

POLITECNICO DI TORINO

Corso di Laurea Magistrale in

Ingegneria Gestionale

Tesi di Laurea Magistrale



**Politecnico
di Torino**

Low-Code/No-Code e Digital Transformation:
caso di studio con Microsoft Power Platform

Relatore:

Prof. Luca Ardito

Laureando:

Emanuele Grandi

Anno accademico 2024/2025

Sommario

Il presente studio si propone di analizzare le tecnologie *low code/no code* e il loro potenziale impatto nel contesto aziendale, in termini di velocità di sviluppo, efficienza e accessibilità. La trattazione inizia con una panoramica sull'evoluzione delle metodologie di programmazione, dalle origini della programmazione strutturata fino alla programmazione ad oggetti e alle architetture a microservizi.

Questo excursus mostra come il progresso verso sistemi distribuiti, cloud platforms e soluzioni sempre più modulari e accessibili abbia portato al naturale sviluppo delle tecnologie *low code/no code*, strumenti progettati per semplificare la creazione di applicazioni complesse e accelerare la digitalizzazione dei processi aziendali.

Dopo aver preso in esame il mercato, le tendenze nel settore e le principali soluzioni *low code* attualmente disponibili, quali Mendix, Appian e Microsoft Power Platform, l'analisi si focalizza sull'applicazione di queste tecnologie attraverso un caso di studio pratico: la realizzazione di un tool per il monitoraggio dei progetti, sviluppato utilizzando Microsoft Power Platform (Power Apps, Power Automate e Power BI) e integrato con Azure DevOps.

Questo caso pratico consente di valutare come la piattaforma *low code* di Microsoft risponda alle esigenze aziendali, permettendo la costruzione rapida di soluzioni su misura senza la necessità di competenze avanzate di programmazione. L'integrazione con Azure DevOps fornisce, inoltre, funzionalità di gestione e monitoraggio del ciclo di vita del progetto, combinando l'approccio visivo delle tecnologie *low code* con le pratiche di sviluppo *DevOps*.

Nel corso della tesi viene approfondito il confronto tra tecnologie *low code/no code* e soluzioni customizzate tradizionali, esaminandone i punti di forza e le limitazioni. Tra i principali vantaggi si annoverano la velocità di implementazione, l'accessibilità

per utenti non tecnici e la disponibilità di connettori preconfigurati che semplificano l'integrazione con sistemi aziendali esistenti. Vengono presi in esame, tuttavia, anche i principali limiti, come i costi legati alle licenze, le restrizioni nelle possibilità di personalizzazione e la potenziale dipendenza dalla piattaforma scelta, che può rendere complessa l'esportazione delle soluzioni sviluppate.

Attraverso una riflessione approfondita sul caso di studio e una valutazione critica delle prospettive future, la tesi fornisce un quadro completo dei benefici e delle sfide legate all'adozione delle soluzioni *low code/no code* nel contesto aziendale, con particolare attenzione alle opportunità di integrazione con i sistemi informativi esistenti e con l'intelligenza artificiale.

Indice

Elenco delle figure	7
1 Tecnologie Low-Code/No-Code: Origini, Diffusione e Prospettive	11
1.1 La Naturale Evoluzione verso il Low-Code/No-Code	11
1.1.1 Gli albori della Programmazione	12
1.1.2 La nascita dei linguaggi di alto livello	12
1.1.3 L'ascesa dei linguaggi moderni	12
1.1.4 L'avvento del web e i linguaggi di scripting	13
1.1.5 Le prime Librerie Visuali	13
1.1.6 Microservizi e Cloud	14
1.1.7 Low code/No Code	15
1.2 Blockchain e IA: Il Ruolo Centrale delle Tecnologie Low-Code nell'Innovazione Digitale	17
1.2.1 Blockchain, applicazioni e sfide	17
1.2.2 Blockchain e Low-Code	19
1.2.3 IA e Piattaforme Low Code: Democratizzazione dello Sviluppo Software e Rivoluzione del processo di Innovazione	20
1.2.4 Vantaggi e Sfide dell'Integrazione	21
1.2.5 Low-code/No-code AI	23
1.3 Il mercato delle Piattaforme Low Code	25
1.3.1 Le ragioni della rapida espansione	25

1.3.2	I fattori che potrebbero limitare l'espansione	28
1.3.3	Settori di applicazione e Panorama Competitivo	29
2	Panoramica delle Soluzioni Low-Code Attuali e l'Approccio DevOps	31
2.1	Introduzione	31
2.2	Mendix	32
2.2.1	Origini, Vision e Mission	32
2.2.2	L'evoluzione della Piattaforma	33
2.2.3	Struttura e Funzionalità Principali	34
2.3	Appian	41
2.3.1	Origini ed Evoluzione verso il Low Code	41
2.3.2	Funzionalità e Componenti Principali	42
2.4	Microsoft Power Platform	48
2.4.1	Introduzione	48
2.4.2	Power Automate	49
2.4.3	Power Apps	54
2.4.4	Power BI	59
2.5	Devops e Low-Code	65
3	Implementazione: Caso di Studio con Microsoft Power Platform	69
3.1	Scopo e Ambito del Progetto	69
3.2	Liste Microsoft Sharepoint: Entità Progetto ed Entità Risorsa	72
3.2.1	Entità Progetto	72
3.2.2	Entità Risorsa	75
3.3	Gestione Entità Progetto	76
3.3.1	Ciclo di vita dell'Entità	76
3.3.2	Inserimento di Una Nuova Proposta e Valutazione Fattibilità	80
3.3.3	Caricamento Offerta	87

3.3.4	Registrazione Esito Trattative	91
3.3.5	Gestione Progetti	104
3.4	Gestione Entità Risorsa	108
3.4.1	Censimento Risorse	108
3.5	Capacity Plan Preventivo	111
3.5.1	Estrazione Risorse Fittizie dal Calculation Model	111
3.5.2	Gantt e Gestione Risorse: Integrazione tra Power BI e Power Apps	115
4	Analisi dei Risultati Ottenuti	119
4.1	Introduzione	119
4.2	Vantaggi dell'approccio Low-Code	120
4.2.1	Velocità di Implementazione	120
4.2.2	Velocità di Apprendimento	124
4.2.3	Prototipazione e Digitalizzazione dei Processi	126
4.3	Limiti nell'utilizzo di Microsoft Power Platform	128
4.3.1	Customizzazioni Limitate ed Integrazione "Forzata"	128
4.3.2	Limiti funzionali dei connettori Power Automate	132
4.3.3	Dipendenza dall'Ecosistema Microsoft e dal Cloud	134
4.3.4	Costi di Licenza	136
4.4	Confronto tra Low-Code e Custom Development	143
5	Conclusioni	153
	Sitografia	156
	Bibliografia	161

Elenco delle figure

1.1	Interfaccia Visual Basic 6.0	14
1.2	Apple Create ML Templates	24
1.3	Revenues Tecnologie di Sviluppo Low Code	27
1.4	Settori di Applicazione Piattaforme Low Code	29
1.5	Key Companies del Settore	30
2.1	Ponte tra Business e IT (Mendix)	33
2.2	Visione di Mendix dello “spettro” degli sviluppatori	34
2.3	Best Practice Recommender (Maia)	37
2.4	Logic Recommender (Maia)	38
2.5	Workflow Recommender (Maia)	38
2.6	Maia Domain Model Generator	39
2.7	Atlas Design System	40
2.8	Integrazione Dati con Data Fabric	43
2.9	Funzionalità Drag-And-Drop Appian SAIL	44
2.10	Espression Editor	45
2.11	Appian Process Modeler Fonte: Appian	48
2.12	Panoramica Microsoft Power Platform	49
2.13	Scelta Trigger Cloud Flow	52
2.14	Compilazione parametri azione Power Automate	53
2.15	Modelli di Flusso Power Automate	53

2.16	Generazione Flusso Con Copilot	54
2.17	Interfaccia Drag-and-Drop Power Apps	55
2.18	Barra delle Formule e Proprietà	56
2.19	Power Automate Pane in Power Apps	57
2.20	Connessioni alle Fonti Dati Power BI	61
2.21	Power Query Editor	62
2.22	Report Power BI Fonte: Microsoft Learn	63
2.23	Ciclo di Vita Approccio DevOps	66
3.1	Fasi Gestione Progetti	70
3.2	Home Page Applicazione	79
3.3	Diagramma Processo Inserimento Nuova Proposta	81
3.4	Diagramma Processo Valutazione Fattibilità	82
3.5	Pagina “Opportunità” dell’Applicazione Power Apps	82
3.6	Proprietà <i>OnSelect</i> pulsante “Carica Nuova Opportunità”	84
3.7	Flusso Valutazione Fattibilità 1	84
3.8	Flusso Valutazione Fattibilità 2	85
3.9	Azione "Pubblica Messaggio su Canale Teams" per notificare rifiuto Opportunità	87
3.10	Diagramma Processo Caricamento Offerta	88
3.11	Pagina “Caricamento Offerta” dell’Applicazione Power Apps	88
3.12	Flusso Caricamento e Conferma Invio Offerta	90
3.13	Diagramma Processo Registrazione Esito Trattative	92
3.14	Pagina “Registra Esito Trattative” Step 1	93
3.15	Conferma Rifiuto Offerta	94
3.16	Caricamento Dettagli Offerta Finale	94
3.17	Flusso Aggiornamento Stato e creazione progetto DevOps	96
3.18	Aggiunta Progetto su <i>DevOps</i>	97
3.19	Corpo Richiesta HTTP per Aggiunta Progetto	99

