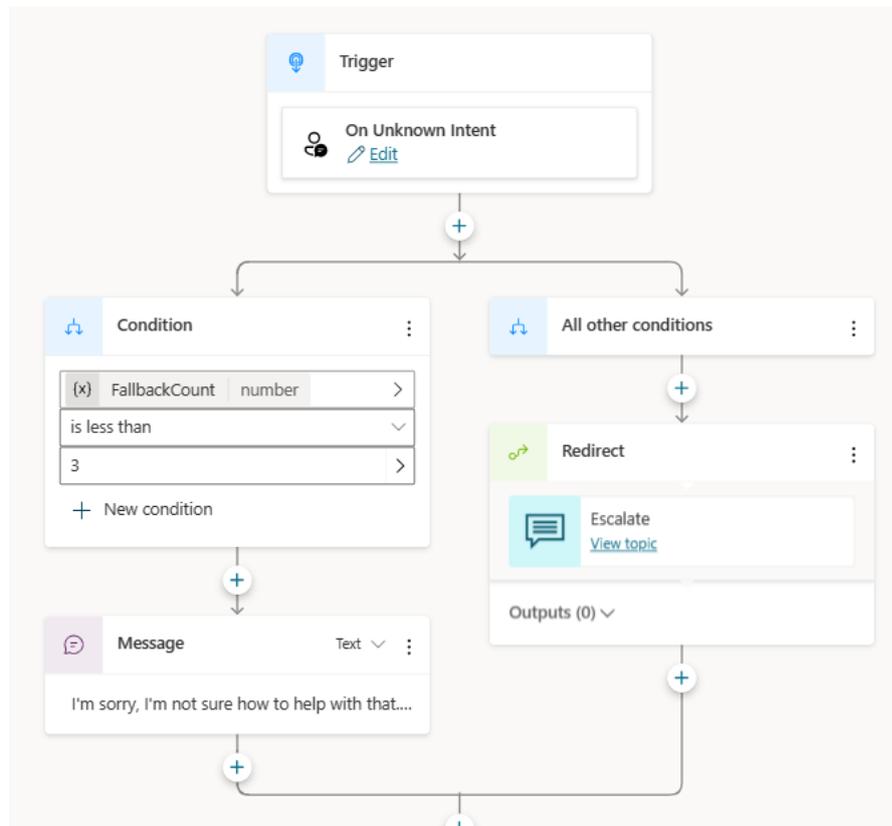


Figura 2.4: Topic di Microsoft Copilot Studio

2.5.3 Plugin Actions

Le Plugin Actions in Microsoft Copilot Studio rappresentano uno strumento cruciale per l'estensione delle funzionalità del chatbot, offrendo un grado elevato di flessibilità e personalizzazione. Definite come blocchi costruttivi, queste azioni consentono di arricchire le capacità di un copilot, permettendo l'esecuzione di operazioni complesse che vanno ben oltre le semplici risposte conversazionali [23]. Attraverso l'integrazione di plugin, i copilot possono interagire con sistemi esterni, automatizzare processi aziendali e accedere a dati specifici, rendendo l'assistente virtuale non solo un interlocutore, ma un e vero e proprio agente operativo.

I tipi di Plugin Actions disponibili in Copilot Studio sono diversi, e ognuno di essi risponde a esigenze specifiche [23]. Alcune delle azioni più comuni includono:

- *Chiamate API*: Consentono al copilot di inviare richieste a servizi esterni e di ricevere dati in tempo reale, facilitando l'integrazione con database, sistemi CRM o altri strumenti aziendali.

- *Automazioni personalizzate*: Azioni che permettono di eseguire script o sequenze di operazioni automatizzate, come l'invio di email, la generazione di report o l'aggiornamento di record in un database. Queste automazioni sono particolarmente utili per ottimizzare processi interni e ridurre l'intervento manuale.
- *Interazioni con l'utente*: Si tratta di azioni che coinvolgono l'utente direttamente, come presentazione di moduli interattivi o la richiesta di input specifici. Queste azioni migliorano l'engagement e garantiscono che il copilot possa raccogliere informazioni utili per fornire risposte più accurate e contestualizzate.

L'integrazione di queste azioni all'interno dell'editor di Copilot Studio è progettata per essere intuitiva e flessibile. Gli sviluppatori possono facilmente aggiungere, configurare e gestire le Plugin Actions direttamente dall'editor, utilizzando un'interfaccia drag-and-drop che rende la personalizzazione del copilot accessibile anche a chi non ha competenze avanzate di programmazione.

Gli scenari d'uso sono molteplici e coprono un'ampia gamma di applicazioni pratiche. Ad esempio, un'azienda potrebbe integrare un copilot con il proprio sistema CRM per automatizzare i record dei clienti in base alle interazioni; oppure grazie alla capacità di eseguire chiamate API, è possibile creare chatbot personalizzati che accedono ai dati in tempo reale, come saldi dei conti o stato di un ordine, fornendo risposte precise e tempestive agli utenti. La loro semplicità d'uso, combinata con la potenza e le potenzialità delle Plugin Actions, permette di sviluppare rapidamente soluzioni sofisticate che rispondono in modo preciso alle necessità aziendali.

2.5.4 Canali di Comunicazione e Analisi delle Prestazioni

La capacità di Microsoft Copilot Studio di raggiungere gli utenti attraverso una vasta gamma di canali di comunicazione è un elemento chiave che ne aumenta l'efficacia e la versatilità. La piattaforma supporta diversi canali, tra cui Microsoft Teams, applicazioni web, dispositivi mobili e altre interfacce di comunicazione aziendale [20]. Questa ampia compatibilità garantisce che le soluzioni sviluppate possano essere utilizzate ovunque, facilitando l'accesso alle informazioni e migliorando l'interazione con gli utenti indipendentemente dal dispositivo o dalla piattaforma utilizzata.

Oltre alla gestione dei canali di comunicazione, Copilot Studio offre una suite completa di strumenti di analisi per monitorare e valutare le prestazioni del copilot [24]. Questi strumenti sono fondamentali per comprendere l'efficacia delle interazioni e per identificare aree di miglioramento. Tra le metriche chiave che possono essere tracciate, troviamo:

- *Tasso di successo delle conversazioni*: Questa metrica misura la percentuale di conversazioni concluse con successo, ovvero quando il copilot riesce a

soddisfare la richiesta dell'utente senza necessità di intervento umano. Un tasso di completamento elevato indica un'alta efficienza del copilot nel gestire le interazioni.

- *Tasso di escalation*: Indica la percentuale di conversazioni che richiedono l'intervento umano, segnalando le situazioni in cui il copilot non è in grado di gestire richieste più complesse. Un tasso di escalation basso suggerisce un'alta autonomia operativa del copilot.
- *Tasso di abbandono*: Questo valore rappresenta la percentuale di utenti che interrompono la conversazione prima che questa sia completata. Un tasso di abbandono basso è indicativo di un copilot capace di mantenere l'attenzione e l'interesse degli utenti durante l'interazione.
- *Soddisfazione del cliente*: Questa metrica riflette la soddisfazione degli utenti in base alle loro esperienze con il copilot. Un alto punteggio di soddisfazione dei clienti suggerisce che il copilot non solo risolve efficacemente le richieste, ma lo fa in modo che gli utenti considerano positivo e appagante.

L'utilizzo di queste metriche analitiche è fondamentale per ottimizzare le prestazioni del copilot. Monitorando il tasso di completamento, le organizzazioni possono valutare, non solo il corretto funzionamento, ma anche l'efficacia complessiva delle risposte dei loro copilot; il tasso di escalation aiuta a individuare i punti in cui potrebbe essere necessario supporto aggiuntivo, mentre un tasso di abbandono basso indica un alto livello di engagement degli utenti con il copilot, suggerendo che le interazioni sono coinvolgenti e soddisfacenti. Infine, il punteggio di soddisfazione del cliente è cruciale per comprendere la qualità percepita dell'interazione, permettendo di apportare miglioramenti mirati per aumentare la soddisfazione degli utenti.

Queste metriche non solo permettono di migliorare continuamente il copilot, ma offrono anche un quadro chiaro dell'interazione complessiva con gli utenti, consentendo di personalizzare ulteriormente Topics o Plugin Actions di modo che il copilot possa soddisfare le esigenze specifiche del pubblico, mantenendo elevati standard di prestazione e soddisfazione dell'utente.

2.6 Servizi Azure

Azure rappresenta la piattaforma cloud di Microsoft, progettata per rispondere alle esigenze di sviluppo, distribuzione e gestione delle applicazioni in un contesto sempre più digitalizzato e competitivo. Fin dalla sua introduzione, Azure ha trasformato il modo in cui le aziende affrontano l'IT, offrendo loro la possibilità di utilizzare infrastrutture e servizi tecnologici avanzati senza dover investire in hardware fisico. Questa caratteristica è particolarmente significativa in un'epoca

in cui la velocità di innovazione e la capacità di adattamento sono fattori critici per il successo aziendale. Azure si distingue per la sua capacità di supportare una vasta gamma di servizi, che spaziano dal calcolo al networking, dall'archiviazione all'analisi dei dati, fino all'implementazione di soluzioni basate sull'intelligenza artificiale.

Un aspetto cruciale di Azure è la sua flessibilità, che consente alle organizzazioni di scalare le risorse in modo dinamico, in linea con le mutevoli esigenze del business. Questa capacità di scalare non solo riduce i costi associati alla gestione delle infrastrutture proprie, ma permette anche di rispondere rapidamente alle variazioni della domanda, generando così una maggiore efficienza operativa. Inoltre, Azure offre un ambiente integrato e sicuro, dove le applicazioni possono essere sviluppate, testate e distribuite con maggiore rapidità, contribuendo ad accelerare l'immissione sul mercato di nuovi prodotti e servizi.

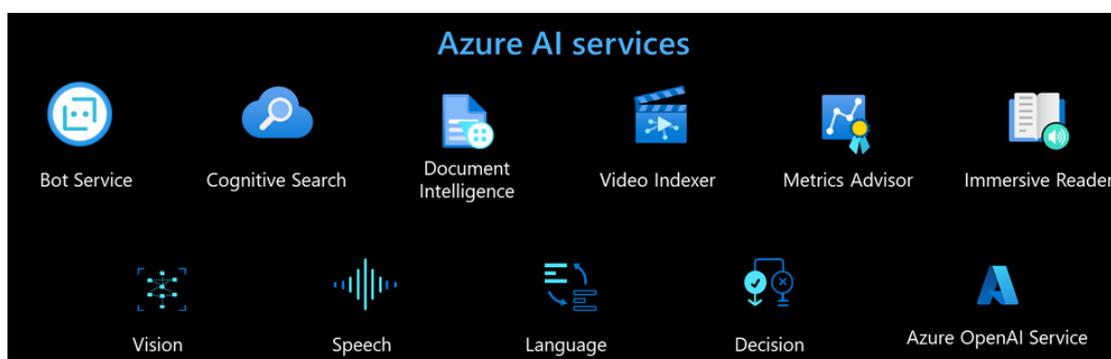


Figura 2.5: Azure AI Services

2.6.1 Azure AI Search

Azure AI Search si distingue come uno strumento fondamentale nell'ambito dell'ecosistema Azure, progettato per soddisfare crescenti esigenze delle aziende che devono gestire e ricercare grandi volumi di dati [25]. A differenza dei motori di ricerca tradizionali, Azure AI Search offre funzionalità avanzate che vanno oltre la semplice ricerca di parole chiave. La capacità di indicizzare e analizzare sia dati strutturati che non strutturati consente di creare esperienze di ricerca estremamente raffinate, in grado di restituire risultati pertinenti in modo rapido e preciso. La ricerca full text è una delle caratteristiche principali di questo servizio, permettendo agli utenti di effettuare ricerche complesse senza sacrificare la velocità o la precisione dei risultati [25].

Un altro punto di forza di Azure AI Search è l'integrazione di funzionalità di analisi linguistica avanzata. Strumenti come lo stemming, che permette di riconoscere diverse forme di una parola, e la gestione dei sinonimi, assicurano che le

ricerche non siano limitate a corrispondenze esatte, ma riflettano un'interpretazione ampia e semantica della query. Questa flessibilità linguistica è ulteriormente rafforzata dalla capacità del sistema di comprendere concetti e contesti, il che è particolarmente utile in scenari dove la ricerca deve essere altamente contestuale, come nell'e-commerce o nella gestione documentale aziendale [25].

2.6.2 Azure OpenAI

OpenAI si afferma come un leader nel campo dell'intelligenza artificiale, con un focus particolare sull'elaborazione del linguaggio naturale. I modelli di linguaggio, come il Generative Pre-trained Transformer, più comunemente conosciuto come GPT, hanno segnato una svolta significativa nel modo in cui le macchine possono comprendere e generare testo. La loro capacità di creare contenuti coerenti, rispondere a domande complesse e tradurre testi ha ridefinito le potenzialità dell'interazione uomo-macchina; questi modelli non solo producono testi accurati e contestualizzati, ma comprendono anche le sfumature del linguaggio, il che li rende estremamente versatili in molteplici applicazioni.

L'integrazione di OpenAI in Azure, attraverso il servizio Azure OpenAI, rappresenta una tappa fondamentale nell'adozione di queste tecnologie a livello aziendale. Questo servizio consente alle imprese di integrare modelli avanzati direttamente nelle loro applicazioni cloud, abilitando funzionalità come la generazione automatica di contenuti, che può migliorare significativamente la qualità e l'efficienza delle interazioni con gli utenti [26]. Questa combinazione di tecnologie non solo ottimizza le operazioni esistenti, ma apre nuove strade per l'innovazione, consentendo alle aziende di personalizzare profondamente le esperienze utente.

La sinergia tra Azure AI Search e i modelli di linguaggio avanzati di OpenAI costituisce un potente ecosistema che offre alle aziende un notevole vantaggio competitivo. Mentre Azure AI Search è progettato per eseguire richieste complesse su grandi volumi di dati, l'integrazione con i modelli di OpenAI potenzia ulteriormente questa capacità, consentendo di interpretare in modo più sofisticato le query formulate in linguaggio naturale [26]. Questo livello di comprensione semantica permette a Azure AI Search di fornire risultati non solo basati su corrispondenze di parole chiave, ma anche sulla comprensione del significato e del contesto, migliorando così la rilevanza e la precisione delle risposte.

Questo approccio è particolarmente vantaggioso in settori come l'e-commerce, dove la capacità di interpretare correttamente le intenzioni degli utenti può tradursi in esperienze di ricerca più soddisfacenti e in un aumento delle conversioni. Un ulteriore esempio di questa sinergia è la creazione di chatbot intelligenti e assistenti virtuali. Grazie all'elaborazione avanzata del linguaggio naturale, questi strumenti possono gestire domande complesse, offrire risposte precise e personalizzate, e

migliorare l'efficienza operativa, riducendo i tempi di risposta e aumentando la produttività.

2.6.3 Direct Line

Direct Line è un canale di comunicazione offerto da Microsoft nell'ambito dell'Azure Bot Service, concepito per facilitare l'integrazione e la gestione dei bot all'interno di applicazioni e servizi personalizzati. Questo canale agisce come un intermediario tra i bot intelligenti e le applicazioni client, sfruttando un'API RESTful che consente la comunicazione bidirezionale tra il bot e l'utente finale [27]. In pratica, Direct Line offre un'interfaccia programmabile che permette agli sviluppatori di incorporare funzionalità di chatbot in ambienti non strettamente legati alle piattaforme di messaggistica predefinite, come Microsoft Teams, Facebook Messenger o Slack.

La funzione primaria di Direct Line è quella di abilitare la comunicazione tra i bot e le applicazioni sviluppate su misura, offrendo un alto grado di flessibilità, personalizzazione e portabilità. Le aziende possono così integrare bot in modo nativo all'interno delle proprie applicazioni web, mobile o desktop, garantendo un'interazione diretta e senza soluzione di continuità con gli utenti. Inoltre, Direct Line supporta l'autenticazione tramite token, un aspetto cruciale per mantenere elevati standard di sicurezza e protezione dei dati durante le interazioni tra bot e utenti.

2.7 SharePoint

SharePoint di Microsoft è una piattaforma robusta e versatile per la gestione documentale e la collaborazione aziendale, largamente adottata nelle organizzazioni di tutto il mondo per la sua capacità di centralizzare e semplificare l'accesso alle informazioni [28]. Originariamente concepito come un sistema per la gestione dei documenti, SharePoint si è evoluto in una soluzione integrata che consente alle aziende di creare portali intranet, gestire flussi di lavoro complessi, e facilitare la collaborazione su larga scala. Uno degli aspetti più rilevanti di SharePoint è la funzione di repository centralizzato, che permette alle aziende di archiviare documenti e contenuti in modo organizzato e sicuro [28]. Questa centralizzazione non solo migliora l'accesso alle informazioni, ma riduce anche la duplicazione dei dati e garantisce che tutti i membri di un'organizzazione lavorino con le stesse versioni dei file. La funzionalità di gestire delle versioni, in particolare, consente di tracciare e ripristinare modifiche, rendendo SharePoint un'ottima soluzione per il controllo qualità e la governance dei dati.

In un contesto aziendale, SharePoint si distingue anche per la sua capacità di integrazione con altri strumenti Microsoft, come Office 365, Teams e la stessa Power

Platform [29]. Questa integrazione crea un ecosistema collaborativo che favorisce una maggiore produttività e facilita l'implementazione di soluzioni aziendali personalizzate. Inoltre, la piattaforma supporta l'automazione dei flussi di lavoro, permettendo di snellire processi ripetitivi e di migliorare l'efficienza operativa. Dal punto di vista della sicurezza, SharePoint offre sofisticati controlli di accesso che assicurano che solo gli utenti autorizzati possano accedere a contenuti specifici [29]. Questo è particolarmente importante per le aziende che gestiscono informazioni sensibili e devono conformarsi a normative sulla privacy e sulla protezione dei dati. Infine, SharePoint offre potenti funzionalità di ricerca, che consentono agli utenti di trovare rapidamente documenti e informazioni in tutto l'ambiente aziendale, migliorando così la condivisione della conoscenza e la collaborazione.

Grazie alla sua versatilità e alle sue capacità avanzate, SharePoint si è affermato come uno strumento essenziale per la gestione delle informazioni e la collaborazione all'interno delle aziende moderne.

Capitolo 3

Progettazione del Chatbot

La fase di progettazione rappresenta un momento cruciale nello sviluppo del chatbot, in cui si definiscono le basi tecniche e concettuali del sistema. Questo capitolo si apre con una descrizione informale che contestualizza il progetto, spiegando quali sono gli obiettivi principali e le funzionalità essenziali che il chatbot dovrà garantire. tale descrizione fornisce una visione d'insieme del problema che si intende risolvere, permettendo di comprendere le esigenze degli utenti finali e le motivazioni che hanno guidato le scelte progettuali.

Successivamente, si affronta la questione dei requisiti, che sono stati identificati e classificati in funzionali e non funzionali. I requisiti funzionali delineano le capacità operative del chatbot, mentre quelli non funzionali si concentrano su aspetti come l'efficienza, la sicurezza e la facilità di manutenzione. La definizione accurata di questi requisiti è essenziale per garantire che il sistema risponda in modo corretto ed efficace alle necessità degli utenti e alle sfide tecniche intrinseche del progetto.