3.20	Esempio azione “Analizza JSON”	100
3.21	Richiesta HTTP “Aggiunta Amministratore”	101
3.22	Creazione <i>Epic</i> per “Data Inizio Progetto”	103
3.23	Diagramma Processo Gestione Progetti	105
3.24	Pagina “Gestione Progetti”	105
3.25	Richiesta Approvazione per Modifica Stato Progetto	107
3.26	Pagina “Risorse”	109
3.27	Pagina “Gestisci Risorse”	109
3.28	Esempio Output Campi “Numero Progetti a carico” e “Progetti a carico”	111
3.29	Foglio “IA Progetti” Calculation Model	112
3.30	<i>Script</i> Excel	113
3.31	Richiesta HTTP a Sharepoint	114
3.32	Capacity Plan Preventivo su Power BI	116
3.33	Modifica Parametri Risorsa su Power BI	117
3.34	Flusso per l’Aggiornamento del Set di Dati in Power BI	118
4.1	ChatBot Copilot integrato in Power Automate	125
4.2	Tabella Riepilogativa Risorse-Macroattività	131
4.3	Ecosistema Microsoft Fonte: Yaveon	135
4.4	Confronto tra funzionalità in base al tipo di licenza	140
4.5	Stima Costo Licenze Progetto	142
4.6	Confronto CRM <i>custom</i> e CRM <i>Low-code/No-code</i>	151

Capitolo 1

Tecnologie Low-Code/No-Code: Origini, Diffusione e Prospettive

1.1 La Naturale Evoluzione verso il Low-Code/No-Code

Negli ultimi decenni, il mondo dello sviluppo software ha vissuto una trasformazione radicale. Partendo da processi complessi che richiedevano una conoscenza approfondita di linguaggi di programmazione tradizionali, si è giunti a tecnologie intuitive che permettono la creazione di applicazioni senza la necessità di scrivere codice. Questa evoluzione riflette il desiderio di semplificare lo sviluppo, rendendolo accessibile a un pubblico più ampio, con un background non necessariamente tecnico. Per comprendere appieno le tecnologie *low-code/no-code*, è essenziale ripercorrere le tappe principali che hanno portato a questa rivoluzione, descritte nell'articolo su Oraichain Academy del 2023 [1] "*The Brief History of Code Languages: Road to No-code*".

1.1.1 Gli albori della Programmazione

Inizialmente la programmazione era un'attività estremamente complessa, limitata a pochi esperti. I computer erano enormi macchine caratterizzate da capacità molto limitate e utilizzavano esclusivamente il linguaggio binario. Ogni istruzione doveva essere scritta manualmente, richiedendo un lavoro minuzioso e altamente tecnico. Questo approccio alla programmazione, pur fondamentale per avviare l'era dell'informatica, presentava evidenti limiti. La complessità e l'errore umano rendevano il processo lungo e complesso. La necessità di strumenti in grado di agevolare l'interazione tra l'utente e la macchina ha spinto i pionieri del settore a sviluppare linguaggi più intuitivi.

1.1.2 La nascita dei linguaggi di alto livello

Negli anni '50 e '60, i primi linguaggi di programmazione ad alto livello cambiarono radicalmente il panorama. Innovatori come Grace Hopper e John Backus crearono strumenti che semplificarono il lavoro dei programmatori. Hopper progettò il *FLOW-MATIC*, un linguaggio precursore del *COBOL*, pensato per applicazioni aziendali. Backus, invece, sviluppò *FORTRAN*, concepito per rispondere alle esigenze del calcolo scientifico e ingegneristico. Questi linguaggi introdussero una sintassi più vicina al linguaggio umano, consentendo di ridurre il divario tra le esigenze dei programmatori e le limitazioni delle macchine. Con l'avvento di linguaggi simbolici come l'assembly, si compì un primo passo verso una programmazione meno vincolata alla struttura rigida dell'hardware.

1.1.3 L'ascesa dei linguaggi moderni

Con il passare degli anni, il panorama si è ulteriormente evoluto. Il linguaggio C, sviluppato negli anni '70, rappresentò una pietra miliare grazie alla sua combinazione di potenza e flessibilità. Su questa base nacquero linguaggi come C++ e Java,

che introdussero nuovi paradigmi, come la programmazione orientata agli oggetti e la portabilità tra piattaforme. Queste innovazioni non solo semplificarono la creazione di software complesso ma democratizzarono l'accesso alla programmazione. La maggiore accessibilità dei linguaggi di alto livello pose le basi per una diffusione su larga scala dello sviluppo software, consentendo la realizzazione di applicazioni sempre più avanzate.

1.1.4 L'avvento del web e i linguaggi di scripting

Negli anni '90, con la crescita di internet, si affermarono linguaggi pensati per il web. JavaScript, originariamente concepito per migliorare l'interattività delle pagine web, divenne uno dei pilastri dello sviluppo front-end. Python, invece, si distinse per la sua semplicità e versatilità, rendendolo ideale per applicazioni web, analisi dei dati e molto altro. Questi linguaggi di scripting resero la programmazione più accessibile a persone con background diversi, aprendo la strada a nuove modalità di sviluppo e portando alla nascita del *Rapid Application Development (RAD)*, un approccio che privilegia la velocità e la flessibilità.

1.1.5 Le prime Librerie Visuali

La spinta verso la semplificazione culminò negli anni '80 e '90 con l'introduzione di strumenti visivi come *Visual Basic* e *Dreamweaver*. Pur richiedendo ancora una certa conoscenza di scripting, questi flussi di lavoro visivi anticipavano il potenziale per semplificare e automatizzare le attività di codifica manuale.

Questi strumenti permisero agli sviluppatori di costruire applicazioni e siti web manipolando elementi grafici invece di scrivere codice, anticipando di fatto le tecnologie *low-code* e *no-code*.

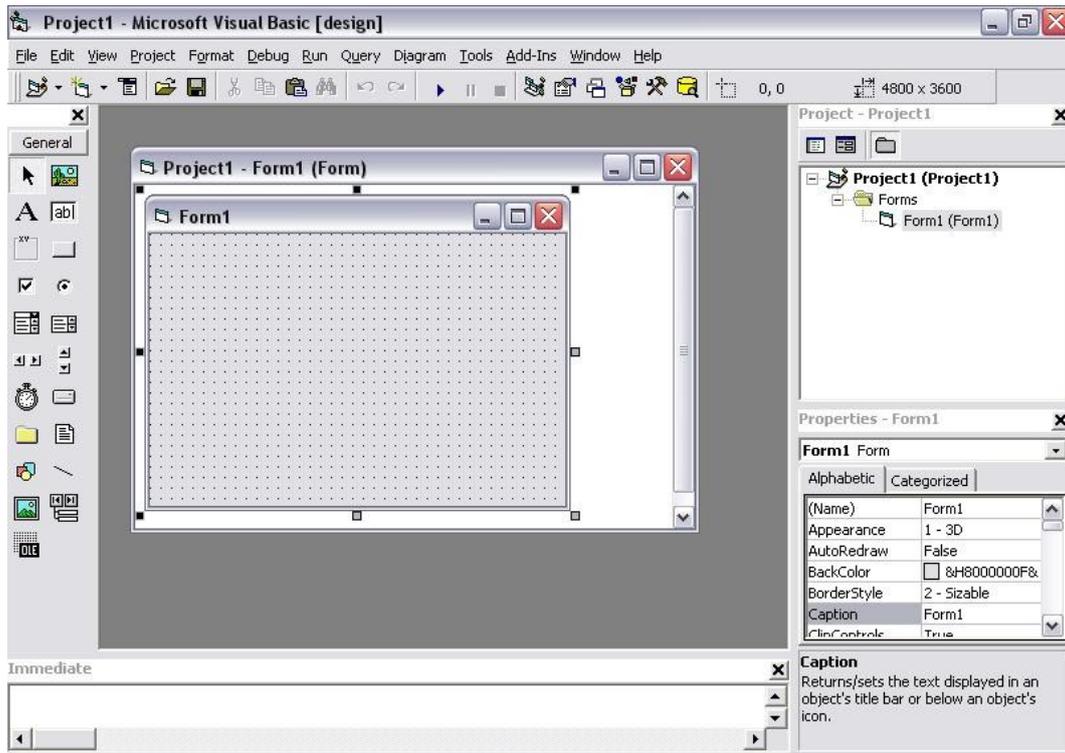


Figura 1.1. Interfaccia Visual Basic 6.0

1.1.6 Microservizi e Cloud

I microservizi rappresentano un approccio rivoluzionario nell'architettura del software, nato nel 2014 con le definizioni di Lewis e Fowler. Questo paradigma, progettato per affrontare le limitazioni delle applicazioni monolitiche, offre una struttura in cui i sistemi complessi sono suddivisi in servizi autonomi, piccoli e granulari. Ogni servizio è responsabile di una specifica funzione aziendale ed è sviluppato, distribuito e mantenuto in modo indipendente. Questa caratteristica consente di migliorare la scalabilità, la resilienza e l'agilità dei sistemi. Aziende come Netflix, Spotify e Amazon hanno dimostrato i vantaggi pratici dei microservizi, adottandoli per supportare sistemi globali ad alta intensità di traffico. Questo successo ha reso i microservizi uno standard per applicazioni *cloud-native*, permettendo aggiornamenti più rapidi e una migliore gestione delle risorse tecnologiche. L'evoluzione verso il

cloud e i microservizi ha rappresentato una svolta significativa nella progettazione e gestione delle applicazioni aziendali. La crescente complessità delle applicazioni, unita alla necessità di gestire carichi di lavoro distribuiti, ha incentivato l'adozione del cloud. Piattaforme come AWS e Microsoft Azure offrono la possibilità di scalare le risorse in modo flessibile, eliminando la dipendenza da hardware fisico e permettendo alle aziende di concentrarsi sulla vendita di servizi piuttosto che di infrastrutture. Questo approccio ha contribuito a trasformare il modello economico del software, spostandosi verso la "servitizzazione", ovvero la fornitura di servizi attraverso il cloud, come dimostrato dall'espansione dei servizi *SaaS* (Software as a Service). Parallelamente, l'adozione di architetture a microservizi ha semplificato lo sviluppo delle applicazioni, consentendo alle aziende di suddividere grandi sistemi in componenti più piccoli, autonomi e facilmente scalabili. I microservizi sono ideali per l'ambiente cloud, in quanto permettono alle applicazioni di evolversi rapidamente, riducendo i tempi di sviluppo e migliorando la resilienza del sistema, anche se la gestione e l'integrazione di numerosi microservizi può risultare complessa e richiede soluzioni avanzate di monitoraggio e gestione [1](MPDI, 2021).

1.1.7 Low code/No Code

Nel 2014, il termine "low-code" venne coniato dall'analista di Forrester Clay Richardson nel report [2] "*New Development Platforms Emerge For Customer-Facing Applications*", per descrivere piattaforme come Mendix e Appian, progettate per automatizzare parti del processo di programmazione.

Il passaggio al "no-code" avvenne poco dopo, con l'introduzione di nuovi strumenti che miravano a democratizzare ulteriormente lo sviluppo.

Mentre il *low-code* riduceva il bisogno di scrivere codice, il *no-code* lo eliminava del tutto in molti casi d'uso.

Questa tipologia di piattaforme, con interfacce intuitive e funzionalità *drag-and-drop*, consentono a utenti non tecnici di creare applicazioni complesse in modo

rapido ed efficiente, abbattendo le barriere tradizionali della programmazione.

Il *low-code* è progettato per semplificare e velocizzare lo sviluppo di applicazioni, utilizzando interfacce grafiche e automazioni al posto di lunghi processi di scrittura di codice. Risulta evidente, inoltre, la convergenza naturale tra microservizi e *low-code*: i microservizi offrono modularità e granularità, mentre le piattaforme *low-code* sfruttano queste caratteristiche per consentire agli sviluppatori di costruire applicazioni componibili e altamente scalabili senza la necessità di approfondite competenze tecniche.

Le piattaforme *low-code*, spesso basate su architetture di microservizi, permettono di integrare facilmente moduli già esistenti e di scalare le applicazioni in modo dinamico. Questo approccio è particolarmente utile per le aziende che vogliono accelerare il *time-to-market* o che dispongono di risorse limitate nel campo dello sviluppo software. Grazie ai microservizi, i team possono concentrarsi sull'innovazione e sugli obiettivi strategici, delegando alla piattaforma *low-code* il lavoro tecnico complesso, come la gestione dell'infrastruttura e delle comunicazioni tra servizi.

Questa sinergia rappresenta un'opportunità straordinaria per il futuro dello sviluppo software. I microservizi possono beneficiare della semplicità del *low-code* per ampliare la loro adozione, mentre le piattaforme *low-code* possono sfruttare la robustezza dei microservizi per offrire soluzioni più potenti e scalabili. Insieme, questi due approcci stanno ridefinendo il modo in cui le aziende progettano e implementano le loro applicazioni, rendendo lo sviluppo software più accessibile ed efficiente. Nonostante i vantaggi evidenti, ci sono sfide significative da affrontare. È cruciale garantire che l'integrazione tra microservizi e *low-code* non comprometta la separazione delle responsabilità o la scalabilità intrinseca dei microservizi. Inoltre, l'astrazione delle piattaforme *low-code* potrebbe rendere più difficile il monitoraggio e la risoluzione dei problemi a livello di sistema.

L'integrazione di tecnologie emergenti come blockchain e intelligenza artificiale

segna il prossimo capitolo di questa rivoluzione.

1.2 Blockchain e IA: Il Ruolo Centrale delle Tecnologie Low-Code nell’Innovazione Digitale

1.2.1 Blockchain, applicazioni e sfide

La tecnologia *blockchain*, conosciuta anche come *Distributed Ledger Technologies (DLT)*, è un sistema decentralizzato che archivia le transazioni in blocchi collegati tra loro, ciascuno dotato di un identificatore unico. Una volta registrati, i dati possono essere modificati solo con il consenso di tutti i partecipanti alla rete, rendendo necessario l’aggiornamento di tutti i blocchi successivi. Questo processo garantisce un elevato livello di sicurezza e integrità.

I principali vantaggi della blockchain includono:

- *Decentralizzazione*, che elimina la dipendenza da intermediari.
- *Immutabilità*, che protegge i dati da modifiche non autorizzate.
- *Trasparenza*, con la possibilità di tracciare ogni transazione visibile ai membri della rete.

Nonostante la tecnologia *blockchain* sia sostanzialmente ancora in una fase iniziale di sviluppo, il suo potenziale è già evidente attraverso numerose applicazioni in settori diversi. Tra gli usi più diffusi oggi troviamo le criptovalute, che rappresentano uno dei primi esempi di implementazione di questa tecnologia. La blockchain funge da registro digitale sicuro e decentralizzato, consentendo di tracciare e verificare tutte le transazioni legate alle criptovalute.

Un ulteriore campo di applicazione significativo è quello delle identità digitali. Grazie alla sua capacità di creare registri resistenti alle manomissioni, la *blockchain*

permette di sviluppare identità virtuali sicure, utili per validare informazioni personali e dati sensibili. Questo aspetto acquisirà ulteriore rilevanza con l'aumento della digitalizzazione dei nostri asset e delle informazioni personali.

Anche il settore elettorale e di votazione può trarre vantaggio dall'utilizzo della *blockchain*. Con un sistema che garantisce un registro decentralizzato e inalterabile dei voti espressi, questa tecnologia offre la possibilità di implementare processi di votazione più trasparenti e sicuri, riducendo il rischio di frodi elettorali e tutelando l'integrità del voto. Nella gestione della supply chain, la *blockchain* consente di tenere traccia di ogni operazione in un registro condiviso e trasparente. Ogni transazione viene registrata come un blocco, creando un archivio immutabile che aumenta la fiducia e l'efficienza di tutto il processo.

Infine, un'innovazione particolarmente promettente è quella degli *Smart Contract*, contratti digitali auto-esecutivi che si attivano automaticamente al verificarsi di condizioni predefinite. La *blockchain* garantisce un ambiente sicuro e decentralizzato per la loro creazione e gestione. Tra gli ambiti più interessanti per gli *smart contract* troviamo le applicazioni decentralizzate (dApp) e le organizzazioni decentralizzate (DAO), che potrebbero rivoluzionare il modo in cui vengono gestiti i contratti e le collaborazioni online.

Malgrado questi evidenti punti di forza, ci sono diversi ostacoli e barriere all'ingresso, che ne limitano l'adozione, tra cui i costi elevati, la necessità di competenze tecniche specializzate e la difficoltà di integrazione con altri sistemi.

La principale sfida deriva dalla complessità intrinseca della tecnologia: progettare nuove applicazioni che sfruttino la *blockchain* richiede una conoscenza approfondita in molteplici discipline, come la crittografia, la teoria dei giochi, l'informatica e il calcolo distribuito. Oltre a ciò, i progetti basati sulla *blockchain* non sono solo difficili da progettare dal punto di vista tecnico, ma richiedono anche una pianificazione accurata dei processi aziendali per garantire che le informazioni vengano trasmesse

in modo preciso ed efficace. Questo implica che persino modifiche semplici possano richiedere sforzi significativi da parte di più team per un lungo periodo di tempo.

1.2.2 Blockchain e Low-Code

Blockchain e *low-code* sono alleati naturali. Le piattaforme di sviluppo *low-code* ridefiniscono il modo in cui le applicazioni vengono progettate e distribuite, offrendo una velocità e un’agilità difficilmente paragonabili con i metodi tradizionali. Allo stesso tempo, la blockchain è emersa come una tecnologia fondamentale per garantire trasparenza e fiducia nelle interazioni digitali.

Sebbene la tecnologia *blockchain* e le piattaforme *low-code* siano già vantaggiose di per sé, la loro combinazione rappresenta uno strumento potente per aziende di diverso genere.

Utilizzando il potenziale della *blockchain* e implementando questa tecnologia nei propri processi, le imprese possono ridurre i costi, il tempo, lo sforzo e i rischi, sviluppando applicazioni *blockchain* sfruttando piattaforme *low-code*.

Queste piattaforme consentono alle aziende di creare e gestire applicazioni basate sulla *blockchain* con conoscenze limitate di programmazione, colmando il divario di risorse e competenze necessarie.

Tra i principali vantaggi nell’utilizzo di piattaforme *low-code* in applicazioni *blockchain* vi sono:

Flessibilità

Le piattaforme *low-code* permettono agli utenti di modificare facilmente un’applicazione già sviluppata. Questo è particolarmente utile nella *blockchain*, dove la tecnologia è in continua evoluzione.

Scalabilità

Le piattaforme *low-code* consentono agli sviluppatori di aggiungere nuovi nodi alla rete blockchain secondo necessità. Questo è fondamentale, poiché la tecnologia

blockchain permette agli utenti di operare “nodi”, ovvero computer che verificano e registrano le transazioni blockchain. La possibilità di aggiungere nodi rende le reti *blockchain* più scalabili.

Sicurezza

Le piattaforme *low-code* offrono una sicurezza aggiuntiva, proteggendo i dati sensibili da propagazioni non autorizzate. Molte delle piattaforme *low-code* includono funzionalità di sicurezza integrate, che le rendono ideali per lo sviluppo di applicazioni *blockchain*.