Infine, il capitolo esplora le architetture progettuali proposte, analizzando come queste siano state ideate per soddisfare i requisiti stabiliti. Vengono valutate diverse soluzioni architetture, con l'obiettivo di scegliere quella più adatta al contesto specifico del progetto. La scelta dell'architettura non si basa solo su considerazioni tecniche, ma ha tenuto conto fin dall'inizio dell'intenzione di risolvere il problema utilizzando tecnologia Microsoft e adottando un approccio low-code. Questo orientamento ha guidato la progettazione, con l'obiettivo di creare una soluzione che non solo fosse in linea con le esigenze emerse ed efficiente dal punto di vista tecnico, ma anche competitiva in termini di costi. Poiché il progetto mira a essere venduto ad altre aziende, è stato fondamentale selezionare un'architettura che mantenesse bassi i costi di attivazione e gli abbonamenti necessari. In questo modo, si è garantito che la soluzione fosse economicamente sostenibile e appetibile per potenziali acquirenti, evitando l'impiego di servizi troppo onerosi che avrebbero potuto scoraggiare l'adozione del chatbot da parte delle aziende.

3.1 Descrizione Informale

Il progetto descritto in questa tesi nasce dalla volontà del team di Cluster Reply DCX di sviluppare una soluzione tecnologica innovativa, capace di sfruttare le potenzialità offerte dalle nuove funzionalità di Microsoft Copilot Studio, in combinazione con l'intelligenza artificiale avanzata. Il team, riconoscendo l'importanza strategica di tali strumenti, ha identificato un'opportunità significativa nella creazione di un sistema che potesse affrontare una delle sfide più comuni per le aziende moderne: la gestione e il recupero efficiente delle informazioni archiviate digitalmente.

In molte organizzazioni, i file e i documenti aziendali sono spesso conservati su piattaforme di archiviazione virtuale, come SharePoint, che offrono soluzioni robuste per la memorizzazione dei dati, ma allo stesso tempo, possono presentare delle sfide notevoli quando si tratta di recuperare informazioni specifiche. Le aziende accumulano quotidianamente un volume crescente di dati, e questa mole di documenti rende sempre più complessa la ricerca di informazioni precise. Spesso, le difficoltà non risiedono nella mancanza di dati, ma piuttosto nella loro localizzazione: è comune che i dipendenti non sappiano esattamente dove si trovi una determinata informazione, in quale documento specifico sia inserita, o persino in qualche cartella essa sia stata archiviata.

Questa problematica non è solo una questione di inefficienza operativa, ma ha ripercussioni dirette sulla produttività aziendale. Il tempo speso dai dipendenti per cercare informazioni all'interno di vasti archivi documentali potrebbe essere significativamente ridotto grazie all'automazione e all'uso dell'intelligenza artificiale. Da questa constatazione è nata l'idea di sviluppare un chatbot documentale, un assistente virtuale capace di interagire con gli utenti e di recuperare informazioni pertinenti dai documenti aziendali in maniera rapida e precisa. Questo strumento non solo ridurrebbe drasticamente i tempi di ricerca, ma contribuirebbe anche a incrementare la produttività, permettendo ai dipendenti di dedicare più tempo ad attività ad alto valore aggiunto, anziché disperderlo in attività ripetitive e a basso rendimento.

Un aspetto cruciale che ha guidato lo sviluppo del chatbot è la gestione delle informazioni sensibili e private presenti nei documenti aziendali. A differenza di altri sistemi di intelligenza artificiale che operano su dati di pubblico dominio e che possono utilizzare le informazioni ottenute dalle conversazioni con gli utenti per migliorare ulteriormente i loro modelli, il chatbot progettato in questo contesto è stato pensato per operare esclusivamente su documenti interni all'azienda. Questo approccio assicura che le informazioni trattate rimangano all'interno del perimetro aziendale, senza essere condivise o esposte a terze parti esterne. La sicurezza dei dati è quindi un elemento centrale del progetto, rendendo la soluzione particolarmente adatta a contesti in cui la riservatezza è fondamentale. A supporto di ciò, il chatbot è stato concepito per autenticare gli utenti, garantendo che solo i membri

autorizzati dell'azienda possano accedere alle informazioni; questo meccanismo di autenticazione non solo protegge i dati sensibili, ma rafforza ulteriormente la sicurezza dell'intero sistema, assicurando che le informazioni riservate siano accessibili esclusivamente a chi ne ha diritto, rispecchiando la struttura aziendale.

In definitiva, il progetto si propone di sviluppare un chatbot documentale che, grazie all'integrazione con tecnologie Microsoft e all'uso dell'intelligenza artificiale, possa risolvere efficacemente le problematiche legate alla ricerca di informazioni aziendali, coniugando la sicurezza dei dati e l'efficienza operativa. Questa soluzione è stata pensata per essere implementata in un ambiente di lavoro reale, con una particolare attenzione alla distribuzione su una piattaforma di chat specifica, per garantire un utilizzo agevole e integrato da parte degli utenti aziendali; in questo modo, il chatbot non solo risponde in modo flessibile alle esigenze specifiche delle aziende, ma offre anche un vantaggio competitivo significativo rispetto ad altre soluzioni presenti sul mercato.

3.2 Requisiti

Il nucleo delle funzionalità del chatbot si concentra sull'efficace recupero e presentazione delle informazioni contenute nei file aziendali ospitati su SharePoint. Il sistema è progettato per rispondere a interrogativi specifici degli utenti riguardanti il contenuto dei file. Questo processo è facilitato dall'implementazione di meccanismi di autenticazione robusti che assicurano che l'accesso alle informazioni sia concesso soltanto agli utenti verificati e autorizzati. Il chatbot, inoltre, deve essere capace di discernere i permessi di accesso degli utenti, limitando la visibilità delle informazioni a quelle pertinenti per ciascun utente in base al loro livello di autorizzazione. Ad esempio, un impiegato standard non potrà accedere ai dati riservati alla dirigenza, ma potrà interrogare il chatbot per ottenere dati dalla propria cartella privata, che a sua volta non sarà interrogabile da nessun altro. Questa capacità di filtraggio delle informazioni basata sui permessi è fondamentale per mantenere la sicurezza e la privacy dei dati aziendali.

Per quanto riguarda i requisiti non funzionali, la velocità di risposta è una priorità: il chatbot deve fornire risposte in un intervallo di tempo ragionevole, tipicamente nell'ordine di alcuni secondi, per non compromettere l'efficienza operativa degli utenti. La sicurezza dei dati è un'altra prerogativa del chatbot; tutte le interazioni con il chatbot devono essere sicure e ogni accesso deve essere autenticato per garantire che le informazioni sensibili siano maneggiate correttamente e rimangano confinate all'interno dei permessi utente assegnati. Un altro requisito fondamentale, è la manutenibilità: il chatbot deve essere, infatti, progettato in modo tale che sia possibile effettuare aggiornamenti, modifiche e correzioni con facilità e minimo impatto sulle operazioni quotidiane. La manutenibilità assicura che il chatbot

possa evolvere in risposta ai continui aggiornamenti tecnologici o alle varie esigenze aziendali, senza richiedere interventi prolungati o complessi; ciò contribuisce a ridurre i tempi di inattività e a garantire una risposta rapida ed efficace alle problematiche che possono emergere durante l'uso quotidiano del sistema. Inoltre, il chatbot deve essere flessibile per essere integrato su diverse piattaforme di messaggistica, in modo da adattarsi alle preferenze tecnologiche di diverse aziende clienti; questa scalabilità permette al chatbot di essere implementato in vari ambienti di lavoro senza richiedere modifiche sostanziali. Infine, è essenziale che l'architettura del chatbot sia facilmente configurabile, per permettere alle aziende di personalizzare le impostazioni di visibilità e accesso ai dati in base alla struttura organizzativa interna.

Questi requisiti rappresentano la base su cui il chatbot deve operare per essere considerato una soluzione efficace e sicura nel recupero delle informazioni, offrendo supporto decisivo alle attività quotidiane delle aziende, riducendo tempi di ricerca e aumentando la produttività complessiva. La progettazione attenta e la chiara definizione di tali requisiti sono indispensabili per il successo del progetto, garantendo che il sistema sia non solo funzionale ma anche sicuro e adattabile alle esigenze in continuo cambiamento del mercato aziendale.

3.3 Architettura Monolitica

3.3.1 Descrizione Tecnica

L'architettura in esame rappresenta un modello di implementazione fortemente centralizzato, incentrato sull'utilizzo esclusivo di Microsoft Copilot Studio per gestire tutte le funzionalità core del chatbot documentale. Questa scelta progettuale implica che tutte le operazioni, dalla gestione dell'autenticazione degli utenti alla navigazione e interrogazione dei documenti SharePoint, fino alla generazione di risposte, sono centralizzate in un unico componente software. Microsoft Copilot Studio è configurato per interfacciarsi direttamente con un ambiente SharePoint privato, utilizzando le potenzialità dell'intelligenza artificiale per facilitare le interazioni. L'architettura supporta l'autenticazione attraverso Entra ID e utilizza chiavi private per garantire la sicurezza nell'accesso ai dati aziendali [30].

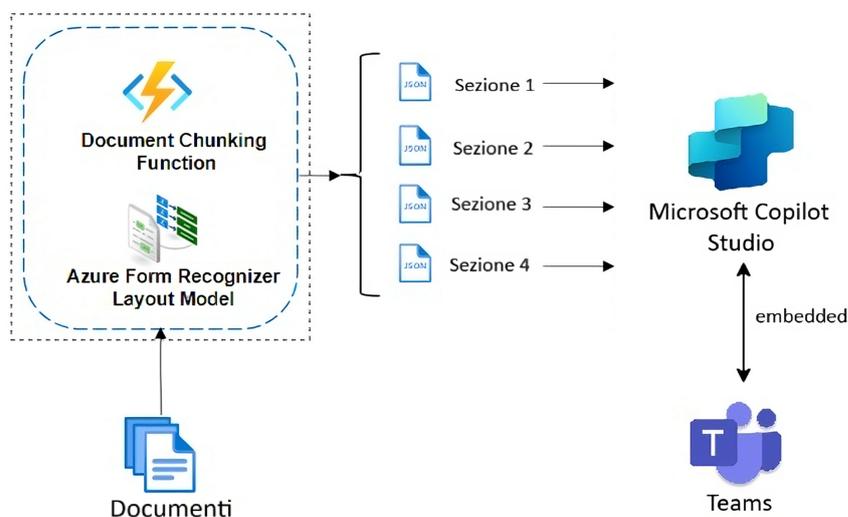


Figura 3.1: Architettura Monolitica basata su Microsoft Copilot Studio

3.3.2 Vantaggi dell'Architettura

L'adozione di un'architettura di questo tipo offre, senza dubbio, una serie di benefici. Uno dei vantaggi più significativi è la centralizzazione delle funzionalità, in quanto, avere un unico punto di controllo per gestire tutte le operazioni riduce la complessità di dover coordinare più sistemi e i relativi processi di comunicazione. Questo non solo minimizza il rischio di errori derivanti da un'integrazione tra più sistemi, ma facilita anche una gestione più coesa e un monitoraggio centralizzato del sistema. Inoltre, elimina i potenziali ritardi associati all'interazione tra componenti distribuiti, riducendo la latenza nelle comunicazioni tra i moduli e migliorando significativamente la velocità complessiva di risposta del sistema.

In aggiunta, Microsoft Copilot Studio facilita enormemente lo sviluppo e l'accesso alle informazioni, offrendo un'interfaccia che astrae le complessità tecniche del recupero delle informazioni da SharePoint e della generazione delle risposte. Tale astrazione consente agli sviluppatori di concentrarsi sull'ottimizzazione delle funzionalità del chatbot, piuttosto che sulle minuzie dell'integrazione dei dati, riducendo così il carico di lavoro e accelerando il processo di sviluppo. Parallelamente, la capacità di Microsoft Copilot Studio di integrarsi con una varietà di piattaforme di messagistica amplia ulteriormente la sua applicabilità, rendendo semplice l'implementazione del chatbot su piattaforme come Microsoft Teams e Facebook Messenger, senza la necessità di adattamenti specifici.

3.3.3 Limitazioni dell'Architettura

Nonostante i numerosi vantaggi offerti dall'adozione di Microsoft Copilot Studio, ci sono delle limitazioni significative che possono influire sulla sua efficacia e sulla sua adattabilità a contesti operativi diversificati. Un primo limite rilevante risiede nella tecnologia di elaborazione del linguaggio naturale adottata da Copilot Studio, che si basa attualmente sul modello GPT-3.5. Sebbene questo modello fosse all'avanguardia al momento del suo rilascio, oggi risulta superato rispetto agli sviluppi più recenti in questo campo, come il modello GPT-4, che offre miglioramenti sostanziali in termini di comprensione e generazione del linguaggio. Questa differenza rende GPT-3.5 meno competitivo e potenzialmente meno efficace in scenari complessi di interazione umana.

Un'altra limitazione critica è legata alla gestione dei documenti all'interno di SharePoint: Microsoft Copilot Studio è in grado di interrogare e gestire solo fino a due livelli di profondità dal punto di accesso fornito, ovvero il link dello SharePoint. Tale restrizione riduce la capacità del sistema di recuperare informazioni da strutture di dati più intricate o profondamente nidificate. Inoltre, Copilot Studio può processare e generare risposte solo per file di dimensioni inferiori a 3 MB, una restrizione che può escludere una vasta gamma di documenti aziendali di dimensioni maggiori, tipicamente ricchi di informazioni utili [31].

Infine, un ulteriore aspetto che merita particolare attenzione è la capacità di Microsoft Copilot Studio di gestire flussi di azioni. Il sistema è principalmente configurato per riconoscere e categorizzare i Topic piuttosto che eseguire o gestire flussi di azioni dinamici. La mancanza di strumenti per analizzare e diagnosticare le cause di fallimento nelle interazioni limita significativamente la manutenibilità del chatbot. Non essendo in grado di identificare con precisione dove e perchè un'interazione fallisce, gli utenti e gli amministratori si trovano spesso di fronte a difficoltà nel correggere gli errori e ottimizzare le prestazioni, compromettendo così l'efficienza operativa complessiva del sistema.

Alla luce dei vantaggi e delle limitazioni esaminati, è chiaro che, nonostante i benefici offerti da Microsoft Copilot Studio, esistono restrizioni significative che limitano la competitività del chatbot, soprattutto riguardo alla gestione dei documenti SharePoint e alla dimensione dei file supportati. Queste limitazioni suggeriscono che, per massimizzare l'efficacia e la competitività dell'architettura, sarebbe opportuno integrare Copilot Studio con altre tecnologie che possono estenderne le capacità. Un approccio ibrido, che combina la potenza di Copilot Studio con componenti aggiuntivi, permetterebbe di superare gli ostacoli correnti, rendendo la soluzione più versatile e pronta ad affrontare una maggiore varietà di sfide operative.

3.4 Architettura Ibrida

L'architettura proposta rappresenta un'evoluzione alla soluzione monolitica precedente, introducendo una serie di componenti aggiuntivi che ne ampliano le capacità e ne migliorano la flessibilità. Sebbene l'incremento nel numero di componenti potrebbe suggerire un aumento sostanziale dei costi, in realtà la struttura economica di questa soluzione è attentamente bilanciata per mantenere i costi accessibili, senza compromettere la qualità del servizio.

3.4.1 Analisi Economica

Dal punto di vista economico, l'architettura non comporta un aumento drastico dei prezzi di attivazione rispetto alla precedente. Microsoft Copilot Studio, componente centrale della soluzione, richiede un abbonamento mensile che va sottoscritto una sola volta per l'intera azienda, indipendentemente dal numero di utenti che utilizzano il chatbot. Questo rappresenta un vantaggio significativo rispetto ad altri servizi, come ad esempio Power Automate, che solitamente richiedono licenze per ogni singolo utente.

L'abbonamento a Microsoft Copilot Studio include 25.000 messaggi complessivi, con la possibilità di aumentare questa capacità in base alle esigenze aziendali qualora il numero di messaggi fosse insufficiente. Inoltre, un aspetto particolarmente vantaggioso di questa architettura è che l'abbonamento a Microsoft Copilot Studio include l'uso gratuito di Power Automate per la gestione di flussi automatici, istantanei e pianificati, che possono essere direttamente richiamati dal Copilot, senza costi aggiuntivi. Considerando che l'abbonamento a Power Automate, in condizioni normali, può rappresentare una spesa non trascurabile, il fatto che sia incluso gratuitamente rende questa scelta non solo economica, ma anche estremamente efficace in termini di funzionalità offerte [32].

Un ulteriore componente fondamentale dell'architettura è Azure AI Search, che richiede anch'esso un abbonamento. Anche con un piano base, Azure AI Search, mette a disposizione funzionalità avanzate, come l'indicizzazione dei documenti SharePoint, offrendo un sistema robusto e scalabile per la ricerca all'interno degli archivi documentali aziendali. Per un'esperienza di ricerca ancora più precisa e sofisticata, è disponibile il servizio di ricerca semantica: sebbene questo comporti un costo aggiuntivo per un numero di richieste superiori a 1000 al mese, la spesa rimane altamente contenuta, esattamente 1 dollaro per ogni 1.000 richieste aggiuntive, rendendolo un'opzione economica per migliorare significativamente la qualità delle ricerche [33].

3.4.2 Descrizione Tecnica

Questa architettura introduce diversi componenti chiave che lavorano in sinergia per fornire un sistema avanzato e più flessibile rispetto alla soluzione monolita vista in precedenza. Al centro di questa architettura si trova Power Automate, che gioca un ruolo cruciale nella gestione dei flussi di azioni, automatizzando il processo e fungendo da ponte di comunicazione tra Microsoft Copilot Studio e gli altri componenti.

Power Automate consente di definire flussi di lavoro complessi che possono includere decisioni condizionali, iterazioni e chiamate a servizi esterni, rendendo il sistema estremamente versatile e adattabile a vari scenari d'uso. Un altro vantaggio significativo dell'integrazione di Power Automate è la sua capacità di monitorare costantemente l'esecuzione dei flussi di azioni che gestisce, facilitando l'individuazione tempestiva di eventuali errori o inefficienze.

Un ulteriore componente fondamentale dell'architettura è Azure AI Search, che è responsabile della creazione e gestione di un indice dei documenti aziendali. In questo contesto, un indice rappresenta una struttura dati altamente ottimizzata che consente di organizzare, catalogare e velocizzare le ricerche all'interno dei documenti indicizzati. L'indicizzazione dei documenti SharePoint è un passaggio essenziale che consente di superare le limitazioni della struttura precedente, dove l'accesso ai documenti era limitato a due livelli di profondità e a file di dimensioni inferiori a 3 MB. Con Azure AI Search, il sistema è ora in grado di gestire documenti di qualsiasi dimensione e complessità, indipendentemente dalla profondità della struttura di cartelle su SharePoint. Questo non solo aumenta la capacità di ricerca e accesso alle informazioni, ma consente anche di restituire risultati molto più rapidi e pertinenti. Inoltre, l'indice creato da Azure AI Search può essere aggiornato dinamicamente, permettendo al sistema di rimanere sempre allineato con i cambiamenti e gli aggiornamenti nei contenuti aziendali. Questo è particolarmente utile in contesti aziendali in cui i dati sono in costante evoluzione.

Il servizio Azure OpenAI gioca, invece, un ruolo fondamentale nell'elaborazione delle query e nella generazione delle risposte. A differenza delle versioni monolitiche precedenti, che si basava esclusivamente sul modello GPT-3.5, Azure OpenAI offre l'accesso a una vasta gamma di modelli di intelligenza artificiale, inclusi quelli più recenti e avanzati come GPT-4, per l'appunto il modello selezionato per questo progetto. In un ambiente dove le richieste degli utenti richiedono un alto grado di comprensione del contesto e di generazione del linguaggio naturale, l'accesso a modelli come GPT-4 permette di fornire risposte estremamente accurate e pertinenti.