Velocità di lancio sul mercato

Il *low-code* accelera lo sviluppo delle applicazioni *blockchain*, riducendo il carico di investimenti in capitale, tempo e talenti [3](Álvaro Llobet,2022).

1.2.3 IA e Piattaforme Low Code: Democratizzazione dello Sviluppo Software e Rivoluzione del processo di Innovazione

L'integrazione tra intelligenza artificiale (AI) e piattaforme *low-code* ha dato vita a un potente motore di innovazione nel panorama dello sviluppo software. Questa combinazione dinamica consente alle aziende di creare e distribuire applicazioni con una velocità ed efficienza senza precedenti. In un contesto in cui le organizzazioni affrontano crescenti pressioni per digitalizzarsi e innovare, le piattaforme *low-code* “potenziate” dall'AI sono diventate strumenti essenziali per la trasformazione, permettendo sia agli utenti tecnici che non tecnici di contribuire attivamente alla rivoluzione digitale.

L'emergere di figure come i cosiddetti *citizen developers* (dipendenti non IT che creano applicazioni o migliorano i sistemi esistenti, spesso senza una formazione ufficiale nello sviluppo di software) segna un cambiamento significativo nell'approccio delle organizzazioni allo sviluppo software e alla trasformazione digitale. Le

piattaforme *low-code* basate sull’AI giocano un ruolo cruciale in questo processo, offrendo interfacce intuitive che permettono ai dipendenti di diversi reparti di creare applicazioni personalizzate per le loro esigenze specifiche, senza necessitare di competenze tecniche avanzate. Questo non solo accelera il processo di sviluppo, ma garantisce che le soluzioni siano allineate agli obiettivi aziendali.

Le piattaforme *low-code* hanno democratizzato lo sviluppo software, rendendolo accessibile anche a chi non possiede una formazione formale nella programmazione. In passato, creare un’applicazione anche semplice richiedeva conoscenze avanzate di coding e notevoli risorse. Con tali piattaforme, queste barriere sono state drasticamente abbassate. Gli utenti possono trascinare e rilasciare componenti, automatizzare flussi di lavoro e integrare sistemi esistenti attraverso interfacce grafiche intuitive. Questa semplicità d’uso consente ai dipendenti di assumere un ruolo attivo nel proprio percorso di trasformazione digitale, favorendo un’innovazione più rapida e un’agilità maggiore.

Le piattaforme *low-code* integrate con l’IA stanno, pertanto, trasformando il panorama dell’innovazione. Tradizionalmente, lo sviluppo di applicazioni era gestito prevalentemente dai reparti IT, creando colli di bottiglia dovuti a risorse limitate e priorità conflittuali. Con l’avvento di queste piattaforme, le singole unità aziendali sono ora in grado di guidare le proprie iniziative di sviluppo. Questo cambiamento non solo riduce la "dipendenza dall’IT", ma accelera anche il ritmo dell’innovazione, consentendo alle aziende di rispondere più rapidamente alle sfide e alle opportunità del mercato [4](Kovaion,2024).

1.2.4 Vantaggi e Sfide dell’Integrazione

L’intelligenza artificiale (IA) aggiunge ulteriori funzionalità alle piattaforme *low-code*, automatizzando compiti complessi, fornendo raccomandazioni intelligenti e migliorando le esperienze degli utenti.

L'IA può, infatti, automatizzare attività ripetitive e dispendiose in termini di tempo nel processo di sviluppo, come la generazione di codice, i test e il debugging, accelerando così lo sviluppo e riducendo al minimo gli errori umani. Inoltre, le piattaforme *low-code* potenziate dall'AI possono integrare capacità di elaborazione del linguaggio naturale (NLP), permettendo agli sviluppatori di creare chatbot e interfacce vocali con facilità, migliorando l'esperienza utente e aumentando la fidelizzazione dei clienti.

La capacità di analizzare grandi quantità di dati consente di fornire analisi predittive e supporto decisionale, prevedendo il comportamento degli utenti, suggerendo funzionalità utili o rilevando potenziali problemi prima che diventino critici. Questo approccio permette alle aziende di prendere decisioni più informate e di costruire applicazioni più robuste.

L'integrazione dell'AI nelle piattaforme *low-code* consente alle aziende di innovare più rapidamente ed efficacemente, permettendo di sviluppare e iterare rapidamente sui prototipi, analizzando i feedback degli utenti e suggerendo miglioramenti, accelerando così il ciclo di innovazione e permettendo alle aziende di portare nuovi prodotti e funzionalità sul mercato più velocemente. Un ulteriore beneficio di tale integrazione riguarda la riduzione dei costi di sviluppo, legata all'automatizzazione dei compiti di routine e alla minimizzazione della necessità di codifica estensiva, che permettono alle aziende di allocare le risorse in modo più efficiente e investire in altre aree di innovazione.

Nonostante i benefici qui presentati, risulta rilevante evidenziare alcune sfide legate all'introduzione di piattaforme *Low Code* integrate con l'IA. Un aspetto cruciale è l'integrazione con i sistemi esistenti: le applicazioni *low-code* integrate con l'IA devono essere compatibili con l'infrastruttura preesistente, il che può rappresentare una sfida significativa. È essenziale, infatti, garantire che le soluzioni siano in grado di integrarsi senza problemi con altre piattaforme e strumenti già in uso, al fine di evitare disfunzioni operative.

In aggiunta, pur riducendo la necessità di competenze avanzate di programmazione, le piattaforme *low-code* richiedono comunque una conoscenza di base dei concetti di IA e data science. Le aziende devono quindi investire nella formazione dei propri dipendenti, affinché possano sfruttare appieno il potenziale di queste tecnologie. Vantaggi e ostacoli legati all’implementazione di tali soluzioni verranno analizzati più approfonditamente nel Capitolo 4 dell’elaborato.

1.2.5 Low-code/No-code AI

Il termine *Low-Code/No-Code AI* si riferisce a quei tool esistenti che permettono di realizzare applicativi basati sull’IA, senza la necessità di scrivere codice tecnico, basati solitamente su un’interfaccia *drag-and-drop*, in cui si selezionano e combinano elementi visivamente, o un sistema guidato con domande e opzioni selezionabili da menu a tendina.

Nell’articolo [5] “*The 10 Best Examples Of Low-Code And No-Code AI*” di Forbes del 2022 vengono presentate alcune di queste soluzioni.

Amazon Sage Maker

Amazon, forte della sua esperienza nell’implementazione di modelli di *machine learning (ML)* in contesti consumer, offre SageMaker, una piattaforma che permette a chiunque di sfruttare questa competenza. SageMaker Jumpstart fornisce una selezione di template per le applicazioni di ML più utili per le aziende.

Akkio

Questo servizio consente di iniziare a utilizzare l’AI senza necessità di competenze di programmazione o data science. Akkio supporta la creazione di workflow basati sull’AI, con un’attenzione particolare alla rapidità di implementazione e valutazione.

Apple Create ML

La soluzione di Apple include una funzionalità *drag-and-drop* per creare applicazioni

iOS che coinvolgono raccomandazioni, classificazioni, riconoscimento delle immagini e elaborazione del testo. I dati possono essere raccolti tramite fotocamera o microfono di un iPhone e, con un Mac dotato di GPU, il processo di addestramento può essere accelerato e migliorato.

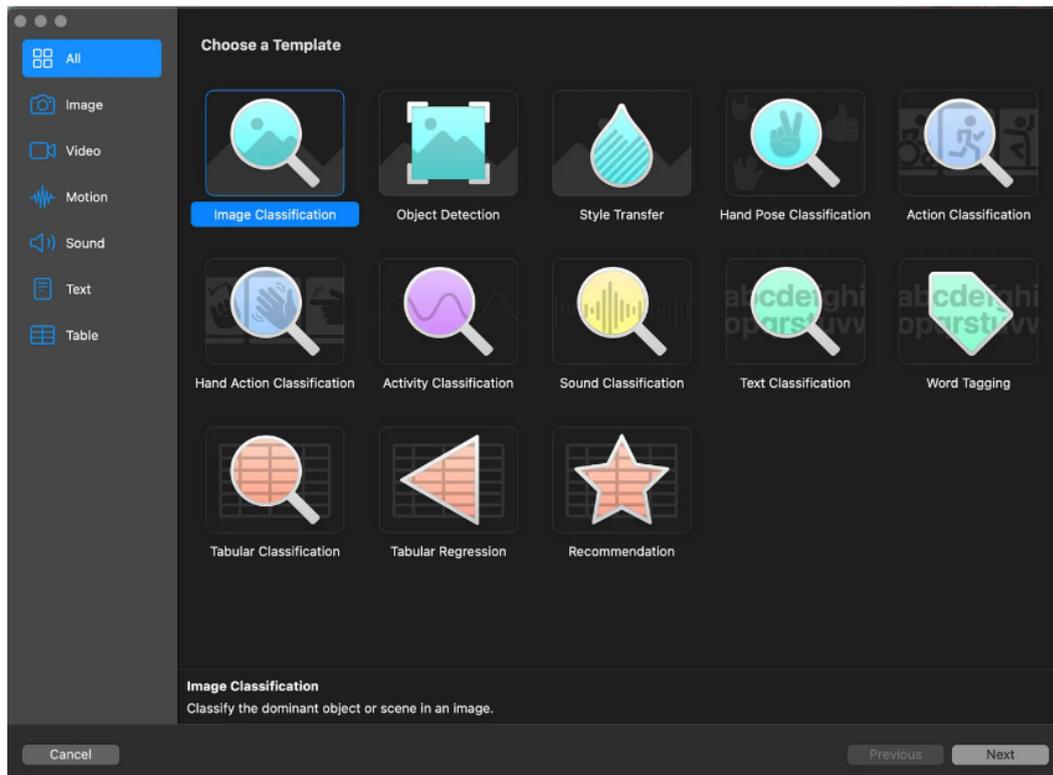


Figura 1.2. Apple Create ML Templates

Data Robot

Questa piattaforma cloud automatizza la preparazione dei dati e la costruzione di algoritmi, con modelli dedicati a settori come banche, vendita al dettaglio, sanità, produzione e pubblica amministrazione. Una caratteristica interessante è il focus sull'*Explainable AI*, un insieme di processi e metodi che consente agli utenti di comprendere e fidarsi dei risultati e degli output generati dall'IA.