Pur mantenendo il suo ruolo centrale all'interno dell'architettura, Microsoft Copilot Studio è ora principalmente utilizzato per comprendere il linguaggio naturale e inoltrare le domande degli utenti al resto dell'architettura. Una volta

che le risposte sono state elaborate dai vari componenti, Copilot Studio si occupa della distribuzione delle risposte all'utente finale, attraverso la piattaforma di messaggistica prescelta.

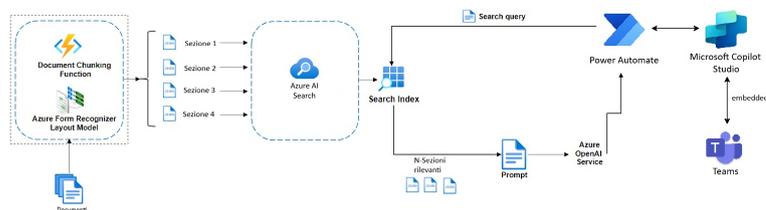


Figura 3.2: Architettura Ibrida con integrazione di Power Automate e Azure AI Search

3.4.3 Vantaggi e Limitazioni

Rispetto alla soluzione monolitica, questa architettura introduce inevitabilmente un po' di latenza aggiuntiva a causa della maggiore complessità e del numero di componenti coinvolti. Tuttavia, questa latenza è compensata da un netto miglioramento della qualità delle risposte e dalla capacità di gestire un numero più variegato di richieste.

Oltre a questi vantaggi, l'architettura offre una serie di benefici aggiuntivi che ne rafforzano la robustezza e la capacità di adattamento a contesti aziendali complessi. La scalabilità è uno dei punti di forza di questa architettura: grazie all'integrazione di componenti come Azure AI Search e Azure OpenAI, il sistema è in grado di adattarsi facilmente a un incremento del volume di dati e delle richieste da parte degli utenti, senza subire un impatto significativo sulle prestazioni.

La modularità di questa architettura permette una gestione più efficiente e mirata delle risorse: ogni componente del sistema è, infatti, specializzato in una funzione ben specifica, e la loro interazione è orchestrata da Power Automate, che garantisce un'esecuzione fluida e combinata. Questa suddivisione delle responsabilità non solo riduce il rischio di colli di bottiglia, ma permette anche una manutenzione più mirata; ad esempio, eventuali aggiornamenti o miglioramenti possono essere implementati in singoli moduli senza interrompere l'intero sistema, riducendo i tempi di inattività e migliorando la continuità operativa.

Tuttavia, la maggiore complessità introdotta da questa architettura potrebbe richiedere competenze tecniche più avanzate, soprattutto nella fase di configurazione iniziale e nell'integrazione dei diversi componenti. Nonostante ciò, l'architettura, data la sua modularità e grazie all'ausilio di Power Automate, che centralizza

gran parte della logica operativa, risulta essere una soluzione facilmente gestibile e monitorabile nel tempo.

I costi aggiuntivi derivanti dall'integrazione di Azure AI Search e Azure OpenAI nell'architettura sono innegabili, ma risultano pienamente giustificati alla luce delle funzionalità avanzate che queste componenti apportano. Queste tecnologie elevano notevolmente la qualità delle ricerche e delle risposte, rendendo il sistema non solo più preciso, ma anche più efficiente nel rispondere alle richieste degli utenti. Pertanto, l'investimento aggiuntivo non è solo comprensibile, ma rappresenta una scelta strategica per le aziende che cercano una soluzione tecnologica solida, in grado di adattarsi ed evolversi con le loro necessità operative.

In conclusione, questa architettura rappresenta un significativo passo avanti rispetto alla soluzione monolitica precedentemente discussa. Sebbene introduca maggiore complessità e alcune sfide gestionali, i benefici che ne derivano, in termini di scalabilità, flessibilità e qualità delle risposte, ne fanno una soluzione decisamente superiore. La capacità di adattarsi a contesti aziendali in evoluzione, unita alla possibilità di sfruttare tecnologie all'avanguardia come GPT-4 e la ricerca semantica, rende questa architettura una scelta ottimale per le aziende che cercano un sistema robusto, versatile e in grado di rispondere efficacemente alle loro esigenze dinamiche.

3.5 Progettazione dell'Interfaccia di Configurazione

Nei paragrafi precedenti sono state analizzate le funzionalità e le componenti principali del chatbot documentale, evidenziando la necessità di un sistema di configurazione che fosse non solo efficiente, ma anche intuitivo e facile da utilizzare. Un aspetto fondamentale emerso è stato quello di garantire una gestione semplice e rapida degli utenti e delle cartellesi è focalizzata sulla realizzazione di un'interfaccia utente che si integrasse perfettamente con l'architettura esistente del chatbot, fornendo un sistema coerente e affidabile per la gestione delle autorizzazioni e assicurando al contempo un'esperienza d'uso semplice e immediata per gli operatori incaricati della gestione.

3.5.1 Scelta dell'approccio

Si è scelto di utilizzare Power Apps per la creazione dell'interfaccia di configurazione in quanto offre un ambiente di sviluppo rapido e flessibile, ideale per la costruzione di applicazioni aziendali che richiedono un'integrazione stretta con l'ecosistema Microsoft, in particolare con Dataverse. Power Apps consente di sviluppare applicazioni con una gestione dei dati strutturati, riducendo i tempi di implementazione senza compromettere la qualità e l'affidabilità del sistema.

Una volta definita la piattaforma, la scelta è ricaduta sulla Model-Driven App all'interno di Power Apps. Questa decisione è stata guidata dalla necessità di gestire flussi di lavoro complessi e grandi volumi di dati strutturati in modo efficiente. La Model-Driven App, a differenza della Canvas App, offre un'interfaccia standardizzata che si adatta perfettamente a scenari dove la coerenza e l'integrazione con Dataverse sono essenziali [15]. Questo approccio ha permesso di concentrarsi sulla logica di business e sull'efficacia operativa, riducendo la necessità di personalizzazioni complesse e garantendo un'implementazione rapida e coerente con le esigenze aziendali.

3.5.2 Progettazione della Struttura Dati

Nella progettazione dell'interfaccia di configurazione, uno degli elementi fondamentali è stato lo sviluppo di una struttura dati che supportasse efficacemente la gestione delle autorizzazioni degli utenti ai documenti contenuti nelle cartelle. Il punto di partenza per questa progettazione è stato l'elaborazione di un diagramma UML, che ha definito le entità principali coinvolte: Contatti, Cartelle e Documenti. Da tale schema, è emerso chiaramente che la relazione chiave da rappresentare e gestire fosse quella tra gli utenti e le cartelle, poiché questa relazione determina quali documenti sono accessibili a ciascun utente.

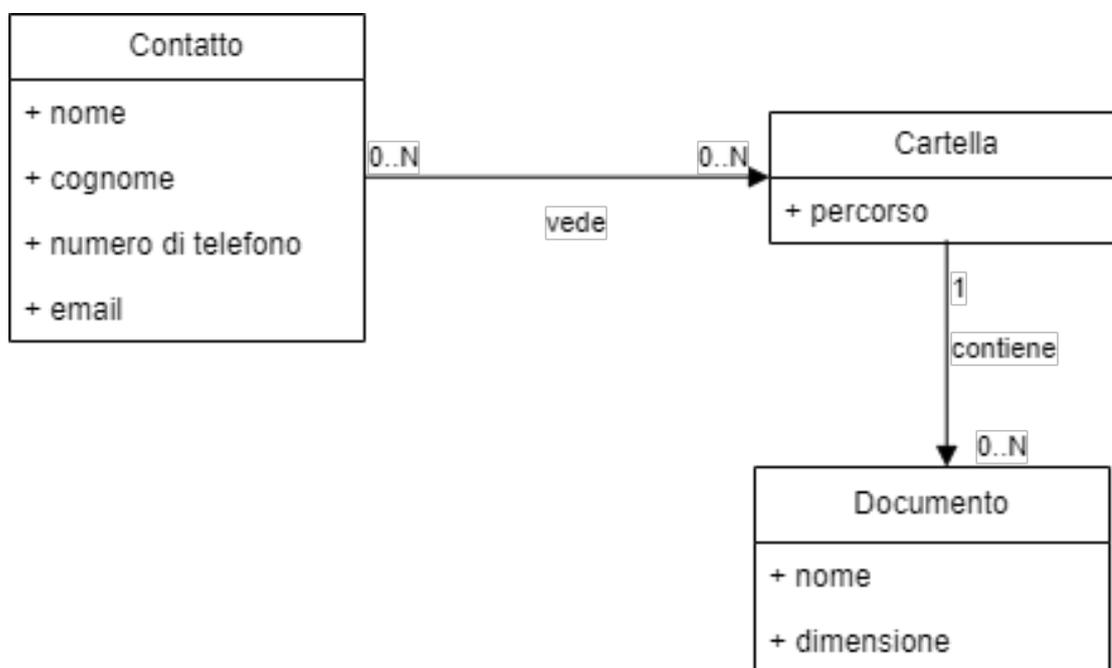


Figura 3.3: Diagramma UML

Per rappresentare gli utenti, è stata selezionata la tabella predefinita “Contatti” di Power Apps, una scelta motivata da diversi fattori strategici. Questa tabella, già ampiamente utilizzata all’interno di Dataverse, è preconfigurata per contenere informazioni su individui con cui una azienda intrattiene rapporti, come clienti, fornitori e colleghi [34]. Tale configurazione la rende particolarmente idonea in un contesto aziendale già integrato con Power Apps, dove la tabella è spesso già popolata e interconnessa con altre funzionalità aziendali, inoltre, la scelta di utilizzare una tabella preesistente oltre a ridurre i tempi di sviluppo garantisce una maggiore coerenza con gli standard operativi dell’azienda. Tuttavia, per adattarla al caso specifico, è stato necessario creare una vista semplificata della tabella stessa, mantenendo solo gli attributi rilevanti per l’interfaccia di configurazione: nome, cognome e ruolo. È stato inoltre introdotto un identificativo unico per ogni utente, basato sul canale di comunicazione utilizzato per interagire con il chatbot: a tal fine, sono stati scelti il numero di cellulare e l’indirizzo email aziendale, poiché costituiscono i principali mezzi di comunicazione con il sistema.

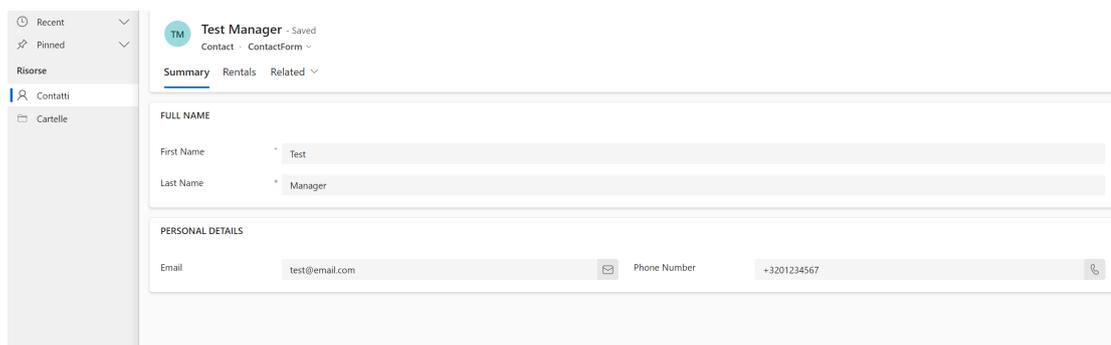


Figura 3.4: Schermata di Power Apps per configurare i Contatti

Parallelamente, è stata progettata la tabella relativa alle cartelle, la quale non è predefinita in Power Apps e ha richiesto una creazione ex novo. La necessità di creare una tabella personalizzata è stata dettata dalla specificità del contesto lavorativo e delle esigenze dell’applicazione. In questo caso, la tabella è strutturata in modo essenziale, prevedendo solo l’attributo “Nome” come identificativo della cartella, semplificando così la gestione delle relazioni con gli utenti e la focalizzandosi sulla funzionalità primaria dell’interfaccia.

La relazione tra le tabelle “Contatti” e “Cartelle” è stata implementata per permettere un controllo granulare sui permessi di accesso ai documenti. L’interfaccia sviluppata consente agli operatori di configurare facilmente questa relazione, garantendo che gli utenti abbiano visibilità solo sui documenti pertinenti. Questo approccio non solo migliora l’efficienza nella gestione delle autorizzazioni, ma assicura anche un alto livello di sicurezza e controllo dei dati all’interno del sistema.

Capitolo 4

Implementazione del Chatbot

Nel presente capitolo, verranno esaminati in dettaglio i processi di implementazione che hanno portato alla realizzazione del chatbot aziendale, concepito per interrogare documenti privati archiviati su SharePoint tramite l'uso di tecnologie avanzate di intelligenza artificiale e integrazioni con sistemi di comunicazione aziendale. L'implementazione di questo progetto rappresenta un esempio concreto di come tecnologie come Azure AI Search, Azure OpenAI Studio, Microsoft Copilot Studio e Power Automate possano essere combinate per creare soluzioni innovative in ambito enterprise.

Il capitolo seguirà un percorso strutturato che parte dalla configurazione iniziale dei servizi utilizzati per l'indicizzazione dei documenti, per poi proseguire con l'integrazione tra i diversi sistemi, passando attraverso la gestione degli accessi e la sicurezza, fino ad arrivare alla descrizione delle specifiche tecniche necessarie per la distribuzione del chatbot su piattaforme di messaggistica esterne, quali Microsoft Teams e Whstapp. Ogni sezione approfondirà gli aspetti tecnici rilevanti e le sfide incontrate, con l'obiettivo di fornire una visione chiara e completa delle soluzioni adottate per garantire il successo del progetto.

4.1 Configurazione dell'Indicizzazione dei Documenti su SharePoint

L'indicizzazione dei documenti su SharePoint è stata una componente fondamentale del progetto, poiché ha reso le informazioni aziendali accessibili e interrogabili tramite un chatbot alimentato da intelligenza artificiale. Questo risultato è stato ottenuto grazie all'utilizzo di SharePoint Online indexer, uno strumento integrato

in Azure AI Search, progettato per connettersi ai siti SharePoint e indicizzare i documenti presenti in una o più librerie documentali. L'indicizzatore estrae e organizza dati e metadati dai documenti in un indice di ricerca strutturato [35].

Oltre a semplificare la connessione tra SharePoint e Azure AI Search, lo SharePoint Online indexer consente di gestire la sicurezza e i permessi di accesso, garantendo che solo le informazioni autorizzate siano incluse nell'indice. Questo processo rende i documenti facilmente ricercabili, permettendo al chatbot di fornire risposte precise ed efficienti, esso inoltre, automatizza la raccolta delle informazioni, assicurando un sistema di ricerca potente e sicuro [35].

Di seguito, viene descritta in dettaglio ogni fase del processo di indicizzazione, con particolare attenzione agli aspetti tecnici e alle configurazioni implementate per garantire un'integrazione ottimale tra SharePoint e Azure AI Search.

4.1.1 Connessione a SharePoint

Per consentire ad Azure AI Search di accedere ai documenti su SharePoint è stato necessario configurare una connessione sicura e delegare i permessi adeguati. In questa fase, scelto di adottare un approccio di least privilege, assegnando all'applicazione solo i permessi strettamente necessari per eseguire l'indicizzazione: questo approccio minimizza i rischi legati alla sicurezza, riducendo l'accesso dell'applicazione alle sole risorse indispensabili.

Per iniziare, è stato creato una registrazione di applicazione in Microsoft Entra ID, la quale ha permesso di configurare i permessi richiesti per l'accesso a SharePoint. Durante questo processo, sono stati abilitati tre permessi specifici: *Sites.Read.All*, *Files.Read.All* e *User.Read*.

- *Sites.Read.All*: Consente all'applicazione di accedere e leggere documenti ed elementi di elenco in tutti i siti SharePoint a cui l'utente autenticato ha accesso [36].
- *Files.Read.All*: Permette all'applicazione di leggere tutti i file che l'utente connesso è autorizzato a visualizzare [36].
- *User.Read*: Consente agli utenti di autenticarsi nell'applicazione, permettendo all'app di accedere ai loro profili e alle informazioni di base relative all'organizzazione di appartenenza [36].

Questa configurazione ha garantito che l'applicazione potesse accedere esclusivamente ai contenuti necessari per l'indicizzazione, evitando esposizioni non necessarie di dati e limitando l'accesso ai documenti, alle strutture di cartelle pertinenti e alle informazioni essenziali sugli utenti autenticati.

La sicurezza è stata inoltre ulteriormente rafforzata tramite la generazione di token di accesso necessari per autenticare l'applicazione durante le operazioni di

indicizzazione. Questo processo ha comportato la creazione di un client ID e di un client secret, che sono stati poi utilizzati nelle richieste HTTP per garantire un'autenticazione sicura [30]. L'adozione di queste misure ha assicurato che solo entità debitamente autorizzate potessero interagire con i dati aziendali, mantenendo un controllo rigoroso su chi e cosa potesse accedere a tali dati.

Inoltre, tutte le operazioni sono state configurate per essere tracciabili e monitorabili, consentendo un'analisi dettagliata delle attività svolte dall'applicazione e offrendo la possibilità di intervenire tempestivamente in caso di anomalie. Questo livello di monitoraggio non solo garantisce la sicurezza del sistema, ma supporta anche la conformità alle politiche aziendali e ai requisiti normativi, fornendo un ulteriore livello di protezione per le informazioni aziendali.

4.1.2 Creazione della Sorgente dei Dati

Una volta configurata la sicurezza e garantito l'accesso ai documenti di SharePoint, il passo successivo è stato la creazione della sorgente dei dati. Essa rappresenta la connessione diretta tra Azure AI Search e il repository di SharePoint, ed è stato configurato utilizzando Postman, uno strumento che facilita l'invio di richieste HTTP.

Attraverso Postman, quindi, è stata inviata una richiesta HTTP Post all'endpoint di Azure AI Search per creare la sorgente dei dati: nella configurazione di quest'ultimo è stato specificato l'URL del sito SharePoint da indicizzare, nonché i permessi delegati precedentemente configurati tramite la registrazione dell'applicazione in Microsoft Entra.

La configurazione della sorgente dei dati è stata progettata per supportare una raccolta continua e aggiornata di dati, con la possibilità di eseguire l'indicizzazione in modo automatico e periodico. Questo ha garantito che l'indice rimanesse aggiornato con le ultime modifiche apportate ai documenti su SharePoint [35].

4.1.3 Creazione dell'Indice

Il passo successivo è stato la creazione dell'indice, che rappresenta la struttura fondamentale in cui i dati indicizzati vengono archiviati e resi disponibili per le interrogazioni del chatbot. Un indice in Azure AI Search specifica i campi presenti in un documento, gli attributi e le altre componenti che modellano l'esperienza di ricerca, in sostanza, l'indice funge da schema per la ricerca, definendo come i dati verranno organizzati, archiviati e successivamente interrogati [35].

Per configurare l'indice, è stato utilizzato nuovamente Postman, inviando una richiesta HTTP Post ad Azure AI Search, definendo la struttura dell'indice e specificando i campi da includere e le loro rispettive proprietà.