Google Auto ML

La prima soluzione *no-code* di Google richiede una conoscenza minima del ML, ma

offre un'interfaccia grafica semplice per esplorare le funzionalità di visione artificiale ed elaborazione del linguaggio naturale. Tutto è integrato su *Google Cloud*, risultando familiare per chi usa altri strumenti Google.

Google Teachable Machine

Teachable Machine è ancora più adatta ai principianti rispetto ad AutoML, grazie a tutorial chiari e semplici che guidano nel processo di addestramento di algoritmi per classificare e categorizzare dati. È ideale come strumento educativo per imparare i concetti di base prima di passare a piattaforme più avanzate.

Microsoft Lobe

Lobe è uno strumento semplice per addestrare algoritmi di riconoscimento delle immagini. Microsoft l'ha sviluppato per introdurre gli utenti ai concetti base, selezionando automaticamente i modelli più adatti alle loro esigenze.

1.3 Il mercato delle Piattaforme Low Code

Dal report di Fortune Business Insights, [6]“*Low Code Development Platform Market Size, Share [2032]*”, emerge come il mercato globale delle piattaforme di sviluppo *low-code*, valutato 22,25 miliardi di dollari nel 2023, crescerà da 28,75 miliardi di dollari nel 2024 a circa 264 miliardi di dollari entro il 2032, con un CAGR del 32% durante il periodo di previsione (2024-2032).

1.3.1 Le ragioni della rapida espansione

Le piattaforme di sviluppo *low-code* (LCDP) aiutano a semplificare e ottimizzare la creazione di applicazioni aziendali sia per professionisti che per i cosiddetti *citizen developers*. I continui progressi tecnologici in diversi settori, inoltre, stanno alimentando la crescente domanda per tali piattaforme.

Le soluzioni *low-code* stanno diventando sempre più centrali nelle operations aziendali grazie alla loro capacità di accelerare lo sviluppo dei progetti a costi inferiori.

Le piccole e medie imprese (PMI), che affrontano sfide dovute a risorse e fondi limitati, sono sempre più propense ad adottare piattaforme di sviluppo di applicazioni istantanee.

Le sfide legate alla diffusione su larga scala dell'IA, quali la carenza di talenti qualificati e i costi di implementazione elevati, hanno portato all'emergere delle già citate *Low Code AI*, con i relativi vantaggi in termini di analisi predittiva dei dati, miglioramento dell'esperienza utente, riduzioni dei costi, processi di sviluppo accelerati. Di conseguenza, si prevede che l'aumento dell'adozione di strumenti di *Low Code AI* spingerà l'espansione del mercato dell'IA.

La crescente attenzione verso il miglioramento delle operations aziendali sta guidando le iniziative di *digital transformation* nelle istituzioni finanziarie. Attraverso la *digital transformation*, le aziende cercano di migliorare la qualità dei servizi e la redditività semplificando i flussi di lavoro. Utilizzando piattaforme di sviluppo *low-code/no-code*, le compagnie assicurative possono ridurre notevolmente i costi degli investimenti IT, adattarsi rapidamente ai cambiamenti ambientali e migliorare l'efficienza operativa riducendo lo skills gap tra i dipendenti. Le piattaforme di sviluppo *low-code* sono diventate indispensabili per la trasformazione digitale, dando ai team di sviluppo maggiore autonomia e riducendo le spese per l'acquisizione di risorse.

Ad esempio, nell'ottobre 2023, InsureMO e Mendix hanno collaborato per aiutare le compagnie assicurative a sviluppare e migliorare le loro iniziative di trasformazione digitale. La collaborazione ha unito lo sviluppo di applicazioni *low-code* di Mendix, la piattaforma *middleware* (software intermediario tra applicazioni, strumenti e database) di InsureMO e l'esperienza nel settore assicurativo per migliorare il processo decisionale.

La società di consulenza Gartner evidenzia nel report, [7] "*Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 20% in 2023*", altri due

fattori chiave che guideranno l'adozione delle tecnologie *Low Code* nei prossimi anni.

L'interesse per l'iperautomazione continua a crescere a causa dell'ampliamento del divario di competenze, della crescente domanda di ottimizzazione operativa, e delle pressioni economiche sempre più forti. Gartner prevede che la spesa per le tecnologie che abilitano l'iperautomazione continui a crescere nei prossimi anni e che una porzione considerevole di questa spesa sarà destinata alle tecnologie di sviluppo *low-code*, tra cui *LCAP*, *iPaaS*, *RPA*, *CADP* e *MXDP*, per supportare casi d'uso come l'automazione dei processi, l'integrazione e l'analisi delle decisioni.

Qui di seguito una panoramica di queste tecnologie e la relativa crescita in termini di revenues, prevista nel periodo (2021-2024).

Table 1. Low-Code Development Technologies Revenue (Millions of U.S. Dollars)

	2021	2022	2023	2024
Low-Code Application Platforms (LCAP)	6,324	7,968	9,960	12,351
Business Process Automation (BPA)	2,416	2,585	2,761	2,940
Multiexperience Development Platforms (MDXP)	2,081	2,508	2,999	3,563
Robotic Process Automation (RPA)	2,350	2,892	3,401	3,879
Integration Platform as a Service (iPaaS)	4,680	5,668	6,668	7,838
Citizen Automation and Development Platforms (CADP)	554	732	953	1,232
Other Low-Code Development (LCD) Technologies*	92	109	126	146
Total	18,497	22,462	26,869	31,949

*Other low-code development technologies include rapid mobile app development (RMAD) tools and rapid application development (RAD) tools. RAD tools are/were on-premises only and desktop-bound applications mainly. Low code is the evolution of RAD to cloud and SaaS models.

Notes: Gartner defines a no-code application platform as an LCAP that only requires text entry for formulae or simple expressions. The LCAP market, therefore, includes no-code platforms. Furthermore, "no code" is not a sufficient criterion for tasks like citizen development, as many complex tooling configuration tasks are no code but still require specialist skills. Columns may not add to totals shown because of rounding.

Source: Gartner (December 2022)

Figura 1.3. Revenues Tecnologie di Sviluppo Low Code

Gli investimenti nelle tecnologie *low-code*, inoltre, cresceranno anche man mano che le organizzazioni abbracciano il cosiddetto modello del *Composable Business*. Le imprese *composable* richiedono un migliore riutilizzo delle *packaged business capabilities* (componenti software standardizzati e preconfezionati che rappresentano

specifiche capacità e funzioni aziendali) per lo sviluppo agile di applicazioni e per creare esperienze utente personalizzate per nuovi flussi di lavoro e processi.

Tale concetto viene così illustrato da Jason Wong, VP Analyst di Gartner: "*Low-code development technologies are supporting the composable enterprise by enabling the creation of more agile and resilient software solutions. These technologies can be used to compose and recompose modular components and PBCs, to create adaptive custom applications for changing business needs.*" [7] (Gartner, 2022).

Un ulteriore fattore rilevante, che non può che spingere l'adozione delle tecnologie *low code*, riguarda il salario degli sviluppatori esperti. Esiste una domanda considerevole per le loro competenze, ma l'offerta è limitata. La maggior parte delle PMI non assume ingegneri software, a causa del loro costo elevato. Un approccio *low-code* per lo sviluppo di applicazioni aziendali richiede meno risorse specializzate per lo sviluppo ed è quindi più attraente per le piccole imprese.

Le tecnologie *low code* permettono perciò agli sviluppatori più esperti di concentrarsi su progetti più strategici e critici per l'azienda.

1.3.2 I fattori che potrebbero limitare l'espansione

Le tecnologie *low-code*, nonostante gli evidenti vantaggi che ne promuovono l'espansione, presentano alcune limitazioni che potrebbero frenare la crescita del mercato di queste piattaforme.

Uno dei principali ostacoli è la mancanza di integrazione universale tra i tool e le piattaforme basate su *low-code* con i diversi sistemi software. Di conseguenza, sviluppatori e progettisti devono considerare queste limitazioni di integrazione durante lo sviluppo del software per evitare potenziali problemi futuri.

Allo stesso modo, altre limitazioni significative di questa tipologia di piattaforme risultano essere la flessibilità e la possibilità di personalizzazione limitate. Solo alcuni fornitori offrono applicazioni su misura che si allineano a modelli di business specifici.

Le scarse possibilità di personalizzazione delle applicazioni e le difficoltà di integrazione con soluzioni software avanzate rappresentano, pertanto, un vincolo per l’espansione del mercato.

1.3.3 Settori di applicazione e Panorama Competitivo

Considerando i settori di applicazione delle tecnologie *low code* il mercato è suddiviso in *BFSI* (banche, servizi finanziari e assicurazioni), sanità, istruzione, IT e telecomunicazioni, media e intrattenimento, manifattura, governo, retail e altri.