I campi selezionati per l'indice includono:

- *id*: Un identificativo unico per ogni documento, che funge da chiave primaria per l'indice, garantendo l'univocità di ogni record.
- *metadata_spo_item_name*: Il nome del documento, utilizzato per identificare e cercare specifici file all'interno dell'indice.
- *metadata_spo_item_path*: Il percorso del documento, che fornisce informazioni sulla posizione del file all'interno della struttura di SharePoint.
- *metadata_spo_item_content_type*: Il tipo di contenuto del documento.
- *metadata_spo_item_last_modified*: La data di ultima modifica del documento, che consente di ordinare i risultati in base alla loro data di aggiornamento.
- *metadata_spo_item_size*: La dimensione del documento, che può essere utilizzata per analizzare il peso dei file archiviati.
- *content*: Il contenuto testuale del documento, che permette di effettuare ricerche all'interno del testo stesso dei file.

Ogni campo nell'indice è stato configurato con specifici flag per determinarne il comportamento durante la ricerca:

- *Searchable*: Indica se un campo può essere incluso nelle query di ricerca testuale. Ad esempio, il contenuto del documento e il nome del documento sono stati configurati come searchable per consentire ricerche precise e pertinenti.
- *Filterable*: Determina se un campo può essere utilizzato per filtrare i risultati di una ricerca. Il percorso del documento è stato reso filterable per permettere agli utenti di restringere i risultati della ricerca a uno specifico sottoinsieme di cartelle.
- *Sortable*: Stabilisce se un campo può essere utilizzato per ordinare i risultati della ricerca. La data di ultima modifica è stata resa sortable per facilitare l'ordinamento dei documenti in base alla loro temporalità.
- *Facetable*: Permette l'aggregazione e la categorizzazione dei risultati in base a un campo specifico, come il tipo di contenuto, per migliorare la navigazione tra i risultati.

Questa configurazione ha permesso di creare un indice altamente strutturato e ottimizzato per le esigenze di ricerca del chatbot, garantendo che ogni campo fosse configurato in modo da rispondere a specifiche esigenze operative.

4.1.4 Configurazione dell'Indicizzatore

L'indicizzato è un ulteriore componente fondamentale in Azure AI Search, che collega una fonte di dati con un indice di ricerca e fornisce una pianificazione automatica per l'aggiornamento dei dati. Dopo aver creato l'indice e la sorgente dei dati, il passo successivo è stato, indubbiamente, configurare l'indicizzatore per automatizzare l'estrazione dei dati da SharePoint e la loro successiva indicizzazione. Esso ha inoltre il compito di gestire la frequenza degli aggiornamenti, garantendo che l'indice rimanga sincronizzato con i cambiamenti nei documenti di SharePoint [35].

L'indicizzatore è stato configurato per operare in modalità incrementale, aggiornando l'indice solo con le modifiche apportate ai documenti dopo l'ultima esecuzione. Questo approccio riduce il carico sul sistema e mantiene l'indice sempre aggiornato senza doverlo ricostruire interamente ogni volta.

Per configurare l'indicizzatore, è stato seguito un processo che ha richiesto l'invio di due richieste HTTP distinte e una procedura di autenticazione. Inizialmente, è stata inviata una richiesta HTTP Post per creare l'indicizzatore, configurandolo per collegarsi alla sorgente dati precedentemente creata e inserire i dati nell'indice target. In questa prima chiamata, oltre a specificare la data source e l'indice, è stato anche indicato quali estensioni di file dovevano essere indicizzate o escluse. Ci si è concentrati su file testuali comuni, come docx, pdf e ppt, escludendo file di tipo immagine come png e jpg, poiché non erano rilevanti per le necessità di ricerca del chatbot. Durante questo passaggio, l'indicizzatore richiede l'autenticazione di un utente con i permessi necessari per accedere ai documenti su SharePoint.

Per garantire l'accesso completo ai documenti necessari, è stato utilizzato un account amministratore per l'autenticazione. Questa fase è stata completata inviando una seconda richiesta HTTP Get, la quale ha restituito un codice dispositivo per un accesso temporaneo, che doveva essere inserito entro 10 minuti dall'invio della prima richiesta HTTP per autorizzare l'indicizzatore e poter procedere.

Una volta completata l'autenticazione e configurato l'indicizzatore, la richiesta iniziale si conclude con un successo se tutti i permessi sono corretti e il processo è stato completato entro il tempo stabilito. Questo assicura che l'indicizzatore sia operativo e pronto per automatizzare il processo di indicizzazione in modo continuo e affidabile.

4.1.5 Aggiornamento della Sorgente dei Dati

Una volta configurato l'indicizzatore, esso viene eseguito automaticamente secondo una pianificazione predefinita, senza richiedere ulteriori interazioni da parte dell'utente, purchè non vi siano modifiche alla sorgente dei dati. Tuttavia, se quest'ultima viene modificata mentre il codice dispositivo è scaduto, sarà necessario ripetere la procedura di autenticazione per consentire all'indicizzatore di eseguire nuovamente

le sue operazioni. Esistono due metodi principali per aggiornare la sorgente dei dati e rieseguire l'indicizzatore [35].

Il primo metodo prevede l'utilizzo dell'interfaccia grafica di Azure AI Search, che offre un'opzione comoda e intuitiva per eseguire manualmente l'indicizzatore. Attraverso l'interfaccia, è possibile utilizzare il pulsante "Rerun Indexer", che consente di aggiornare l'indice con le ultime modifiche apportate ai documenti su SharePoint in qualsiasi momento. Questo approccio è particolarmente utile per gli utenti che preferiscono tale gestione tramite interfaccia grafica e in maniera diretta, senza dover ricorrere a strumenti di programmazione.

Il secondo metodo, invece, consiste nell'esecuzione dell'indicizzatore tramite una richiesta HTTP, come descritto in precedenza. Sebbene richieda un'interazione più tecnica, questo approccio ha il vantaggio di poter essere facilmente automatizzato, ad esempio con un flusso su Power Automate che esegua l'indicizzatore automaticamente in base alle modifiche apportate su SharePoint o a intervalli regolari. Questo assicura che l'indice rimanga costantemente aggiornato senza necessità di interventi manuali.

4.2 Configurazione del Chatbot

L'integrazione di Azure OpenAI Studio nella mia architettura ha reso possibile l'utilizzo di modelli di intelligenza artificiale avanzata, come GPT-4, per permettere al chatbot di interrogare l'indice creato su Azure AI Search e fornire risposte precise basate sul contenuto dei documenti aziendali. Dopo aver indicizzato i dati presenti su SharePoint, Azure OpenAI Studio ha facilitato il collegamento tra la struttura dati e il modello linguistico, consentendo al chatbot di comprendere e rispondere a domande complesse in linguaggio naturale [37].

Con Azure AI Studio, è stato possibile creare un chatbot capace di interpretare non solo termini esatti, ma anche il contesto più ampio delle query, sfruttando la potenza del linguaggio naturale [26]. Questo va ben oltre la semplice ricerca basata su parole chiave, permettendo al chatbot di analizzare e comprendere i documenti in maniera più sofisticata. L'integrazione con il modello GPT-4 ha garantito risposte più accurate, con una maggiore comprensione semantica e una gestione ottimale delle complessità linguistiche.

La capacità di Azure OpenAI Studio di sfruttare l'indice creato ha trasformato il chatbot in uno strumento che interpreta richieste complesse e restituisce risposte contestuali. L'architettura sviluppata non si limita alla mera estrazione di informazioni, ma gestisce l'elaborazione del significato delle domande, analizzando i dati in profondità e restituendo risposte precise, anche in situazioni dove il linguaggio può variare tra le richieste dell'utente e i contenuti dei documenti. Questa sinergia tra i dati aziendali indicizzati e l'intelligenza artificiale generativa ha elevato la qualità

delle interazione, rendendole più naturali e fluide, caratteristiche indispensabili in un ambiente aziendale dove è necessaria una comprensione approfondita per garantire risposte accurate e contestuali.

4.2.1 Scelta del modello

Quando si configura un chatbot basato sull'intelligenza artificiale, la scelta del modello è fondamentale per garantire risposte precise e pertinenti. Per implementare il servizio di Azure OpenAI Studio, è stato scelto di effettuare la distribuzione dell'ambiente nella regione EastUS. Questa scelta è stata dettata dal fatto che non tutti i modelli di intelligenza artificiale fossero disponibili in ogni regione geografica, ed EastUS si è rivelata una delle aree che offre una gamma più vasta e completa di opzioni [38]. Disporre di un'infrastruttura regionale così ricca di modelli ha facilitato notevolmente l'implementazione del chatbot, garantendo l'accesso a risorse robuste e performanti per soddisfare le esigenze operative del sistema. Durante il progetto, è stato analizzato diversi modelli disponibili su Azure OpenAI Studio, tra cui GPT-3, GPT-3.5-turbo, e versioni più avanzate come GPT-4, al fine di identificare la soluzione più adatta per l'interrogazione dei documenti aziendali.

Il modello GPT-3.5-turbo-16k, ad esempio, rappresenta una scelta intermedia molto interessante grazie alla sua capacità di gestire fino a 16.000 token, rendendolo utile in contesti che richiedono l'elaborazione di prompt più lunghi [25]. Tuttavia, in un ambiente aziendale dove è necessaria una comprensione profonda del linguaggio tecnico e una capacità di gestire documenti complessi e formali, si è deciso di optare per GPT-4-32k.

GPT-4-32k si distingue per la sua capacità di gestire fino a 32.000 token, il che lo rende ideale per l'elaborazione di documenti lunghi e complessi come manuali tecnici e report aziendali. Questo modello è in grado di cogliere le sfumature linguistiche e rispondere a richieste più articolate, offrendo un livello di comprensione contestuale superiore rispetto ai modelli precedenti. È particolarmente efficace nella gestione di domande complesse e nella restituzione di risposte pertinenti anche quando i termini della query non corrispondono esattamente alle parole contenute nei documenti, grazie alla sua avanzata capacità di comprensione semantica. La scelta di GPT-4-32k è stata ulteriormente supportata dalla necessità di garantire risposte accurate e coerenti in un contesto aziendale, dove ogni risposta deve essere precisa e basata su dati specifici. Rispetto a GPT-3 e GPT-3.5-turbo, GPT-4 offre una precisione maggiore nel comprendere e gestire richieste dettagliate, minimizzando il rischio di risposte non pertinenti o fuori contesto.

4.2.2 Configurazione della Ricerca

L'ultimo passaggio nella configurazione del chatbot su Azure OpenAI Studio è stata la selezione della tipologia di ricerca più adeguata per l'interrogazione dell'indice creato su Azure AI Search. Di default, Azure AI Search utilizza una modalità di ricerca per parole chiave (keyword o full-text search), che si basa sulla corrispondenza esatta dei termini inseriti nella query [39]. Sebbene questo approccio sia efficiente per interrogazioni semplici, presenta limitazioni quando si affrontano domande complesse, poiché non tiene conto del contesto semantico delle parole. In un contesto aziendale, dove i documenti possono contenere sinonimi, varianti linguistiche o concetti complessi, la ricerca per parole chiave può risultare inefficace, ad esempio, restituendo solo i documenti che contengono esattamente solo i termini ricercati, si rischia di escludere eventuali documenti rilevanti che trattano concetti simili ma espressi con termini diversi.

Per superare questi limiti, è stato configurato il chatbot di modo che operasse tramite ricerca semantica, la quale utilizza modelli avanzati di machine learning per comprendere il significato complessivo delle domande, andando oltre la semplice corrispondenza testuale [40]. La ricerca semantica permette di interpretare le query nel loro contesto e di restituire risultati che rispondono non solo ai termini esatti inseriti, ma anche ai concetti correlati. Questo è particolarmente vantaggioso quando il vocabolario usato dall'utente differisce da quello presente nei documenti: ad esempio, cercando "politiche di sicurezza", la ricerca semantica non si limita a individuare documenti contenenti esattamente questi termini, ma può includere anche documenti che parlano di "procedure di sicurezza" o di "misure di protezione", migliorando così la pertinenza dei risultati.

Uno degli aspetti più utili della ricerca semantica è, senza dubbio, la capacità di stabilire un ranking dei risultati basato su diversi fattori, quelli da me impostati in questo scenario sono stati il nome del documento e il suo contenuto [40]. Questo permette di ordinare i risultati della ricerca non solo in base alla presenza di parole chiave, ma anche in funzione della rilevanza semantica rispetto alla domanda dell'utente. In questo contesto, i documenti più pertinenti vengono mostrati per primi, riducendo il tempo necessario per trovare informazioni richieste. Tale funzionalità è particolarmente utile in presenza di grandi volumi di documenti, tipici dei contesti aziendali, dove è fondamentale ottenere rapidamente le risposte più accurate.

Infine, la ricerca semantica è in grado di migliorare continuamente grazie ai meccanismi di apprendimento automatico [40]. Man mano che viene utilizzata, essa ottimizza progressivamente i risultati sulla base del feedback degli utenti, rendendola un sistema dinamico e in grado di adattarsi meglio alle esigenze di ricerca nel tempo. Questo approccio garantisce che il chatbot non solo restituisca risposte pertinenti, ma che lo faccia in modo sempre più efficiente, migliorando

l'esperienza utente e la precisione delle informazioni recuperate.

4.3 Integrazione tramite Power Automate

Dopo aver configurato il chatbot su Azure OpenAI Studio, il passo successivo è far sì che esso riceva le richieste degli utenti provenienti da Microsoft Copilot Studio. Mentre Azure OpenAI Studio offre una chat sandbox utile per testare il chatbot, questo ambiente non è sufficiente per gestire le interazioni reali. Pertanto, si rende necessario un meccanismo che permetta di inoltrare i messaggi degli utenti da Microsoft Copilot Studio al chatbot su Azure OpenAI, e viceversa, reindirizzando le risposte generate dal chatbot verso Copilot Studio. A tal fine, è stato utilizzato Power Automate, che funge da ponte tra questi due servizi, automatizzando il flusso delle richieste e delle risposte in modo efficace e senza interruzioni.

4.3.1 Configurazione di Microsoft Copilot Studio

Il primo passo del processo di integrazione riguarda la configurazione di Microsoft Copilot Studio, che rappresenta l'interfaccia di comunicazione diretta con l'utente. In questo contesto, è stato utilizzato un Topic di sistema, denominato Conversation Boosting, la cui funzione principale è generare risposte basate su dati provenienti da fonti specifiche [41]. Tuttavia, è stato personalizzato il comportamento di questo Topic, assegnandogli una priorità maggiore rispetto agli altri trigger predefiniti: in particolare, se il messaggio dell'utente non corrisponde a nessun altro trigger configurato, Conversation Boosting si attiva, garantendo che la richiesta venga elaborata. Questo meccanismo assicura che il sistema non lasci senza risposta alcuna query dell'utente.

In questo flusso, Conversation Boosting non risponde direttamente all'utente, ma raccoglie il messaggio e lo memorizza in una variabile di sistema. Questa variabile rappresenta l'input per il passaggio successivo, che avviene tramite Power Automate. Il messaggio dell'utente viene infatti trasferito a Power Automate, che è responsabile di inviarlo ad Azure AI Search per ottenere una risposta pertinente.

La configurazione di Conversation Boosting garantisce che il Topic si attivi solo quando nessun altro trigger è rilevato, mantenendo un'operatività continua ed evitando che le richieste dell'utente rimangano in sospeso. Questo è particolarmente importante in scenari in cui le query non attivano comandi predefiniti. In definitiva, Microsoft Copilot Studio gestisce l'interfaccia utente, mentre la logica e l'elaborazione dei contenuti avvengono tramite l'integrazione con Power Automate e Azure OpenAI Studio.

4.3.2 Flusso di Lavoro su Power Automate

Dopo aver configurato Copilot Studio per raccogliere i messaggi degli utenti, il passo successivo è la costruzione del flusso di lavoro su Power Automate, che permette di inoltrare l'input dell'utente a Azure OpenAI Studio e di restituire una risposta significativa. Il flusso di Power Automate parte dal trigger che raccoglie i dati provenienti da Microsoft Copilot Studio, ossia il messaggio dell'utente, e li prepara per l'invio a Azure OpenAI Studio.

Il primo step consiste nell'acquisire l'input dell'utente da Microsoft Copilot Studio, che lo passa direttamente a Power Automate al momento dell'attivazione del flusso di azioni. Una volta ricevuto questo input, Power Automate lo elabora e lo converte nel formato appropriato per l'invio a Azure OpenAI tramite una richiesta HTTP, garantendo così l'interazione efficace con il servizio di ricerca e intelligenza artificiale.

La parte fondamentale di questo processo è rappresentata dalla configurazione della richiesta HTTP. La richiesta viene costruita utilizzando l'endpoint di Azure OpenAI Studio, che contiene tutte le informazioni necessarie per identificare il servizio e il modello configurato. Nel mio caso, l'URI del servizio è rappresentato dal nome del chatbot su Azure, concatenato al modello GPT-4-32k e alla versione dell'API [42]. Oltre all'URI, è fondamentale fornire nell'header della richiesta l'API key generata per il chatbot, che consente di autenticare l'accesso al servizio Azure. Questo passaggio è essenziale per garantire che solo gli utenti o i processi autorizzati possano inviare richieste al modello di intelligenza artificiale.

Il corpo della richiesta HTTP è un oggetto JSON composto principalmente da due campi:

- *data_sources*: Questo campo contiene i dettagli relativi alla fonte di dati utilizzata per ottenere le informazioni necessarie. In questo caso, si tratta di "azure_search", che permette di eseguire ricerche avanzate all'interno dell'indice preconfigurato su Azure. Nel JSON, oltre a indicare la sorgente dati, devono essere specificati ulteriori parametri come l'endpoint dell'indice di ricerca, il nome dell'indice, la chiave di accesso, il tipo di ricerca e eventuali filtri specifici [42].
- *messages*: Questo campo è l'elemento centrale che definisce il messaggio dell'utente [42]. È strutturato in due sotto-campi: *role* e *content*. Il campo *role* specifica il tipo di messaggio, che in questo caso è impostato su "user", mentre *content* contiene il testo dell'input inviato dall'utente, raccolto dal Topic Conversation Boosting in Microsoft Copilot Studio. Questa parte del JSON è essenziale per trasferire il messaggio al modello GPT-4, che lo utilizzerà come prompt per generare una risposta adeguata.

URI *

https://[redacted].azure.com/openai/deployments/gpt-4-32k/chat/completions?api-version=2024-02-15-preview

Method *

POST

Headers

Content-Type	application/json	✕	🔒
api-key	[redacted]		

Body

```
{
  "data_sources": [
    {
      "type": "azure_search",
      "parameters": {
        "endpoint": "https://[redacted].net",
        "index_name": "sharepoint-index",
        "semantic_configuration": "Semantic1",
        "query_type": "semantic",
        "fields_mapping": {},
        "in_scope": true,
        "role_information": "You are an AI assistant that helps people find information.",
        "filter": "search.ismatch('{x} Path ', 'metadata_spo_item_path')",
        "strictness": 3,
        "top_n_documents": 5,
        "key": "[redacted]"
      }
    }
  ],
  "messages": [
    {
      "role": "user",
      "content": "[redacted] Question "
    }
  ]
}
```

Figura 4.1: Richiesta HTTP al chatbot

4.3.3 Elaborazione della Risposta

Una volta inviata la richiesta HTTP, Azure OpenAI Studio elabora il messaggio e restituisce una risposta sotto forma di JSON. La risposta contiene i vari campi che devono essere interpretati da Power Automate per poter restituire una risposta finale all'utente. Il campo più importante della risposta è `choices[0].message.content`, che contiene il testo della risposta generata dal modello di intelligenza artificiale [42]. Questo messaggio viene poi processato da Power Automate per essere successivamente reinviato a Copilot Studio. Il primo passo riguarda il parsing della risposta JSON ricevuta da Azure OpenAI Studio. Power Automate deve elaborare attentamente la struttura del JSON per estrarre il contenuto della risposta, il quale schema include diversi campi, come:

- *content*: Il contenuto della risposta generata.
- *choices*: Un array che può contenere diverse opzioni di risposta. Viene preso in considerazione il primo elemento (`choices[0]`), in quanto esso rappresenta la risposta più rilevante.
- *tokens*: Includono informazioni sui token utilizzati dal modello per l'elaborazione della richiesta e la generazione della risposta, come il numero di token di prompt e token di completamento.

Dopo aver estratto il campo `content` dal primo elemento di `choices`, Power Automate utilizza questo valore per restituire la risposta finale, che viene successivamente inoltrata a Microsoft Copilot Studio, il quale, a sua volta, ricevuto il messaggio elaborato, lo presenta all'utente come risposta alla query originale. È importante, inoltre, ricordare che l'intero processo di parsing è completamente automatizzato, garantendo che le risposte siano gestite in modo rapido ed efficiente, senza margine di errore.

Grazie a questa integrazione tra Microsoft Copilot Studio, Azure OpenAI Studio e Power Automate, il sistema è in grado di fornire risposte accurate e pertinenti, sfruttando l'intelligenza artificiale per gestire e risolvere anche richieste complesse in modo efficace.

4.4 Gestione degli Accessi e Sicurezza

Il chatbot aziendale è ora in grado di processare le richieste degli utenti e fornire risposte precise basate sui documenti archiviati su SharePoint. Tuttavia, affinché questo sistema operi in un ambiente sicuro e conforme alle politiche aziendali, è essenziale implementare dei meccanismi rigorosi di controllo degli accessi. La gestione della sicurezza assume particolare rilevanza in ambienti aziendali, dove le

informazioni sensibili devono essere accessibili solo da utenti autorizzati, in base ai rispettivi ruoli e permessi. In questa sezione, verrà trattato come Dataverse è stato utilizzato per gestire l'identificazione degli utenti e la determinazione dei permessi di accesso, e come le query OData sono state configurate per limitare l'accesso ai documenti in base ai ruoli definiti.

4.4.1 Introduzione a OData

Prima di analizzare come vengono gestiti i permessi degli utenti e come le query filtrano i risultati all'interno del sistema, è utile introdurre il protocollo OData (Open Data Protocol), un protocollo standard sviluppato da Microsoft per facilitare l'accesso e la manipolazione di dati strutturati tramite API RESTful. OData consente di eseguire query complesse in modo efficiente utilizzando URL per rappresentare le risorse e parametri come *\$filter*, *\$expand*, *\$select*, e *\$orderby* per navigare, filtrare e ordinare i dati [43].

Nel contesto di un'architettura di chatbot aziendale, OData risulta particolarmente utile per controllare l'accesso ai documenti e ai metadati [44]. Ad esempio, con il parametro *\$filter*, è possibile filtrare i risultati di una query per selezionare solo i dati rilevanti, migliorando così l'efficienza del sistema e riducendo il carico sulle risorse. Il parametro *\$expand*, invece, consente di recuperare entità correlate in una singola chiamata, come nel caso di un utente e le cartelle a cui ha accesso.

L'utilizzo di OData è una componente chiave non solo per ottimizzare l'interazione con i dati ma anche per garantire che solo i documenti pertinenti e autorizzati siano interrogati e visualizzati dal chatbot, rafforzando le misure di sicurezza.

4.4.2 Controlli di Accesso tramite Dataverse

Una delle principali sfide nell'implementazione di chatbot in contesti strettamente aziendali riguarda la necessità di garantire che solo gli utenti autorizzati possano accedere ai contenuti pertinenti. Per affrontare questa sfida, è stato integrato Microsoft Dataverse, un sistema di gestione dei dati che consente di centralizzare e proteggere le informazioni relative agli utenti e alle risorse disponibili per ciascuno di essi. Questo approccio consente di mantenere un alto livello di sicurezza, assicurando che ogni richiesta venga processata solo dopo un controllo rigoroso sui permessi.

In questo scenario, Dataverse memorizza informazioni essenziali riguardanti ogni utente, inclusi i suoi identificativi univoci e l'elenco delle cartelle aziendali a cui ha visibilità. Questi identificativi, come l'indirizzo email o il numero di cellulare, vengono utilizzati come chiavi di accesso per determinare i permessi di ciascun utente, in base al canale di comunicazione utilizzato. Quando un utente invia una richiesta tramite il chatbot, il Microsoft Copilot Studio cattura il suo identificativo univoco e il canale di provenienza del messaggio. Queste informazioni sono inviate

a Power Automate, che esegue una query su Dataverse per verificare se l'utente è presente nel sistema. Se l'utente è riconosciuto, vengono recuperati tutti i dati relativi alle cartelle e ai documenti cui ha accesso.

Il flusso di lavoro su Power Automate è stato, di conseguenza, esteso per gestire questa logica.

```
{
  "type": "OpenApiConnection",
  "inputs": {
    "parameters": {
      "entityName": "Contacts",
      "$filter":
        "@{if(equals(triggerBody()?['channel'], 'msteams'), '
new_idteams', 'mobilephone')} eq '@{triggerBody()?['id']}'",
      "$expand": "new_Contact_new_Cartella_new_Cartella"
    },
  },
}
```

Nel dettaglio, la query inviata a Dataverse utilizza l'entità "Contacts" per cercare le informazioni sugli utenti. Questa query include un filtro personalizzato che seleziona l'attributo corretto per l'identificazione, come ad esempio *new_idteams* per le richieste provenienti da Microsoft o *mobilephone* per quelle inviate tramite Whatsapp, verificando se l'identificativo univoco dell'utente corrisponde ai valori archiviati nel sistema. Inoltre, grazie al filtro *\$expand*, è possibile recuperare in una singola chiamata non solo le informazioni sull'utente, ma anche le relazioni con le cartelle su cui quest'ultimo ha visibilità.

Dopo aver recuperato tali le informazioni, entra in gioco un altro componente essenziale del flusso: un ciclo foreach.

```
{
  "type": "Foreach",
  "foreach": "@outputs('List_rows')?['body']['value'][0]['Cartella
']",
  "actions": {
    "Compose": {
      "type": "Compose",
      "inputs":
        "@if(empty(variables('Path')), items('Apply_to_each')['
Name'],
        concat(variables('Path'), ' | ', items('Apply_to_each')['
Name']))\r\n"
    },
    "Set_variable": {
```

```
    "type": "SetVariable",
    "inputs": {
      "name": "Path",
      "value": "@{outputs('Compose')}"
    }
  }
}
```

Questo ciclo elabora ogni cartella a cui l'utente ha accesso singolarmente, concatenando i percorsi delle cartelle in un'unica stringa che verrà poi utilizzata per filtrare la richiesta HTTP inviata al chatbot. Questo passaggio è cruciale perché consente di costruire dinamicamente il contesto della ricerca, garantendo che il chatbot acceda solo alle risorse pertinenti per ciascun utente. Il ciclo `foreach` processa quindi ogni cartella e, a ogni iterazione, compone il percorso di ciascuna cartella, concatenandolo con i percorsi già elaborati. In questo modo, si genera un elenco completo e strutturato di percorsi, che può essere passato come parametro nella successiva richiesta HTTP inviata al chatbot. Questo elenco di percorsi viene successivamente utilizzato per filtrare la ricerca all'interno dei documenti archiviati su SharePoint, limitando i risultati ai soli documenti presenti nelle cartelle autorizzate.

4.4.3 Controllo degli Accessi con OData

Una volta identificato l'utente e ottenuto l'elenco delle risorse a cui ha accesso, è necessario limitare l'interrogazione del chatbot ai soli documenti autorizzati. Questo processo viene implementato attraverso l'uso di una query OData, un potente linguaggio di query che permette di filtrare risultati in base ai metadati associati ai documenti.

Per garantire che la ricerca effettuata dal chatbot rispetti i permessi dell'utente, viene modificata la richiesta HTTP inviata ad Azure OpenAI Studio, aggiungendo un filtro basato sui percorsi delle cartelle ottenuti da Dataverse. Utilizzando la funzione `search.ismatch()` di Odata, il sistema può limitare la ricerca dei documenti contenuti solo nelle cartelle per cui l'utente ha i necessari permessi di accesso [45].

Un esempio di query OData che implementa questa logica è la seguente:

```
search=search.ismatch('Percorso specifico', 'metadata_spo_item_path')
```

In questa query, il campo `metadata_spo_item_path` rappresenta il percorso del documento all'interno di SharePoint. Il valore della query viene confrontato con i percorsi delle cartelle su cui l'utente ha visibilità, permettendo così di restituire solo i documenti rilevanti per quella specifica ricerca. Se un documento non si trova in una cartella

autorizzata, esso verrà automaticamente escluso dai risultati della ricerca, garantendo un controllo rigoroso su chi può accedere a cosa.

Questo documento non solo protegge i dati sensibili, ma migliora anche l'efficienza del sistema, limitando la quantità di documenti esaminati dal chatbot e restituendo solo le informazioni strettamente pertinenti. La possibilità di eseguire ricerche mirate e sicure su grandi quantità di documenti consente una gestione efficace delle risorse, senza compromettere la sicurezza o la velocità delle risposte.

L'adozione di Dataverse per la gestione degli accessi e la configurazione delle query OData per limitare la ricerca ai documenti autorizzati rappresentano una solida base per la sicurezza del chatbot aziendale. Questi meccanismi permettono di controllare efficacemente chi può accedere a determinati contenuti, riducendo il rischio di esposizione non autorizzata dei dati sensibili. La flessibilità offerta da Power Automate e Azure OpenAI Studio ha reso possibile l'implementazione di un sistema sicuro e automatizzato, in grado di rispondere rapidamente alle richieste degli utenti, senza compromessi sulla protezione delle informazioni. Grazie a questa architettura, il chatbot può ora operare in totale sicurezza, gestendo in modo accurato e controllato le interazioni con gli utenti, e garantendo che ogni richiesta venga soddisfatta nel rispetto delle politiche aziendali. In un'epoca in cui la sicurezza dei dati è una delle principali preoccupazioni delle aziende, questa soluzione rappresenta un esempio di come tecnologie avanzate possano essere combinate per affrontare con successo le sfide della protezione dei dati.

4.5 Distribuzione del Chatbot

Una volta completata la configurazione del chatbot, compreso l'aspetto fondamentale della gestione degli accessi, il passaggio successivo consiste nel distribuirlo in modo che gli utenti finali possano effettivamente utilizzarlo nei loro ambienti di lavoro quotidiani. In questo contesto, l'implementazione del chatbot su piattaforme di messaggistica come Microsoft Teams e WhatsApp si rivela strategica per garantire un'interazione fluida e immediata tra il dipendente e il chatbot [46]. Queste piattaforme, ampiamente adottate negli ambienti professionali, offrono interfacce user-friendly che facilitano l'adozione del sistema da parte degli utenti e ne migliorano l'efficienza comunicativa. La distribuzione su queste piattaforme richiede una gestione accurata dei canali di distribuzione e una configurazione precisa dei parametri di sicurezza e autenticazione, garantendo così l'integrità e la protezione dei dati aziendali sensibili.

Microsoft Teams, grazie alla sua stretta integrazione con l'ecosistema Microsoft e Azure, si pone come il canale ideale per la distribuzione del chatbot, offrendo un processo di implementazione semplice e altamente scalabile. Al contempo, la distribuzione su WhatsApp, pur presentando alcune sfide tecniche specifiche, è necessaria per estendere l'accessibilità del sistema a ulteriori modalità di interazione, adattandosi a contesti dove l'uso di questa piattaforma è diffuso anche all'interno dell'organizzazione. Nei paragrafi seguenti verrà illustrato in dettaglio il processo di implementazione su queste due piattaforme, partendo dall'integrazione con Microsoft Teams.

4.5.1 Distribuzione su Microsoft Teams

L'integrazione del chatbot con Microsoft Teams è un passaggio cruciale, reso particolarmente agevole grazie all'ecosistema offerto da Microsoft Copilot Studio. In questo contesto, la piattaforma fornisce un'interfaccia grafica intuitiva per gestire le pubblicazioni del chatbot su diversi canali di comunicazione, tra cui Teams, il quale rappresenta uno degli strumenti di collaborazione più utilizzati a livello aziendale.

Il processo di distribuzione su Microsoft Teams attraverso Copilot Studio si articola in diverse fasi, entrambe gestibili dall'interfaccia utente senza richiedere interventi complessi sul codice o configurazioni avanzate. Una volta completata la configurazione del chatbot, inclusa l'impostazione dei permessi di accesso, l'utente può procedere con la distribuzione attraverso un'interfaccia user-friendly. All'interno di Copilot Studio, è presente una sezione dedicata alla gestione dei canali di distribuzione, da cui è possibile aggiungere facilmente il bot selezionando Microsoft Teams come piattaforma di distribuzione. Grazie all'automatizzazione di diverse operazioni, l'intero processo è semplificato al punto da ridurre notevolmente i potenziali errori, garantendo al contempo un'implementazione rapida ed efficiente del sistema [47].

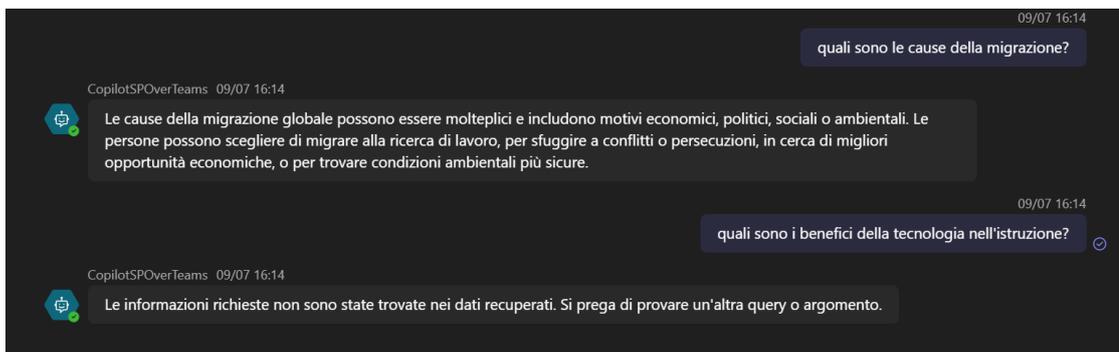


Figura 4.2: Chat su Microsoft Teams

Una delle caratteristiche centrali che rendono questa integrazione particolarmente efficiente è il supporto nativo per l'autenticazione tramite Entra ID, il servizio di gestione delle identità di Azure Active Directory [47]. Grazie a questa integrazione, gli utenti di Microsoft Teams possono accedere al chatbot utilizzando le stesse credenziali aziendali, che impiegano per altre applicazioni di Microsoft 365. Questo aspetto non solo facilita l'accesso, ma garantisce anche che le politiche di sicurezza e controllo degli accessi definite precedentemente siano applicate in modo coerente e sicuro. In tal modo, si evita la necessità di configurare ulteriori meccanismi di autenticazione, riducendo i costi operativi e i tempi di implementazione.

La gestione dell'accesso al chatbot si basa sull'account Microsoft Teams dell'utente, utilizzando la mail associata per identificare chi effettua la richiesta. Questo approccio consente di configurare i permessi di accesso in maniera dinamica e personalizzata, senza la necessità di implementare meccanismi complessi come il controllo degli accessi basato sui ruoli. In questo modo, ogni utente viene autenticato attraverso il proprio account

aziendale su Teams, permettendo al chatbot di rispondere in base alle autorizzazioni assegnate a livello di sistema. Questo metodo assicura la protezione dei dati aziendali sensibili, impedendo che informazioni riservate siano accessibili a utenti non autorizzati.

Una volta completata l'integrazione con Teams, il chatbot sarà disponibile direttamente nella piattaforma di collaborazione, accessibile sia tramite l'interfaccia desktop che mobile. Gli utenti possono semplicemente cercare il bot all'interno della barra di ricerca di Microsoft Teams e iniziare a interagire con esso come farebbero con qualsiasi altro contatto o applicazione disponibile all'interno della piattaforma. La semplicità e l'immediatezza di questa operazione sono essenziali per garantire una user experience positiva e incentivare l'adozione del sistema.

4.5.2 Distribuzione su WhatsApp

L'integrazione del chatbot aziendale con Whatsapp ha richiesto un approccio più articolato e complesso rispetto alla distribuzione su piattaforme più integrate, come Microsoft Teams. A differenza di quest'ultima, che offre un supporto nativo e un processo di distribuzione semplificato attraverso Microsoft Copilot Studio, WhatsApp necessita di un'infrastruttura specifica e di strumenti aggiuntivi per gestire la comunicazione tra il client di messaggistica e il chatbot [48]. In questo contesto, due tecnologie hanno giocato un ruolo cruciale: Twilio, per gestire il canale di messaggistica di WhatsApp, e la Direct Line API 3.0, che permette di creare un ponte tra il sistema di messaggistica e il chatbot tramite un relay server. Questa configurazione ha richiesto una gestione approfondita delle sessioni utente, dell'autenticazione e delle risposte del bot, tutto orchestrato mediante un server backend sviluppato in C#.

Il Ruolo di Twilio

Twilio è una piattaforma leader nel settore della comunicazione cloud, progettata per abilitare l'integrazione di funzionalità di messaggistica, voce e video all'interno di applicazioni e servizi aziendali, offrendo così alle imprese una soluzione flessibile per connettersi con i propri utenti attraverso più canali di comunicazione. Nel contesto di questo progetto, Twilio è stata selezionata per gestire l'integrazione con WhatsApp, sfruttando le sue API per creare una connessione robusta e facilmente personalizzabile tra la piattaforma di messaggistica e il chatbot aziendale [49]. Un elemento centrale offerto da Twilio è la sandbox per WhatsApp, che consente agli sviluppatori di sperimentare rapidamente le funzionalità di invio e ricezione di messaggi prima di implementare una configurazione di produzione completa. Questa sandbox, accessibile dalla sezione Messaging della dashboard Twilio sotto la voce Try it out, fornisce un ambiente di test facile da configurare, ideale per le prime fasi di sviluppo e verifica del sistema. Attraverso, questa sandbox, è possibile inoltre configurare un Webhook, il quale permette di ricevere i messaggi inviati dagli utenti WhatsApp e di inoltrarli al server backend tramite una richiesta HTTP POST. Questo Webhook consente la creazione di un endpoint personalizzato, attraverso cui è possibile gestire i messaggi in tempo reale [50]. Il corpo della richiesta POST include due parametri fondamentali:

- *From*: il numero di telefono dell'utente che ha inviato il messaggio, identificato con il prefisso "whatsapp:" [50].

- *Body*: il contenuto del messaggio inviato dall'utente [50].

Questa struttura facilita il riconoscimento immediato dell'utente e del suo messaggio, consentendo al server di backend di mantenere una sessione attiva tra l'utente e il chatbot, garantendo, quindi, una comunicazione continua e senza interruzioni.