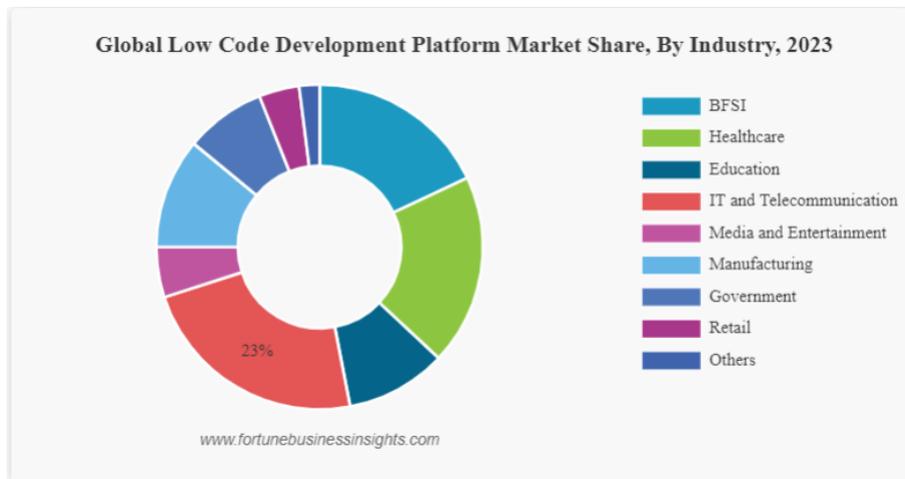


Figura 1.4. Settori di Applicazione Piattaforme Low Code

Il segmento IT e telecomunicazioni detiene la quota più alta del mercato (23%). Le piattaforme di sviluppo *low-code* aiutano i dipartimenti IT a migliorare l’efficienza e a semplificare i processi complessi. Sfruttando l’ascesa dei *citizen developers*, aziende e organizzazioni possono ridurre i colli di bottiglia IT e il sovraccarico di lavoro, eliminando molte attività ripetitive.

L’applicazione di piattaforme *low code* nel segmento della sanità, seppur già ampiamente diffusa, dovrebbe crescere rapidamente durante il periodo di previsione (2024-2032). Inoltre, Le normative e i requisiti di conformità in continua evoluzione, congiuntamente ai rischi crescenti per la sicurezza dei dati, incentiverebbero

l'adozione di strumenti di sviluppo *low-code*.

In previsione della crescente adozione di tali tecnologie, i principali attori del mercato stanno stipulando partnership strategiche e collaborando con altri leader del settore per ampliare il proprio portafoglio e offrire strumenti *low-code* e *no-code* avanzati che soddisfino le esigenze applicative dei clienti.



Figura 1.5. Key Companies del Settore

Capitolo 2

Panoramica delle Soluzioni

Low-Code Attuali e l'Approccio

DevOps

2.1 Introduzione

Questo capitolo fornisce una panoramica delle soluzioni *low code* attualmente disponibili sul mercato, evidenziando il loro impatto sulla semplificazione dei processi di sviluppo e sulla promozione di metodologie collaborative come *DevOps*.

Dopo l'analisi delle piattaforme Mendix e Appian, il focus si sposterà su Microsoft Power Platform e i suoi applicativi principali. Questo esame preliminare fornirà le basi per comprendere il contributo concreto di Power Platform nell'ambito del progetto presentato nel Capitolo 3.

2.2 Mendix

2.2.1 Origini, Vision e Mission

Mendix è una piattaforma *cloud-based* di sviluppo applicazioni *low-code*. Lanciata nel 2005 da Derek Ross e Derckjan Kruit, è stata progettata per creare rapidamente applicazioni reattive e di alta qualità, compatibili con desktop, tablet e smartphone. Dal 2018, Mendix è una sussidiaria di Siemens e rappresenta, ad oggi, una delle piattaforme leader di sviluppo *low code* sul mercato.

Mendix consente la realizzazione di una vasta gamma di soluzioni innovative e flessibili, adattabili a diversi contesti. Grazie alla sua versatilità, è possibile sviluppare applicazioni native per iOS e Android, *Progressive Web Applications (PWA)* e *Responsive Web Applications (RWA)*, offrendo esperienze digitali ottimizzate per vari dispositivi. Inoltre, la piattaforma supporta la creazione di dashboard e applicazioni front-end integrabili con altre soluzioni esistenti, così come strumenti e sistemi complessi destinati a migliorare l'efficienza dei processi interni di un'organizzazione.

È stata fondata per affrontare il principale problema nello sviluppo software: la disconnessione tra business e IT. La difficoltà nel trasmettere le esigenze aziendali al reparto IT, con le relative complessità nel tradurle in un software adeguato, derivano dall'approccio tradizionale al codice, che limita il processo di sviluppo ai soli programmatori, pur richiedendo, però, gli input per la realizzazione del software ai vertici aziendali. Questa "distanza" provoca ritardi, revisioni continue e opportunità mancate, fattori che riducono la capacità di innovazione rapida e reattiva.

Mendix si pone l'obiettivo di colmare questo divario, consentendo ai team aziendali di costruire applicazioni, utilizzando un linguaggio visivo comune [8](Mendix, 2024).



Figura 2.1. Ponte tra Business e IT (Mendix)

2.2.2 L'evoluzione della Piattaforma

Inizialmente la piattaforma era chiaramente suddivisa in due ambienti distinti.

1. *Mendix Studio*

Una soluzione *no code* basata su un editor WYSIWYG (What You See Is What You Get) che consente la creazione di applicazioni relativamente semplici senza alcuna programmazione. Rappresentava una buona opzione per principianti e utenti senza conoscenze tecniche che desiderano avvicinarsi allo sviluppo di applicazioni.

2. *Mendix Studio Pro*

Un vero e proprio ambiente *low code* con strumenti e funzionalità aggiuntive che offre agli sviluppatori, con le necessarie competenze tecniche, molta più flessibilità.

Nel giugno 2023, con il rilascio di Mendix 10, i due *IDE* (Integrated Development Environments) sono stati effettivamente fusi in uno solo. L'attuale versione di Mendix è una piattaforma *low code* progettata per sviluppatori di diversi livelli di competenza e background.

La motivazione alla base di questa decisione risulta piuttosto evidente: man mano che le piattaforme *low code* e *no code* maturavano come prodotti, i confini tra esse cominciavano a sfumare.

Allo stesso tempo, le persone che lavoravano con questi framework, i quali non

sempre iniziavano la loro carriera come sviluppatori software, acquisivano nuove competenze, spesso progredendo rapidamente, passando dall'utilizzo di Studio a Studio Pro.

È emerso, parallelamente, che coloro che possedevano conoscenze tecniche spesso utilizzavano strumenti *no code* per velocizzare il proprio lavoro, e viceversa, gli utenti con un background orientato al business richiedevano regolarmente funzionalità più complesse e opzioni che esigevano una certa abilità tecnica [9](Pretius, 2024).

Alla luce di queste considerazioni, cominció a perdere di significato la definizione di una sorta di “spettro” degli sviluppatori e la relativa associazione degli strumenti *low code/no code*, basata sul livello di esperienza delle diverse figure professionali. Concentrarsi su un unico ambiente flessibile risultò la scelta più opportuna, permettendo uno sviluppo più rapido di nuove funzionalità.

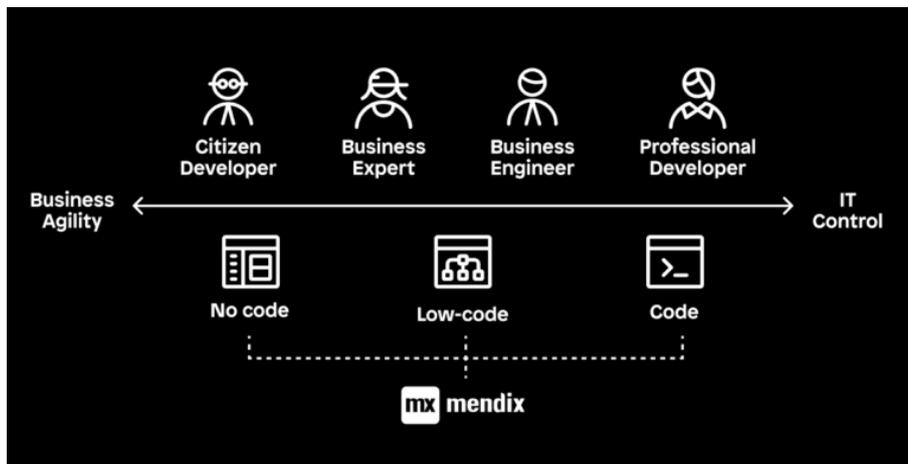


Figura 2.2. Visione di Mendix dello “spettro” degli sviluppatori

2.2.3 Struttura e Funzionalità Principali

La struttura, le funzionalità principali della piattaforma e i punti chiave dell’offerta vengono descritti sul sito web di Mendix e nella relativa documentazione

[10](Mendix, 2024).

Un'unica IDE, Estendibilità, Integrazione e Collaborazione

Mendix offre un unico ambiente di sviluppo, basato sui principi dello sviluppo *model-driven*.

Mendix Studio Pro consente a utenti con diversi livelli di competenza, dai *citizen developer* agli sviluppatori esperti, di partecipare alla creazione del software.

La piattaforma di sviluppo *low code* di Mendix è basata su standard aperti e tecnologie *open-source*, offrendo agli sviluppatori molteplici opportunità per estendere le funzionalità della piattaforma e delle applicazioni.

Le funzionalità principali della piattaforma sono accessibili tramite *API* (*Application Programming Interfaces*), ovvero protocolli che permettono di integrare tecnologie diverse e sfruttare servizi di terze parti. Gli sviluppatori possono, infatti, collegare Mendix a tecnologie esistenti e integrare servizi di test di terze parti, consentendo un'ampia personalizzazione e adattamento delle applicazioni.