Figura 4.3: Chat su WhatsApp

Implementazione del Relay Server

Per gestire correttamente la comunicazione tra Twilio, e quindi WhatsApp, e il chatbot, è stata utilizzata la Direct Line API 3.0, una soluzione fornita da Microsoft per abilitare l'integrazione tra bot e client esterni attraverso chiamate HTTP RESTful. La Direct Line API è particolarmente adatta per scenari in cui il bot non può essere distribuito direttamente su una piattaforma, come nel caso di WhatsApp, fornendo un meccanismo di comunicazione flessibile e scalabile. Questa API è stata cruciale per instaurare conversazioni bidirezionali tra gli utenti WhatsApp e il chatbot [27]. A differenza dell'integrazione nativa con piattaforme come Microsoft Teams, dove Microsoft Copilot Studio gestisce automaticamente il distribuzione e la configurazione, la Direct Line API richiede una gestione più personalizzata.

Per facilitare la comunicazione tra WhatsApp e il chatbot, è stato implementato un Relay Server in C# basato su ASP.NET Core, che funge da intermediario tra i messaggi ricevuti su WhatsApp e il bot. Questo server gestisce l'inoltro dei messaggi in tempo reale, ricevendo le richieste da Twilio e inviandole al chatbot tramite la Direct Line API. Inoltre, il Relay Server tiene traccia delle conversazioni attive, garantendo che ogni interazione venga gestita correttamente e senza interruzioni. Quando Twilio riceve un messaggio da un utente tramite WhatsApp, invia immediatamente una richiesta HTTP POST al relay server, fornendo il numero di telefono dell'utente e il contenuto del messaggio. Questa richiesta viene gestita dal RelayConnector, una classe appositamente creata nel server C# per ricevere i dati e iniziare o continuare una conversazione con il chatbot.

Uno degli aspetti più critici nella gestione delle conversazioni multiple è l'autenticazione degli utenti e la gestione delle sessioni. Per comunicare con il chatbot tramite la Direct Line API, ogni utente deve essere autenticato con un token, che viene generato attraverso un endpoint specifico fornito da Microsoft Copilot Studio. Questo token permette al relay server di stabilire una connessione sicura tra il client, in questo caso, WhatsApp, e il bot. Nel relay server è stato implementato un dizionario che associa ogni numero di telefono utente a un token e a un watermark, quest'ultimo indica fino a che punto è stata letta la conversazione corrente con il chatbot. Prima di ogni interazione, il server verifica se l'utente ha già ottenuto un token, controllando la presenza del suo numero di telefono nel dizionario, successivamente si assicura che sia valido; in caso contrario, ne richiede uno nuovo attraverso l'endpoint fornito da Copilot Studio. Questo sistema garantisce che ogni utente possa continuare la propria conversazione anche dopo lunghi periodi di inattività, senza perdere il contesto.

Una volta autenticato l'utente, il relay server invia una richiesta POST alla Direct Line API per iniziare una nuova conversazione o continuare quella esistente. La richiesta all'endpoint `"/conversations"` crea una nuova sessione di conversazione con il chatbot, generando un `conversationId` univoco che viene utilizzato per tutte le successive interazioni. Con il `conversationId` ora a disposizione, il relay server costruisce un messaggio strutturato, che contiene il testo inviato dall'utente, il suo numero di telefono e altre informazioni rilevanti come la lingua, questi dettagli sono fondamentali per permettere successivamente al chatbot di identificare correttamente l'utente e di rispondere in maniera adeguata alla sua domanda. Il messaggio viene quindi inviato al chatbot tramite una seconda richiesta POST all'endpoint `"/activities"` della Direct Line API, associata al `conversationId`. Questo garantisce che il messaggio venga elaborato dal chatbot all'interno della sessione corretta.

Elaborazione delle risposte del chatbot

Una volta che il messaggio dell'utente è stato inoltrato al chatbot tramite la Direct Line API, il relay server entra in una fase di attesa per ricevere la risposta del bot. Questo processo, gestito mediante una chiamata GET alla stessa API, ha lo scopo di recuperare tutte le attività associate alla conversazione in corso, incluse le risposte generate dal bot. La gestione delle risposte è un passaggio cruciale per garantire una comunicazione fluida tra l'utente e il chatbot, specialmente in un contesto di messaggistica come WhatsApp, dove è fondamentale che le risposte siano tempestive e coerenti. Per ottimizzare questo processo, viene utilizzato il già citato *watermark*, un indicatore che consente di tracciare lo stato della conversazione e di evitare la duplicazione di messaggi già processati. Ogni volta che il relay server recupera le attività della conversazione tramite una richiesta GET alla Direct Line API, utilizza il watermark associato all'utente specifico; questo parametro permette al sistema di continuare a leggere la conversazione esattamente dal punto in cui era stata interrotta, ignorando i messaggi già processati nelle interazioni precedenti. L'adozione del watermark è particolarmente utile in scenari di lunga durata, dove l'utente può sospendere e riprendere la conversazione in momenti diversi, evitando così ridondanze e garantendo una continuità del dialogo senza confusione. Una volta recuperati i messaggi, si rende necessario un processo di filtraggio. La conversazione restituita dalla Direct Line API, infatti, include non solo le risposte del chatbot, ma anche altre attività, come i messaggi inviati dall'utente e possibili attività di sistema. Per evitare di restituire all'utente i messaggi che egli stesso ha inviato, il relay server filtra i risultati, selezionando esclusivamente le attività di tipo "Message" generate dal chatbot. Questo filtraggio avviene tramite una verifica esplicita del mittente del messaggio, controllando che il campo "From.Name" corrisponda al nome del chatbot. Tale approccio permette di isolare le risposte pertinenti, eliminando tutto ciò che non è rilevante per l'utente finale. Una volta ottenute le risposte sotto forma di una lista di oggetti Activity, queste vengono elaborate dalla funzione *BotReplyAsAPIResponse*. Il compito principale di questa funzione è trasformare le attività ricevute in un formato testuale adeguato per essere inviato attraverso il canale di WhatsApp. La funzione opera iterando su ciascun oggetto della lista e concatenando i testi delle risposte prodotte dal chatbot, in modo da generare una stringa unificata che può essere restituita all'utente finale. In particolare, per ogni attività, viene verificato se il campo "Text" contiene una risposta valida: se presente, il testo viene aggiunto alla stringa di output, inoltre, nel caso in cui l'attività includa azioni suggerite, come opzioni interattive per l'utente, queste vengono concatenate alla stringa sotto forma di elenco, separato da barre verticali, per mantenere un formato leggibile e coerente. Il risultato finale è una stringa completa, che aggrega tutte le informazioni necessarie e che è pronta per essere inviata all'utente tramite il canale WhatsApp, assicurando così una comunicazione efficace e senza interruzioni.

Uno degli aspetti centrali del relay server implementato è la capacità di gestire in modo efficace più conversazioni simultanee con utente diversi. Grazie all'associazione univoca tra numero di telefono e conversazione, garantita dai token e dai *conversationId*, il sistema può scalare facilmente, mantenendo prestazioni elevate e assicurando una gestione fluida anche in scenari di alta concorrenza. Questo approccio distribuito e ottimizzato per la gestione parallela delle interazioni non solo assicura la continuità del servizio, ma migliora anche la resilienza del sistema in caso di carichi di lavoro elevati.

Un ulteriore vantaggio chiave del relay server sviluppato risiede nella sua elevata flessibilità e modularità. La decisione di implementare una funzione separata per la formattazione delle risposte del chatbot, denominata `BotReplyAsAPIResponse`, si basa sul principio di *separation of concerns*. Questo approccio consente di isolare la gestione delle attività provenienti dal chatbot in un modulo dedicato, facilitando così l'aggiornamento o la modifica della logica di elaborazione delle risposte senza compromettere l'intero funzionamento del relay server. Grazie ciò, il sistema risulta non solo più manutenibile e scalabile, ma anche altamente personalizzabile in base a specifiche esigenze aziendali o preferenze degli utenti. La formattazione dei messaggi e la logica di elaborazione delle risposte, infatti, possono essere adattate senza richiedere interventi strutturali, rendendo il sistema compatibile con nuove piattaforme o formati di comunicazione. Nel contesto di questo progetto, ad esempio, le risposte del bot sono state ottimizzate per una conversazione fluida e naturale su WhatsApp, ma il relay server è predisposto per gestire agevolmente altre piattaforme o regole di filtraggio personalizzate, garantendo così una notevole adattabilità del sistema. Inoltre, la configurazione del relay server è stata semplificata mediante l'uso di un file `appsettings.json`, che contiene tutti i parametri necessari per collegare il relay server a uno specifico bot. Questo approccio centralizzato alla configurazione non solo facilita la gestione, ma rende anche il sistema facilmente riutilizzabile in contesti differenti. Modificando i parametri di configurazione, come ad esempio il "BotId" o il "BotTokenEndpoint", è possibile adattare rapidamente il sistema a nuovi chatbot o ambienti operativi, senza dover intervenire sul codice sorgente principale.

Distribuzione del Relay Server su Azure

Per rendere operativo il relay server al di fuori dell'ambiente locale, si è reso necessario pubblicarlo su una piattaforma cloud in modo che fosse accessibile tramite un endpoint pubblico. A tal fine, si è deciso di utilizzare Microsoft Azure, sfruttando i servizi offerti da Azure API Management per distribuire il server in un ambiente di produzione sicuro e scalabile [51].

Per prima cosa, è stato creato un Azure App Service, una piattaforma che fornisce un'infrastruttura scalabile e gestita per eseguire applicazioni web e API. Dopo aver verificato il corretto funzionamento del relay server in locale, tramite l'interfaccia di Visual Studio, è stato possibile avviare la configurazione del servizio scegliendo l'opzione di pubblicazione all'interno del progetto e successivamente selezionando Azure come destinazione di pubblicazione. Questo passaggio ha avviato il processo di configurazione guidata, che ha consentito di collegare l'applicazione a un'istanza di Azure App Service già esistente o, come nel caso di questo progetto, di crearne una nuova [51]. È stato quindi necessario specificare alcuni parametri essenziali, come il nome del servizio, l'area geografica in cui ospitare l'applicazione, e il piano tariffario per determinare le risorse computazionali allocate. Una volta configurato l'ambiente di destinazione, il passaggio successivo ha previsto l'autenticazione su Azure tramite l'account associato all'abbonamento di Microsoft Azure, per garantire l'accesso alle risorse del cloud. Dopo l'autenticazione, Visual Studio ha offerto un riepilogo delle impostazioni di pubblicazione, confermando la configurazione del profilo di distribuzione, che includeva dettagli riguardanti l'endpoint pubblico e le impostazioni di rete.

L'ultimo step della procedura è stato l'effettivo deploy del relay server, durante il quale il codice è stato compilato, pacchettizzato e caricato sull'infrastruttura cloud di Azure App Service. Al termine di questa operazione, il sistema ha generato un endpoint accessibile via HTTP o HTTPS, permettendo al server di ricevere e inoltrare messaggi provenienti da Twilio in tempo reale. Contestualmente, è stata verificata l'integrazione con Azure API Management, un servizio che ha ulteriormente rafforzato la sicurezza e la scalabilità dell'applicazione, fornendo funzionalità come il throttling delle richieste, il monitoraggio delle API e la gestione centralizzata degli endpoint [51]. Questo approccio ha reso possibile esporre in modo sicuro il relay server, mantenendo un controllo rigoroso sugli accessi e garantendo che la comunicazione tra il server backend e gli utenti finali tramite WhatsApp avvenisse senza interruzioni. La flessibilità e la modularità del sistema, combinate con l'infrastruttura cloud di Azure, hanno assicurato una gestione efficiente e una distribuzione scalabile del server in ambiente di produzione.

L'integrazione di WhatsApp con il chatbot aziendale tramite Twilio e la Direct Line API 3.0 ha rappresentato una sfida complessa, ma necessaria per fornire un canale di comunicazione versatile e ampiamente utilizzato in contesti aziendali. L'uso di Twilio ha permesso di gestire la ricezione dei messaggi e la loro trasmissione al bot, mentre la Direct Line API ha garantito una comunicazione sicura e scalabile tra WhatsApp e il sistema del bot. Il relay server sviluppato in C# ha reso possibile la gestione multiutente e ha fornito un livello elevato di personalizzazione, consentendo di adattare il sistema a diverse esigenze aziendali. Tale soluzione si è rivelata particolarmente efficiente nel gestire conversazioni simultanee, garantendo una distribuzione ottimale del carico e una gestione sicura dei token di autenticazione. Grazie alla sua modularità, il sistema può essere facilmente riutilizzato per altri bot, rendendolo una soluzione flessibile e scalabile per futuri progetti aziendali.

Capitolo 5

Risultati e Valutazione

Il presente capitolo ha lo scopo di analizzare i risultati ottenuti nell'ambito del progetto, mettendoli in relazione con gli obiettivi definiti nelle fasi iniziali. Come indicato nel capitolo introduttivo, il principale obiettivo della tesi consiste nella progettazione e implementazione di un chatbot capace di interrogare documenti confidenziali in modo efficiente e sicuro, sfruttando un approccio low-code. Per verificare il raggiungimento di tali obiettivi, è stata condotta una serie di test mirati, che hanno permesso di valutare l'efficacia del sistema sotto diversi aspetti, tra cui la precisione delle risposte, la velocità di elaborazione e la capacità di integrarsi con altri sistemi aziendali.

I risultati riportati sono frutto di una rigorosa validazione delle funzionalità implementate, supportata da una serie di test unitari, di integrazione e end-to-end. Questi test hanno permesso di identificare i punti di forza della soluzione e, al contempo, di individuare alcune aree che potrebbero beneficiare di miglioramenti futuri. In particolare, verranno discussi gli esiti delle valutazioni effettuate e saranno proposti suggerimenti per l'evoluzione del chatbot, tenendo in considerazione le possibili applicazioni aziendali e le opportunità di ottimizzazione delle prestazioni.

5.1 Ambiente di Test e Metodologia

Il processo di validazione del chatbot sviluppato ha richiesto la definizione di un ambiente di test strutturato e la selezione di una metodologia mirata a garantire la copertura completa delle funzionalità previste. Il sistema è stato sottoposto a diverse tipologie di test, suddivisi in test unitari, test di integrazione e test end-to-end, per assicurare il corretto funzionamento di ciascun componente, l'interoperabilità tra i diversi moduli e la qualità dell'esperienza utente finale.

5.1.1 Ambiente di Test

Il contesto di test è stato progettato per simulare un ambiente aziendale realistico, in cui utenti con diversi livelli di autorizzazione potevano accedere e interrogare documenti confidenziali tramite il chatbot. Sono stati creati utenti con profili variegati, ciascuno con

accesso differenziato a una gamma di cartelle su SharePoint, al fine di testare in maniera approfondita i privilegi di accesso e le restrizioni basate sui permessi assegnati. Questo ha consentito di simulare scenari complessi, in cui solo alcuni utenti erano autorizzati a visualizzare determinati documenti, permettendo così una valutazione precisa del funzionamento delle meccaniche di autenticazione e autorizzazione, oltre che del rispetto delle politiche di accesso.

In particolare, i documenti utilizzati per i test sono stati distribuiti in diverse cartelle su SharePoint, con vari livelli di profondità e differenze significative in termini di contenuto, lunghezza, complessità e formato. Sono stati inclusi documenti di tipologia diversa, come file PDF, Word e PowerPoint, al fine di verificare la capacità del chatbot di gestire una gamma eterogenea di dati. Questi documenti spaziavano da testi più brevi a report dettagliati, alcuni dei quali contenevano contenuti tecnici complessi, mentre altri erano caratterizzati da uno stile di scrittura più discorsivo. Tale varietà ha consentito di testare non solo la precisione delle risposte, ma anche la robustezza del sistema nell'interpretare correttamente richieste di natura differente.

5.1.2 Metriche valutate

Durante l'intero processo di test, sono state definite e monitorate diverse metriche per misurare la performance del chatbot e la sua capacità di rispondere agli obiettivi definiti. Le principali metriche valutate includono:

- *Precisione delle risposte*: è stata misurata la percentuale di risposte esatte fornite dal chatbot rispetto al contenuto effettivo dei documenti. In particolare, si è verificato se il sistema fosse in grado di individuare correttamente le informazioni rilevanti e rispondere in modo accurato alle domande poste dagli utenti. Oltre alla correttezza delle informazioni, è stata valutata la capacità del sistema di limitare l'accesso alle informazioni riservate solo agli utenti autorizzati. Ad esempio, se una determinata informazione si trovava in una cartella "Admin", il chatbot non doveva renderla visibile a utenti senza i necessari permessi. Il mancato rispetto di questa restrizione avrebbe compromesso la precisione delle risposte, nonché la sicurezza del sistema.
- *Velocità di risposta*: sono stati monitorati i tempi medi di elaborazione delle richieste, sia per query semplici che per quelle più complesse. Questo ha permesso di valutare l'efficienza del chatbot nel rispondere alle diverse tipologie di richieste, così come la sua capacità di mantenere tempi di risposta competitivi in scenari di crescente complessità.
- *Gestione delle richieste parallele*: il sistema è stato sottoposto a test di carico per verificare la sua capacità di gestire richieste multiple simultanee. Sono stati lanciati più thread concorrenti per simulare richieste simultanee, con l'obiettivo di valutare la stabilità e le prestazioni del chatbot in condizioni di utilizzo intenso, senza degrado significativo delle performance.

5.1.3 Tipologia di test eseguiti

Test unitari

I test unitari hanno costituito il primo livello di verifica del sistema, concentrandosi sul corretto funzionamento dei singoli componenti del relay server. Per garantire che ogni componente fosse testato in isolamento, sono stati mockati i vari servizi esterni e le dipendenze, permettendo di simulare il comportamento delle interazioni esterne senza richiedere una connessione reale. Questo ha consentito di verificare il comportamento dei singoli moduli senza influenze da parte di altri elementi del sistema.

Tra i principali test unitari eseguiti, si possono citare quelli relativi alla formattazione delle risposte, un aspetto critico per garantire la chiarezza e la struttura delle informazioni fornite agli utenti. Inoltre, sono stati eseguiti test specifici per la generazione e validazione dei token di autenticazione, fondamentali per assicurare che solo gli utenti autorizzati possano accedere ai documenti riservati. Un altro gruppo di test ha riguardato la gestione della comunicazione con Microsoft Copilot Studio, in questo contesto, è stato verificato il tipo di scenario prima di effettuare la chiamata alle API, le quali ovviamente sono state mockate, e successivamente sono stati controllati i codici di risposta restituiti, per assicurarsi che il server gestisse correttamente ogni possibile esito, incluso il trattamento di errori.

Un ulteriore aspetto essenziale è stato la verifica della capacità del sistema di gestire scenari di fallimento. Sono stati simulati casi in cui si verificavano errori, come la mancata ricezione delle risposte dai servizi esterni o problemi nel parsing delle risposte stesse. Questi test hanno permesso di valutare come il sistema affrontasse situazioni critiche, implementando meccanismi di fallback come il retry automatico o la segnalazione di errori all'utente.

L'uso dei mock ha consentito di eseguire i test in un ambiente controllato, esplorando non solo le condizioni normali di funzionamento, ma anche le eccezioni e gli scenari limite. Questo ha permesso di identificare tempestivamente errori logici nel codice e di correggerli prima di passare ai test di integrazione.

Test di integrazione

I test di integrazione si sono concentrati sulla verifica della corretta interazione tra i vari componenti del sistema, distribuiti su piattaforme differenti. Dopo la pubblicazione del chatbot su Azure, i test sono stati eseguiti sull'URL del relay server per assicurarsi che tutti i passaggi infrastrutturali operassero in modo coordinato, includendo la gestione di processi complessi come l'autenticazione degli utenti, l'invio e ricezione delle richieste e l'accesso ai documenti su SharePoint.

Il primo aspetto verificato è stata la corretta creazione e gestione dei token di autenticazione, necessari per garantire che il relay server potesse autenticarsi correttamente al chatbot e gestire le comunicazioni in modo sicuro. I test hanno simulato diverse condizioni per verificare che il token venisse generato, validato e scadesse correttamente, garantendo la sicurezza del sistema anche in caso di tentativi non autorizzati di accesso.

Successivamente, è stata verificata l'interazione tra i moduli del sistema, dall'invio delle richieste al server da parte del chatbot fino al recupero e alla formattazione delle risposte provenienti dai documenti archiviati su SharePoint. Questi test hanno assicurato

che il sistema gestisse correttamente tutte le fasi della richiesta, incluse risposte complesse o di grande lunghezza.

Per simulare situazioni di carico intensivo e testare la robustezza del sistema, sono stati generati cinque thread che inviavano richieste simultaneamente. Questo ha permesso di valutare la capacità del chatbot di gestire più richieste in parallelo, un aspetto cruciale in contesti aziendali con interazioni multi-utente. Questa porzione di test è stata particolarmente importante per valutare l'idoneità del chatbot in ambienti aziendali, dove è comune che più utenti interagiscano con il sistema contemporaneamente.

Infine, un ulteriore aspetto rilevante è stato il monitoraggio dei tempi di risposta, sia per query semplici, che per richieste complesse. I tempi medi di elaborazione sono stati misurati per diverse tipologie di documenti, con attenzione alla latenza della rete e alla capacità di gestire documenti lunghi o specifici. Questo ha permesso di identificare eventuali colli di bottiglia e di ottimizzare le prestazioni del chatbot.

Test end-to-end

I test end-to-end sono stati eseguiti per valutare l'intero flusso di lavoro del chatbot, dalla richiesta dell'utente alla ricezione della risposta, simulando un ambiente reale. Questi test sono stati condotti su due piattaforme di comunicazione: WhatsApp e Microsoft Teams, poiché rappresentano i due canali attraverso cui gli utenti aziendali potrebbero interagire con il sistema. Selenium è stato utilizzato per automatizzare e simulare con precisione le interazioni degli utenti.

L'obiettivo principale era verificare che il chatbot fosse in grado di fornire risposte corrette e tempestive, mantenendo un'esperienza utente coerente su entrambe le piattaforme. I test hanno incluso scenari che variavano da richieste semplici, come l'accesso a documenti specifici, a query più complesse che richiedevano l'elaborazione di dati più articolati.

Un altro aspetto cruciale dei test era la gestione degli accessi per utenti con diversi livelli di autorizzazione. Sono stati simulati tentativi di accesso non autorizzato per garantire che il sistema limitasse correttamente l'accesso ai dati riservati, rispettando le politiche di sicurezza aziendali. I test hanno anche monitorato i tempi di risposta, valutando la capacità del sistema di gestire richieste di diversa complessità e di rispondere contemporaneamente su più piattaforme senza compromettere le prestazioni.

Infine, sono stati verificati i meccanismi di fallback e la gestione degli errori, simulando problemi come la mancata connessione ai servizi esterni o l'indisponibilità delle informazioni richieste. Questi test hanno permesso di garantire che il sistema potesse gestire correttamente le anomalie, fornendo messaggi di errore chiari e attuando soluzioni adeguate.

5.2 Risultati dei Test

Dopo aver definito l'ambiente di test, le metriche valutate e la metodologia seguita, in questa sezione vengono analizzati i risultati ottenuti durante le diverse fasi di test. L'analisi si concentrerà sull'efficacia del chatbot nel rispondere alle esigenze definite, con particolare attenzione alla sua capacità di fornire risposte accurate, gestire in modo

efficiente le richieste degli utenti e mantenere prestazioni elevate anche in situazioni di carico intensivo. I dati raccolti durante i test permetteranno di comprendere in che misura il sistema ha soddisfatto gli obiettivi prefissati e di individuare eventuali aree di miglioramento.

5.2.1 Precisione delle risposte

Uno degli aspetti fondamentali valutati durante i test è stata la precisione delle risposte. Il chatbot ha dimostrato una buona capacità di comprendere le query degli utenti e di fornire risposte pertinenti e coerenti con il contenuto dei documenti presenti su SharePoint. In scenari semplici, come la richiesta di documenti di facile accesso e con contenuto lineare, la percentuale di risposte esatte si è attestata intorno al 97%. In questi casi, il chatbot ha saputo individuare correttamente le informazioni rilevanti e presentarle in modo chiaro e conciso. In scenari più complessi, dove le query riguardavano documenti con strutture più articolate o contenuti tecnici, la precisione delle risposte è leggermente diminuita, attestandosi intorno al 90%. Questo risultato è stato influenzato principalmente dalla complessità delle informazioni da elaborare e dall'eterogeneità dei dati presenti nei documenti, in particolare, il chatbot ha mostrato maggiori difficoltà nell'elaborare correttamente documenti in formato Excel, dove la struttura a tabellare e la natura dei dati organizzati in celle hanno reso più complesso il recupero delle informazioni rilevanti. Tuttavia, il chatbot ha mantenuto un buon livello di accuratezza, dimostrando di essere in grado di gestire richieste complesse con discreta efficacia, nonostante le sfide presentate da documenti più strutturati come i fogli di calcolo.

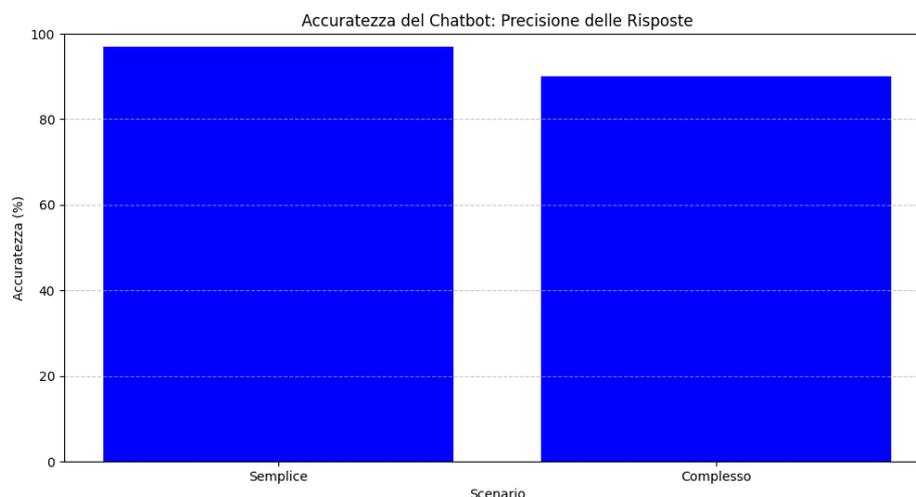


Figura 5.1: Grafico della Precisione delle risposte

Un altro aspetto cruciale valutato è stata la gestione degli accessi. Il chatbot ha correttamente limitato l'accesso alle informazioni confidenziali in base ai privilegi dell'utente.

In nessun caso sono state fornite informazioni sensibili a utenti senza i permessi necessari o semplicemente non autenticati. Questo risultato sottolinea la capacità del chatbot di rispettare le politiche di sicurezza e di limitare l'accesso alle informazioni riservate, garantendo così sia la precisione delle risposte che la protezione dei dati sensibili.

5.2.2 Velocità di risposta

La velocità di risposta del chatbot è stata monitorata attentamente durante i test, con particolare attenzione ai tempi medi di elaborazione in scenari con livelli di complessità variabili. I risultati mostrano una chiara differenza nei tempi di risposta a seconda del canale utilizzato e della complessità delle query.

Su WhatsApp, il tempo medio di risposta si è attestato intorno agli 11 secondi, sia per query di bassa complessità, come richieste di documenti brevi e ben strutturati, sia per richieste più articolate. Per query più complesse, come documenti più lunghi o contenenti dati tecnici, il tempo di risposta è leggermente aumentato, ma non ha mai superato i 15 secondi, rimanendo comunque entro i limiti accettabili per un'esperienza utente fluida. Questo ritardo è stato in parte condizionato dalla logica implementata nel relay server, necessario per gestire l'integrazione con WhatsApp. Il relay server, infatti, si occupa di ottenere il token, creare la conversazione e formattare la risposta, introducendo una latenza aggiuntiva rispetto all'integrazione più diretta con altre piattaforme.

Su Microsoft Teams, invece, i tempi di risposta sono stati generalmente inferiori. In media, il chatbot ha impiegato tra i 7 e gli 8 secondi per fornire una risposta, indipendentemente dalla complessità della query. Anche per le richieste più complesse, il tempo di risposta non ha mai superato i 12 secondi, evidenziando una maggiore efficienza su questa piattaforma, grazie all'integrazione più diretta con Microsoft Copilot Studio, che riduce la latenza rispetto al sistema relay.

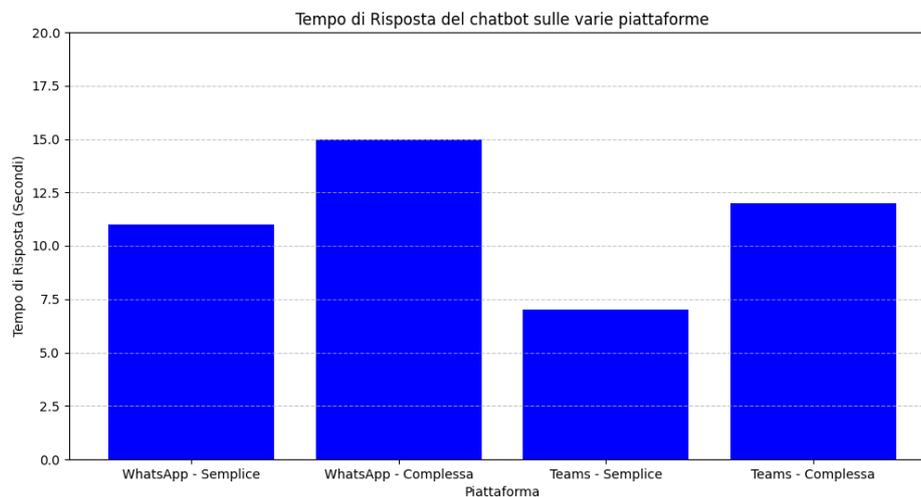


Figura 5.2: Grafico della velocità di Risposta

In generale, i risultati ottenuti rispettano i criteri di velocità stabiliti inizialmente, confermando che il chatbot è in grado di rispondere in tempi competitivi anche in scenari di maggiore complessità. La differenza tra i canali di comunicazione è stata un fattore rilevante, ma in entrambi i casi le prestazioni sono rimaste entro limiti accettabili per l'uso aziendale.

5.2.3 Scalabilità

La scalabilità del chatbot è stata testata attraverso simulazioni di carico per verificare la capacità del sistema di gestire un numero crescente di richieste e utenti simultanei. Il sistema è stato sottoposto a un test di carico con cinque thread paralleli, ciascuno dei quali inviava richieste simultaneamente. Questo test ha avuto lo scopo di replicare un contesto aziendale in cui più utenti interagiscono con il chatbot contemporaneamente.

I risultati hanno mostrato che il chatbot è riuscito a gestire correttamente tutte le richieste simultanee, mantenendo tempi di risposta stabili e senza alcun degrado significativo delle prestazioni. Anche sotto carico, il sistema ha mantenuto un tempo medio di risposta di 11 secondi su WhatsApp e di 7-8 secondi su Microsoft Teams, dimostrando la capacità di mantenere prestazioni consistenti anche in situazioni di utilizzo intensivo.

Un ulteriore aspetto importante valutato durante i test di scalabilità è stata la gestione delle conversazioni multiple. Il chatbot è stato in grado di gestire correttamente le conversazioni provenienti da utenti diversi, mantenendo il contesto di ogni interazione e fornendo risposte pertinenti a ciascun utente. Non sono stati riscontrati problemi nella gestione dei thread paralleli, e le risposte non hanno subito errori dovuti alla sovrapposizione di conversazioni.

Questi risultati indicano che il sistema è progettato per supportare un'espansione progressiva sia in termini di utenti che di carico di lavoro, confermando che il chatbot può scalare senza necessità di rivedere l'architettura di base o di affrontare costi elevati di implementazione.

5.3 Valutazione Economica e Competitività Commerciale

Valutazione dei costi

Dal punto di vista economico, la soluzione proposta per il chatbot si è rivelata vantaggiosa, grazie a un'architettura attentamente bilanciata tra costi e funzionalità offerte. Come discusso in precedenza, l'architettura ibrida ha evitato un aumento significativo dei costi, mantenendo la spesa iniziale accessibile grazie all'uso mirato dei servizi Microsoft. Un elemento chiave è stato l'integrazione di Microsoft Copilot Studio, il cui abbonamento mensile copre l'intera azienda a un costo di 200 dollari per tenant, indipendentemente dal numero di utenti attivi. Questo si è dimostrato un vantaggio competitivo rispetto ad altre soluzioni, come Power Automate, che richiederebbero licenze per ogni utente. In questo modo, il sistema ha offerto un'opzione scalabile senza costi aggiuntivi per il numero di licenze utilizzate. In termini di messaggistica, l'abbonamento a Microsoft Copilot Studio

include un pacchetto di 25.000 messaggi mensili, sufficiente per la maggior parte delle operazioni aziendali quotidiane. Se necessario, è possibile acquistare capacità aggiuntiva a un costo incrementale di 200 dollari per 25.000 messaggi ulteriori, permettendo una flessibilità economica e l'adattamento alle esigenze crescenti dell'organizzazione [32].

L'Azure AI Search, fondamentale per l'indicizzazione e la ricerca semantica nei documenti aziendali, comporta costi aggiuntivi, ma ben giustificati dalla qualità e velocità delle ricerche offerte. Il servizio di ricerca semantica, che può essere abilitato per un numero superiore a 1.000 richieste mensili, costa appena 1 dollaro per ogni 1.000 richieste aggiuntive. Questo costo relativamente contenuto rappresenta un ottimo rapporto qualità-prezzo, soprattutto in un contesto aziendale che richiede precisione e rapidità nell'accesso alle informazioni [33].

Inoltre, la modularità dell'architettura e l'uso di Power Automate, incluso nell'abbonamento a Microsoft Copilot Studio, consente di gestire i flussi di lavoro automatizzati senza ulteriori costi operativi. La gestione efficiente delle risorse attraverso componenti specializzati riduce al minimo i colli di bottiglia e ottimizza l'intero flusso di lavoro aziendale.

Competitività sul mercato

Dal punto di vista della competitività commerciale, il chatbot sviluppato si distingue per diverse caratteristiche chiave che lo rendono una soluzione appetibile rispetto alle alternative disponibili sul mercato. La prima, e più evidente, è la facilità d'uso offerta dall'approccio low-code: grazie all'integrazione di tecnologie come Power Automate e Copilot Studio, il sistema può essere gestito e configurato con facilità anche da utenti non tecnici. Questo riduce la necessità di un intervento esterno o di risorse IT specializzate per mantenere e aggiornare il sistema, abbassando così i costi di manutenzione e formazione del personale.

Un altro punto di forza significativo è la scalabilità del sistema: grazie all'architettura modulare, il chatbot è in grado di adattarsi facilmente a un incremento del numero di utenti o richieste senza richiedere modifiche sostanziali alla sua infrastruttura. Ciò garantisce una crescita graduale e sostenibile, anche in contesti aziendali in rapida espansione, senza la necessità di affrontare costi elevati di implementazione o di rivedere l'intera architettura del sistema. Questo aspetto è particolarmente importante in un mercato competitivo, dove la flessibilità e la capacità di adattamento sono cruciali per il successo a lungo termine.

Infine, il chatbot si presenta come una soluzione economicamente sostenibile grazie alla sua combinazione di costi fissi prevedibili e opzioni scalabili. Questo lo rende competitivo rispetto a soluzioni alternative più rigide o costose, specialmente in contesti aziendali di medie e grandi dimensioni, dove la necessità di automatizzare processi su larga scala può comportare costi molto più elevati con soluzioni meno flessibili.

5.4 Sicurezza e Conformità alla Normativa

La sicurezza dei dati è stata una priorità durante l'implementazione del chatbot aziendale, progettato per garantire che solo gli utenti autorizzati potessero accedere alle informazioni

riservate. Sebbene il sistema sia pensato per interrogare documenti aziendali generici, è stato comunque sviluppato tenendo in considerazione la possibilità che alcuni di essi contengano dati personali. Pertanto, il trattamento dei dati confidenziali è stato progettato in modo conforme alle politiche aziendali di sicurezza e, quando applicabile, alle normative sulla privacy come il GDPR [52].

Controllo degli Accessi e Protezione dei Dati

Uno dei principali meccanismi di sicurezza implementati riguarda il controllo degli accessi ai documenti indicizzati su SharePoint tramite Azure AI Search. Come discusso nella sezione relativa alla configurazione dell'indicizzazione, è stato adottato un approccio di least privilege, che limita l'accesso dell'applicazione ai soli documenti necessari per l'indicizzazione. In particolare, l'assegnazione dei permessi è stata gestita attraverso l'utilizzo di Microsoft Entra ID, con l'abilitazione di permessi specifici come Sites.Read.All e Files.Read.All. Questo garantisce che l'applicazione possa accedere esclusivamente ai file e alle directory pertinenti per ciascun utente autenticato, minimizzando così i rischi di esposizione non autorizzata dei dati.

Inoltre, il sistema utilizza token di autenticazione generati dinamicamente per ogni sessione di indicizzazione o di interrogazione da parte del chatbot. Questi token vengono creati e gestiti attraverso un client ID e un client secret, che vengono utilizzati per autenticare in modo sicuro ogni richiesta. L'adozione di questi meccanismi di autenticazione ha assicurato che solo le entità debitamente autorizzate potessero accedere ai dati aziendali, mantenendo un controllo rigoroso su chi può interagire con tali dati. Se i documenti interrogati contenessero dati personali, il sistema rispetterebbe i principi del GDPR, assicurando che solo chi ne ha diritto possa accedervi e che tali dati siano protetti da accessi non autorizzati [52].

Trattamento dei Dati Confidenziali

Per quanto riguarda la gestione delle informazioni confidenziali, il sistema garantisce che tutte le interazioni con il chatbot siano trattate in maniera sicura, sebbene la maggior parte dei documenti trattati non contenga necessariamente dati personali, le interazioni sono state progettate per essere conformi ai principi di protezione dei dati qualora tali informazioni fossero presenti. In particolare, i dati relativi agli utenti e alle loro interazioni non vengono memorizzati o utilizzati per scopi non autorizzati. Questo significa che le conversazioni gestite dal chatbot non vengono utilizzate per addestrare ulteriori modelli di intelligenza artificiale o per scopi di marketing, ma vengono trattate esclusivamente per fornire risposte contestuali e precise agli utenti in tempo reale. Tale approccio garantisce la minimizzazione dei dati, trattando solo le informazioni strettamente necessarie.

Inoltre, tutte le operazioni effettuate dal chatbot sono completamente tracciabili grazie ai log sistema generati durante l'accesso e l'uso dei documenti indicizzati. Questo sistema di monitoraggio consente una revisione continua delle attività del chatbot, offrendo la possibilità di intervenire rapidamente in caso di anomalie o violazioni della sicurezza. Questo approccio assicura una protezione dei dati confidenziali, mentre, nel caso di dati personali, supporta la conformità alle normative come il GDPR, dimostrando l'aderenza ai requisiti di tracciabilità e protezione dei dati.