Mendix facilita lo sviluppo in team grazie a strumenti utili come:

- *Mendix Portfolio Management*: utilizzando un inventario centralizzato i vertici aziendali e gli sviluppatori possono visualizzare tutte le iniziative, monitorare i progressi e collaborare, riducendo la distanza tra Business a IT.
- *App Insights*: il tool di Mendix per la gestione dei feedback degli utenti.
- *Sviluppo Agile*: fornendo *Epics*, uno strumento di pianificazione Agile, Mendix consente ai team di definire le proprie *epics* e *user stories* e pianificare il lavoro, utilizzando metodologie come *Scrum* o *Kanban*. *Epics* è integrato con gli altri strumenti, *Mendix Portfolio Management* e *App Insights*, per supportare l'intero ciclo di sviluppo collaborativo.

- *Version Control*: Mendix supporta l'uso di un *repository* di controllo versione centralizzato basato su Git, Mendix Team Server. Questo strumento consente ai team di sviluppo di gestire e sincronizzare facilmente il codice, mantenendo una visione centralizzata delle modifiche e delle versioni del progetto.

MAIA

La versione 10.12 di Mendix ha introdotto *Mendix AI Assistance* (Maia), un *AI Powered Assistant* che aiuta gli sviluppatori fornendo consigli e assistenza per determinate attività di sviluppo e persino generando parte delle applicazioni.

- *MAIA Chat*

Maia Chat è una chat integrata in Studio Pro, alimentata da intelligenza artificiale generativa.

Consente agli sviluppatori di fare domande sullo sviluppo di applicazioni in Mendix, come l'applicazione di concetti, le best practices e i modelli di sviluppo. Si basa su *large language models* (LLMs), modelli computazionali progettati per compiti di elaborazione del linguaggio naturale, come la generazione di linguaggio.

- *Assistenza in tempo reale*

Maia utilizza suggeritori contestuali per assistere gli sviluppatori Mendix con opzioni in tempo reale e guidate dal contesto. In particolare, Maia fornisce tre tipologie di Recommenders. Tali “suggeritori” comprendono:

1. *Best Practice Recommender*: un assistente virtuale intelligente che aiuta gli sviluppatori a migliorare le applicazioni, ispezionandole rispetto alle *best practices* di sviluppo in *Mendix Studio Pro*. Rileva e segnala gli *anti-pattern* durante la progettazione e lo sviluppo, suggerisce possibili soluzioni e, in alcuni casi, può correggere automaticamente queste criticità.

Qui di seguito viene mostrata una lista degli *anti-pattern* segnalati dalla piattaforma, ovvero procedure utilizzate nello sviluppo software che seppur lecite si potrebbero rivelare successivamente controproducenti.

Maia mostra una descrizione dell'*anti-pattern*, dell'elemento che ha causato la criticità e suggerisce le best practices a cui fare riferimento per risolvere il problema.

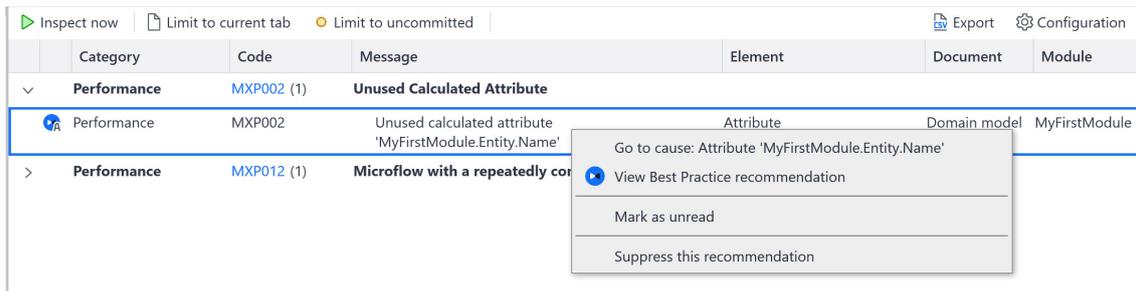


Figura 2.3. Best Practice Recommender (Maia)

2. *Logic Recommender*: un co-sviluppatore virtuale che assiste nella modellazione e configurazione della logica applicativa in *Mendix Studio Pro*, offrendo suggerimenti contestuali sulle azioni successive ottimali, basandosi sulle attività già progettate, su parametri e sul contesto.

Tra le principali funzionalità, include una ricerca intelligente che consente di individuare facilmente elementi da aggiungere al flusso, suggerimenti contestuali per le azioni più adatte e un'automazione che precompila i parametri richiesti. Inoltre, analizza il contesto osservando la logica circostante e la pagina di riferimento, evolvendo e migliorando continuamente, attraverso il training del modello.

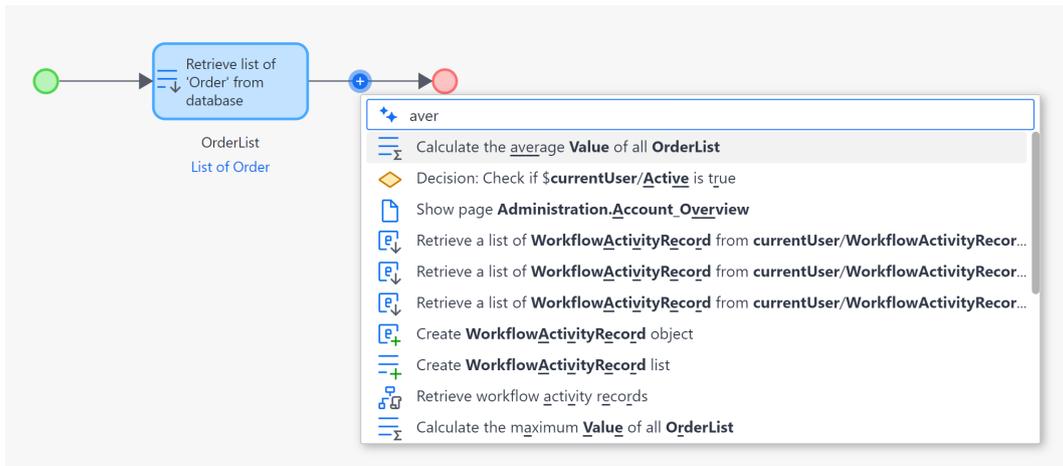


Figura 2.4. Logic Recommender (Maia)

3. *Workflow Recommender*: assiste nella modellazione e configurazione dei workflow in *Mendix Studio Pro*, tramite suggerimenti contestuali, simili a quelli forniti dal *Logic Recommender*, ma a livello di workflow.

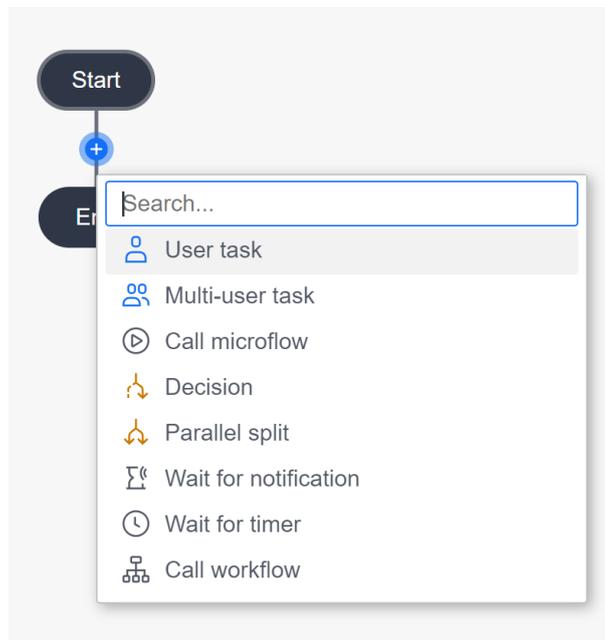


Figura 2.5. Workflow Recommender (Maia)

- *IA Generativa*

Il *Maia Domain Model Generator* è uno strumento, attualmente in fase sperimentale, che consente di generare un *domain model* (attualmente solo a partire da un domain model vuoto) creando entità e associazioni a partire da un input testuale.

Il *domain model* è un modello di dati concettuale, che astrae la struttura di un sistema di gestione di database relazionali (RDBMS) e rappresenta un elemento centrale nell'architettura di un'applicazione.

È costituito da entità, i relativi attributi, e dalle loro relazioni reciproche, rappresentate tramite associazioni.

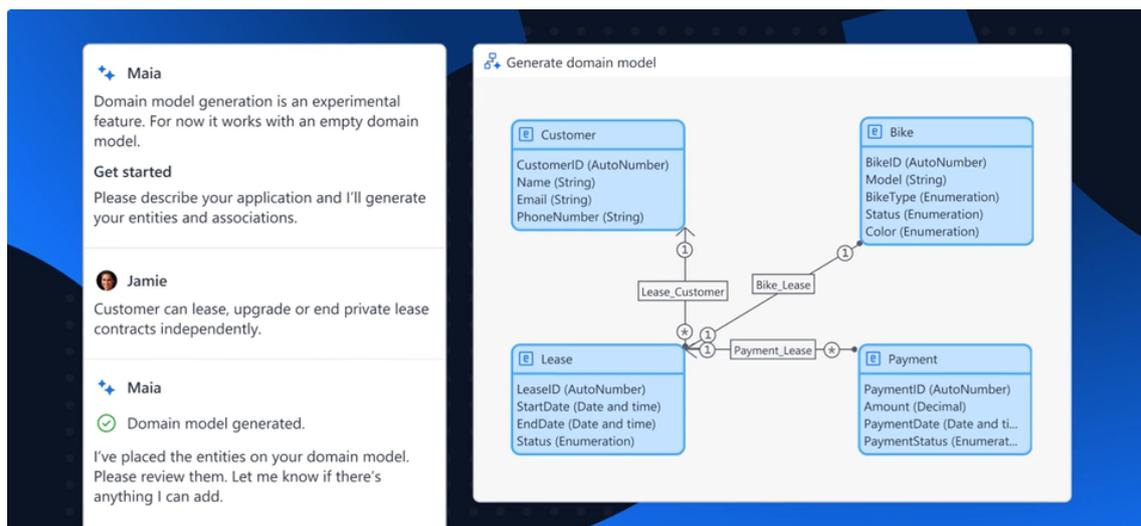


Figura 2.6. Maia Domain Model Generator

User Interface (UI) e User Experience (UX)

Mendix Studio Pro offre un'interfaccia utente intuitiva con temi personalizzabili, funzionalità di *drag-and-drop* e componenti riutilizzabili. È possibile creare applicazioni web multicanale e applicazioni mobili partendo da un unico modello. Gli utenti possono gestire facilmente gli elementi grafici delle diverse pagine come la struttura, il layout, i pulsanti e la logica lato client, semplicemente trascinando e rilasciando i componenti già pronti all'uso, sfruttando il framework *Atlas UI*, un linguaggio di design responsivo, *open-source* ed estensibile.