Valutazione della Conformità

Durante l'implementazione del sistema, è stata condotta una valutazione preliminare della conformità del chatbot alle normative sulla privacy, con particolare attenzione ai requisiti del GDPR, nel caso in cui i documenti contenessero dati personali. L'analisi ha verificato che le interazioni degli utenti e i documenti trattati fossero gestiti in modo sicuro e trasparente, e i test condotti hanno confermato che il chatbot soddisfa i requisiti fondamentali di *privacy by default* e *privacy by design*, due principi chiave del GDPR che assicurano la protezione dei dati fin dalle prime fasi della progettazione del sistema [52].

In conclusione, l'adozione di rigorosi meccanismi di controllo degli accessi, l'uso di token di autenticazione dinamici e la tracciabilità delle operazioni assicurano che il sistema possa operare in un contesto aziendale conforme e sicuro. Sebbene la maggior parte dei documenti trattati non contenga dati personali, il sistema è comunque progettato per garantire la protezione di tali informazioni qualora fossero presenti, rispettando le normative in materia di privacy e sicurezza.

5.5 Miglioramenti Futuri

A seguito della fase di validazione, il chatbot ha dimostrato di essere uno strumento efficace per interrogare documenti confidenziali archiviati su SharePoint. Tuttavia, alcune limitazioni emerse durante i test suggeriscono spunti per possibili miglioramenti futuri: tali aspetti offrono l'opportunità di riflettere su come il sistema possa essere ulteriormente perfezionato e ampliato, integrando nuove funzionalità e migliorando le prestazioni in scenari più complessi.

Limitazioni Attuali

Uno degli aspetti principali emersi durante i test è legato alla scalabilità del sistema. Sebbene il chatbot abbia dimostrato in fase di test di gestire efficacemente cinque thread paralleli senza degradazioni significative delle prestazioni, l'aumento del carico in un contesto aziendale molto ampio potrebbe portare a rallentamenti. In aziende di grandi dimensioni, con molti dipendenti attivi che interrogano simultaneamente i documenti su SharePoint, il sistema potrebbe subire un rallentamento dovuto all'aumento del numero di richieste simultanee. Questa potenziale limitazione si manifesta soprattutto nel caso dell'integrazione con WhatsApp, dove la comunicazione attraverso il relay server aggiunge un ulteriore strato di latenza.

Un'ulteriore limitazione emersa riguarda la gestione delle query complesse, soprattutto nei documenti Excel: il sistema ha mostrato una minore precisione nell'elaborazione di fogli tabellari rispetto a formati come PDF o Word. La difficoltà deriva dall'eterogeneità dei campi presenti in questi documenti, i quali non sempre sono facilmente indicizzabili o interpretabili dal chatbot. Questa eterogeneità rende complesso estrarre informazioni strutturate e rispondere in modo accurato a query che richiedono una comprensione approfondita del contenuto dei fogli di calcolo.

Miglioramenti Futuri

Per affrontare le sfide legate alla scalabilità, un possibile miglioramento potrebbe consistere nel riprogettare il relay server adottando un approccio più moderno e scalabile, come un approccio reattivo. Questo tipo di architettura, basata sui principi di Reactive Programming, consentirebbe al server di gestire in modo più efficiente le richieste concorrenti, migliorando la risposta in scenari di carico particolarmente elevato. In un approccio reattivo, il sistema potrebbe rispondere in modo asincrono a un numero maggiore di richieste senza sovraccaricare il server, riducendo la latenza e migliorando la gestione delle risorse in ambienti con molti utenti attivi. Questo approccio sarebbe particolarmente utile in contesti dove l'interazione con il chatbot avviene tramite piattaforme di messaggistica esterna come WhatsApp, che già introduce una certa latenza nel flusso di comunicazione.

Per quanto riguarda, invece, e query più complesse, in particolare quelle relative ai documenti Excel, una possibile soluzione potrebbe consistere nel migliorare il processo di indicizzazione di questi documenti. Attualmente, la struttura eterogenea dei file Excel rappresenta una sfida poiché i campi possono variare notevolmente in base all'uso specifico del documento. Per risolvere questo problema, si potrebbe implementare una strategia di indicizzazione più granulare che tenga conto delle caratteristiche specifiche dei fogli di calcolo. Ciò potrebbe includere la definizione di regole personalizzate per l'indicizzazione dei campi più comuni, come numeri, date e tabelle, permettendo al chatbot di interpretare meglio i contenuti e rispondere con maggiore precisione. Tuttavia, data la diversità dei documenti Excel, questa strategia richiederebbe una fase di analisi approfondita per identificare le tipologie di campi più rilevanti e uniformare il processo di indicizzazione.

Espansione del Sistema

Un'importante opportunità di evoluzione per il chatbot risiede nell'espansione delle sue funzionalità oltre l'interrogazione dei documenti. Attualmente, il sistema è progettato principalmente per rispondere a query basate su documenti archiviati su SharePoint, ma potrebbe essere ulteriormente ampliato per supportare un'attività di gestione documentale completa. Ad esempio, il chatbot potrebbe essere integrato per facilitare l'inserimento di nuovi documenti, la modifica di quelli esistenti o la classificazione automatica dei file in base ai contenuti. Questo tipo di automazione sarebbe particolarmente utile in ambienti aziendali in cui il volume di documenti cresce rapidamente e dove l'ordinamento manuale potrebbe comportare errori o inefficienze. L'integrazione di queste funzionalità trasformerebbe il chatbot da un semplice strumento di interrogazione a una piattaforma di gestione documentale dinamica e interattiva, offrendo un controllo più granulare sui flussi documentali aziendali.

Un'altra potenziale area di espansione riguarda l'integrazione del chatbot con nuovi servizi aziendali, come sistemi di Customer Relationship Management (CRM) e Enterprise Resource Planning (ERP). Integrando il chatbot con un sistema CRM, come Microsoft Dynamics 365, sarebbe possibile fornire risposte personalizzate basate sui dati dei clienti, migliorando il supporto clienti e ottimizzando la gestione delle relazioni commerciali: ad esempio, il chatbot potrebbe recuperare dettagli su interazioni passate, ordini e preferenze, aiutando il personale a rispondere in modo più efficace alle richieste. L'integrazione con sistemi ERP, come SAP, permetterebbe al chatbot di assistere in attività più complesse, come la gestione delle risorse, il monitoraggio degli ordini o la verifica della disponibilità

di prodotti. Questo amplierebbe l'utilizzo del chatbot, rendendolo un punto di accesso centrale per le operazioni aziendali quotidiane, e riducendo la necessità di interventi manuali.

Infine, il sistema potrebbe evolvere con l'introduzione di capacità predittive e di analisi avanzata. Integrando strumenti di data analytics e modelli di intelligenza artificiale predittiva, il chatbot potrebbe analizzare tendenze storiche e offrire previsioni, ad esempio sulla domanda di prodotti o sull'andamento delle vendite. Queste funzionalità fornirebbero un valore aggiunto significativo, trasformando il chatbot in uno strumento proattivo, capace non solo di rispondere, ma anche di supportare le decisioni aziendali basate su dati e analisi.

Capitolo 6

Conclusione

L'elaborato ha esplorato la progettazione, implementazione e valutazione di un chatbot aziendale per l'interrogazione di documenti confidenziali archiviati su SharePoint, utilizzando un approccio low-code integrato con tecnologie Microsoft, come Power Platform, Azure AI Search e Azure OpenAI. L'obiettivo principale di questa tesi è stato quello di creare una soluzione efficace e sicura, in grado di migliorare l'accesso alle informazioni aziendali e ottimizzare i processi operativi, riducendo i tempi necessari per il recupero di informazioni da parte degli utenti.

I risultati ottenuti hanno dimostrato che un approccio low-code non solo semplifica il processo di sviluppo, ma rende anche più accessibili tecnologie avanzate come l'intelligenza artificiale e i servizi di ricerca semantica. Il chatbot ha saputo rispondere a richieste complesse con elevata precisione, mostrando notevoli capacità di comprensione linguistica grazie al modello GPT-4, e ha gestito efficacemente documenti aziendali in formati diversificati. L'integrazione di Microsoft Copilot Studio con Azure AI Search ha reso possibile un sistema di ricerca più intelligente e contestuale, migliorando notevolmente la pertinenza delle risposte rispetto ai tradizionali metodi di ricerca basati su parole chiave.

Dal punto di vista della sicurezza, l'implementazione di rigorosi meccanismi di autenticazione, combinati con il controllo degli accessi tramite Microsoft Entra ID e Dataverse, ha garantito la protezione dei dati. Il sistema ha operato in modo conforme alle normative sulla privacy, tra cui il GDPR, assicurandosi che solo gli utenti autorizzati potessero accedere a informazioni riservate. La tracciabilità delle operazioni e la gestione delle sessioni utente hanno, inoltre, contribuito a consolidare la robustezza della soluzione, rendendola idonea per ambienti aziendali in cui la sicurezza e la conformità alle normative sono di primaria importanza.

Nonostante i risultati soddisfacenti, i test hanno evidenziato alcune limitazioni, specialmente in contesti di elevata complessità. In particolare, la gestione di file Excel e la latenza introdotta dall'integrazione con piattaforme esterne, come WhatsApp, attraverso un relay server rappresentano sfide che possono essere affrontate con futuri miglioramenti. In un contesto aziendale di grandi dimensioni, l'aumento del numero di utenti e di richieste simultanee potrebbe richiedere un ripensamento dell'architettura del sistema, introducendo soluzioni più scalabili e reattive per garantire prestazioni ottimali anche

sotto carico elevato. I miglioramenti futuri potrebbero includere l'adozione di un'architettura reattiva per la gestione delle richieste concorrenti, nonché una più sofisticata indicizzazione dei documenti complessi, come i fogli di calcolo, che consentirebbe una maggiore accuratezza nella ricerca delle informazioni. Inoltre, vi è un enorme potenziale per l'espansione delle funzionalità del chatbot, che potrebbe evolvere fino a diventare uno strumento di gestione documentale integrato, capace non solo di rispondere a query, ma anche di gestire operazioni più complesse, come l'inserimento e l'ordinamento automatico dei file.

Dal punto di vista commerciale, il sistema proposto si distingue per la sua economicità e scalabilità. L'integrazione di servizi come Microsoft Copilot Studio e Azure AI Search, unitamente all'approccio low-code, consente di ridurre notevolmente i costi di sviluppo e manutenzione, rendendo il sistema accessibile anche ad aziende di medie dimensioni. Il potenziale per integrarsi facilmente con altre piattaforme aziendali, come CRM ed ERP, apre ulteriori opportunità di applicazione, rendendolo uno strumento che potrebbe diventare ancora più versatile e competitivo in un mercato in continua evoluzione.

In definitiva, la tesi ha dimostrato come un approccio low-code, combinate con tecnologie avanzate di intelligenza artificiale e gestione documentale, rappresenti una soluzione efficace e sostenibile per migliorare l'accesso e la gestione delle informazioni aziendali. Le prospettive future suggeriscono che il chatbot potrebbe essere ulteriormente potenziato e integrato con nuovi servizi, offrendo un valore aggiunto significativo per le imprese. Questa soluzione rappresenta, dunque, un importante passo verso l'ottimizzazione dei processi aziendali e la digitalizzazione delle attività, confermando l'importanza strategica dell'innovazione tecnologica nel contesto attuale.

Bibliografia

- [1] Gartner. *Top 10 Strategic Technology Trends for 2023*. 2023 (cit. a p. 1).
- [2] Andrew McAfee e Erik Brynjolfsson. «Big Data: The Management Revolution». In: *Harvard Business Review* (ott. 2012) (cit. a p. 1).
- [3] David J. Snowden e Mary E. Boone. «A Leader’s Framework for Decision Making». In: *Harvard Business Review* (nov. 2007) (cit. a p. 2).
- [4] Daniel E. O’Leary. «Artificial Intelligence and Big Data». In: *IEEE Intelligent Systems* 28.2 (mar. 2013), pp. 96–99 (cit. a p. 2).
- [5] Asbjørn Følstad e Petter Bae Brandtzaeg. «Chatbots and the New World of HCI». In: *Interactions* 24.4 (2017), pp. 38–42 (cit. alle pp. 3, 4).
- [6] Daniel Jurafsky e James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 3rd. Draft, August 2024. Prentice Hall, 2024 (cit. a p. 4).
- [7] Yajing Luo, Peng Liang, Chong Wang, Mojtaba Shahin e Jing Zhan. *Characteristics and Challenges of Low-Code Development: The Practitioners’ Perspective*. Accessed: September 2024 (cit. a p. 8).
- [8] Beth Stackpole. *Why companies are turning to ‘citizen developers’*. Accessed: August 2024. URL: <https://mitsloan.mit.edu/ideas-made-to-matter/why-companies-are-turning-to-citizen-developers> (cit. a p. 9).
- [9] George Orwell. *Nineteen Eighty-Four*. Penguin Classics, 2024 (cit. a p. 10).
- [10] *Introduction to Microsoft Power Platform for developers*. Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-platform/developer/get-started> (cit. alle pp. 10, 11).
- [11] Microsoft. *Microsoft Dataverse Developer Guide*. Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/developer/data-platform/overview> (cit. alle pp. 12, 13).
- [12] Microsoft. *What is Microsoft Dataverse?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-intro> (cit. a p. 13).

- [13] Microsoft. *What is Power Apps?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/powerapps-overview> (cit. alle pp. 13, 15).
- [14] Microsoft. *What are canvas apps?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/maker/canvas-apps/getting-started> (cit. a p. 14).
- [15] Microsoft. *What are model-driven apps in Power Apps?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/model-driven-app-overview> (cit. alle pp. 14, 16, 40).
- [16] Microsoft. *Understand model-driven app views.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/create-edit-views> (cit. a p. 14).
- [17] Microsoft. *Create and design model-driven app forms.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/create-design-forms> (cit. a p. 15).
- [18] Microsoft. *Get started with Power Automate.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-automate/getting-started> (cit. alle pp. 16, 18, 19).
- [19] Microsoft. *Overview of the different types of flows.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-automate/flow-types> (cit. alle pp. 17, 20).
- [20] Microsoft. *Microsoft Copilot Studio overview.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/fundamentals-what-is-copilot-studio> (cit. alle pp. 20, 21, 24).
- [21] Microsoft. *Orchestrate copilot topics and actions with generative AI (preview).* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/advanced-generative-actions> (cit. a p. 21).
- [22] Microsoft. *Create and edit topics.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/authoring-create-edit-topics?tabs=webApp> (cit. a p. 22).
- [23] Microsoft. *Use actions with custom copilots (preview).* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/advanced-plugin-actions> (cit. a p. 23).
- [24] Microsoft. *Key concepts – Analytics in Copilot Studio.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/analytics-overview> (cit. a p. 24).

- [25] Microsoft. *What's Azure AI Search?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/azure/search/search-what-is-azure-search> (cit. alle pp. 26, 27, 48).
- [26] Microsoft. *What is Azure OpenAI Service?* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/azure/ai-services/openai/overview> (cit. alle pp. 27, 47).
- [27] Microsoft. *Key concepts in Direct Line API 3.0.* Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/bot-service/rest-api/bot-framework-rest-direct-line-3-0-concepts?view=azure-bot-service-4.0> (cit. alle pp. 28, 61).
- [28] ShareGate. *The Benefits of a SharePoint Intranet.* Accessed: August 2024. 2023. URL: <https://sharegate.com/blog/benefits-sharepoint-intranet> (cit. a p. 28).
- [29] Microsoft. *Introduction to Microsoft Power Platform for developers.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-platform/developer/get-started> (cit. a p. 29).
- [30] Microsoft. *Configure user authentication with Microsoft Entra ID.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/configuration-authentication-azure-ad> (cit. alle pp. 33, 44).
- [31] Microsoft. *Use SharePoint content for generative answers.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/nlu-generative-answers-sharepoint-onedrive> (cit. a p. 35).
- [32] Microsoft. *Power Platform Licensing Guide.* Accessed: September 2024. Giu. 2024 (cit. alle pp. 36, 72).
- [33] Microsoft. *Azure AI Search pricing.* Accessed: August 2024. URL: <https://azure.microsoft.com/en-us/pricing/details/search/> (cit. alle pp. 36, 72).
- [34] Microsoft. *Contact table/entity reference.* Accessed: August 2024. URL: <https://learn.microsoft.com/en-us/power-apps/developer/data-platform/reference/entities/contact> (cit. a p. 41).
- [35] Microsoft. *Index data from SharePoint document libraries.* Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/search/search-howto-index-sharepoint-online> (cit. alle pp. 43, 44, 46, 47).
- [36] Microsoft. *Microsoft Graph permissions reference.* Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/graph/permissions-reference> (cit. a p. 43).

- [37] Microsoft. *Azure OpenAI On Your Data*. Accessed: September 2024. URL: <https://learn.microsoft.com/it-it/azure/ai-services/openai/concepts/use-your-data?tabs=ai-search%2Ccopilot> (cit. a p. 47).
- [38] Microsoft. *Azure OpenAI Service models*. Accessed: September 2024. URL: <https://learn.microsoft.com/it-it/azure/ai-services/openai/concepts/models> (cit. a p. 48).
- [39] Microsoft. *Full text search in Azure AI Search*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/search/search-lucene-query-architecture> (cit. a p. 49).
- [40] Microsoft. *Semantic ranking in Azure AI Search*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/search/semantic-search-overview> (cit. a p. 49).
- [41] Microsoft. *Add and manage knowledge for generative answers*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/nlu-boost-conversations> (cit. a p. 50).
- [42] Microsoft. *Azure OpenAI Service REST API reference*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/ai-services/openai/reference> (cit. alle pp. 51, 53).
- [43] Odata. *Basic Tutorial*. Accessed: September 2024. 2024. URL: <https://www.odata.org/getting-started/basic-tutorial/> (cit. a p. 54).
- [44] Microsoft. *Welcome to OData*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/odata/overview> (cit. a p. 54).
- [45] Microsoft. *OData \$filter syntax in Azure AI Search*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/azure/search/search-query-odata-filter> (cit. a p. 56).
- [46] Microsoft. *Key concepts - Publish and deploy your copilot*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/publication-fundamentals-publish-channels?tabs=web> (cit. a p. 57).
- [47] Microsoft. *Connect and configure a copilot for Microsoft Teams*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/publication-add-bot-to-microsoft-teams> (cit. a p. 58).
- [48] Microsoft. *Publish a copilot to mobile or custom apps*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/microsoft-copilot-studio/publication-connect-bot-to-custom-application> (cit. a p. 59).

- [49] Twilio. *What is Twilio? An introduction to the leading customer engagement platform*. Accessed: September 2024. URL: <https://www.twilio.com/en-us/resource-center/what-is-twilio-an-introduction-to-the-leading-customer-engagement-platform> (cit. a p. 59).
- [50] Twilio. *Programmable Messaging for WhatsApp and C#/.NET Quickstart*. Accessed: September 2024. URL: <https://www.twilio.com/docs/whatsapp/quickstart/csharp> (cit. alle pp. 59, 60).
- [51] Microsoft. *Publish an ASP.NET Core web API to Azure API Management with Visual Studio*. Accessed: September 2024. URL: <https://learn.microsoft.com/en-us/aspnet/core/tutorials/publish-to-azure-api-management-using-vs?view=aspnetcore-8.0&viewFallbackFrom=aspnetcore-8.0capitolo> (cit. alle pp. 63, 64).
- [52] European Parliament e Council. *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)*. Accessed: September 2024. 2016. URL: <https://eur-lex.europa.eu/eli/reg/2016/679/oj> (cit. alle pp. 73, 74).