Atlas UI promuove, infatti, il riutilizzo di elementi e layout facilitando un approccio a strati, fornendo *page templates*, *building blocks* (combinazione di più elementi grafici in un'unità facilmente riutilizzabile) e *widgets* (componenti fondamentali dell'interfaccia, come bottoni e grafici), come modelli di avvio per accelerare lo sviluppo e garantire un'interfaccia utente coerente e standardizzata su più dispositivi. Questo approccio consente una progettazione più rapida ed efficiente, mantenendo al contempo l'uniformità dell'esperienza utente.

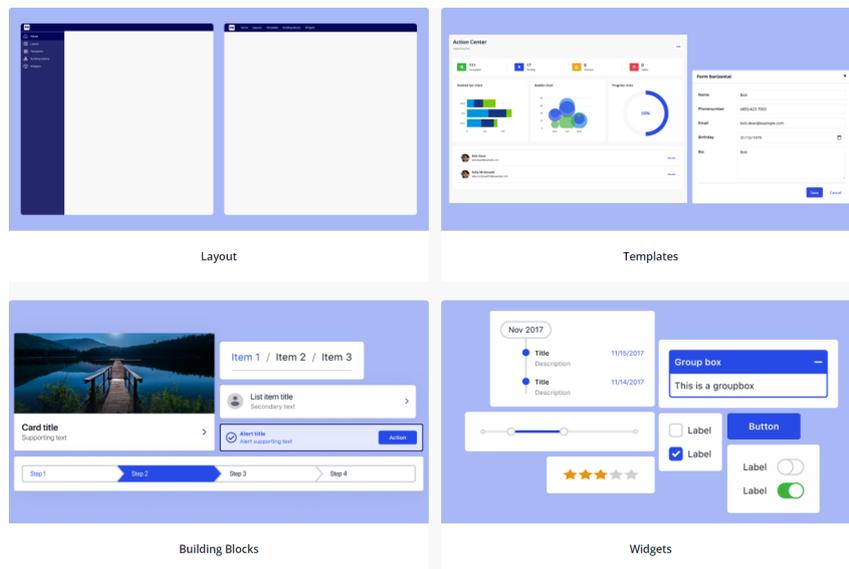


Figura 2.7. Atlas Design System

2.3 Appian

2.3.1 Origini ed Evoluzione verso il Low Code

Appian è stata fondata nel 1999 da quattro esperti del settore, Matt Calkins, Michael Beckley, Marc Wilson e Bob Kramer, con l'obiettivo comune di creare una piattaforma software in grado di ottimizzare e automatizzare i processi aziendali. Nei primi anni, Appian si è concentrata, infatti, sullo sviluppo di soluzioni di gestione dei processi aziendali (*Business Process Management*, BPM), fornendo alle organizzazioni strumenti per analizzare, modellare e ottimizzare i propri flussi di lavoro. Crescendo, l'azienda ha intravisto il potenziale di combinare il BPM con lo sviluppo *low code* per offrire una soluzione più potente, flessibile e facile da usare. Nel 2004, Appian ha introdotto la prima versione della sua piattaforma di sviluppo *low code*, permettendo agli utenti di creare applicazioni personalizzate con un uso minimo di codice. Questo rappresentava un momento cruciale per Appian, che ha spostato il proprio focus dalle soluzioni BPM tradizionali a un'offerta più completa basata sul *low code*.

Nel decennio successivo, Appian ha continuato a migliorare e ampliare la sua piattaforma, aggiungendo nuove funzionalità per soddisfare le crescenti esigenze dei clienti. Nel 2013, è stata lanciata *Appian Cloud*, una versione completamente *cloud-based* della piattaforma che ha semplificato ulteriormente la creazione, il *deployment* e la gestione delle applicazioni.

Con la progressiva espansione del mercato delle tecnologie *low code*, Appian si è affermata come leader nel settore, ottenendo riconoscimenti sia dagli analisti di mercato che dai clienti. Ad oggi, Appian è ampiamente riconosciuta come un pioniere nel settore dello sviluppo *low code* [11](Low-code Guru, 2023).

2.3.2 Funzionalità e Componenti Principali

Analizzando la documentazione fornita da Appian, quelli riportati di seguito risultano essere gli strumenti e le funzionalità messi a disposizione dalla piattaforma *low code* [12](Appian Documentation, 2024).

Data Fabric

Data Fabric di Appian è un'architettura di dati che consente di unificare, proteggere e ottimizzare la gestione dei dati distribuiti nell'intera azienda. Sfruttando questa struttura unificata, è possibile sviluppare rapidamente applicazioni *data-driven* per automatizzare i processi aziendali.

Data Fabric consente di connettersi direttamente alla posizione originaria dei dati, creando uno strato virtuale sopra i dati aziendali, permettendo di accedere a tutte le informazioni necessarie da un'unica piattaforma centralizzata.

L'utilizzo di un modello dati uniforme, che elimina la necessità di migrazioni complesse di dati, permette agli utenti di accedere e interrogare le informazioni aziendali in modo immediato, senza dover creare *API* personalizzate o scrivere codice per la gestione dei database.

L'accesso ai dati è ulteriormente migliorato da un sistema di sicurezza granulare che consente di controllare con precisione l'accesso alle informazioni. Tale sistema centralizzato, non solo garantisce la protezione dei dati, ma rende anche possibile l'accesso a un numero maggiore di utenti, garantendo un'esplorazione sicura dei dati aziendali.

La piattaforma è in grado, inoltre, di ottimizzare automaticamente le prestazioni delle applicazioni, eliminando la necessità di interventi manuali per migliorarne l'efficienza. Ciò permette agli sviluppatori di concentrarsi su attività critiche e strategiche, assicurando al contempo applicazioni performanti e capaci di rispondere alle richieste aziendali.

Un esempio di una possibile applicazione di *Data Fabric* consiste nella creazione di un'applicazione per la gestione degli ordini e della logistica per un'azienda, in cui i dati provengono da fonti diverse come distinti database e strumenti di *Enterprise Resource Planning* (ERP) e *Customer Relationship Management* (CRM).

Utilizzando *Data Fabric*, è possibile collegare le diverse fonti di dati e centralizzare l'accesso alle informazioni, semplificando lo sviluppo dell'applicazione. [13](Appian Documentation, 2024).

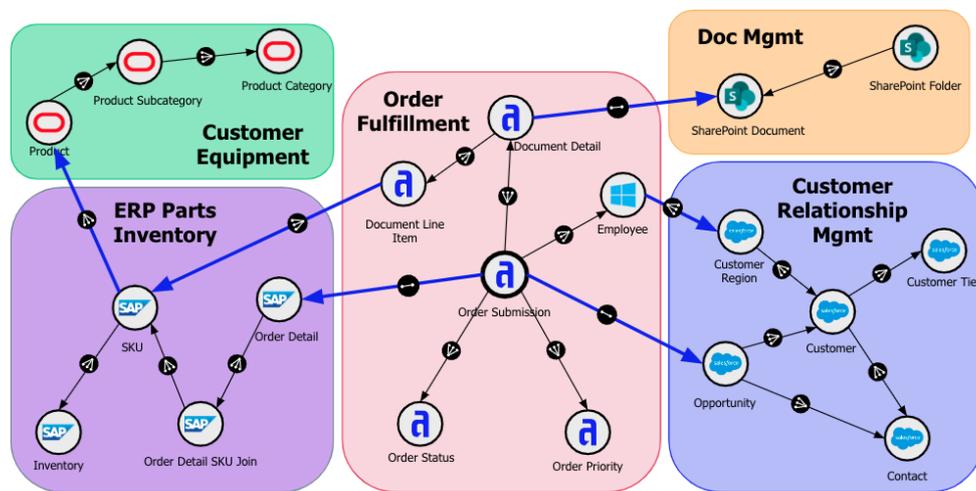


Figura 2.8. Integrazione Dati con Data Fabric

Sail Design System

Appian SAIL è il framework low-code utilizzato per progettare e sviluppare interfacce per le applicazioni.

Quando si creano interfacce in Appian, ovvero i componenti visivi con cui gli utenti interagiscono direttamente, è possibile operare in due modalità:

- *Design Mode:* consente di utilizzare funzionalità *drag-and-drop* e componenti riutilizzabili, come immagini, contenitori, pulsanti, per velocizzare la definizione del layout e la configurazione delle interfacce. Analogamente rispetto

al framework *Atlas UI* di Mendix descritto in precedenza.

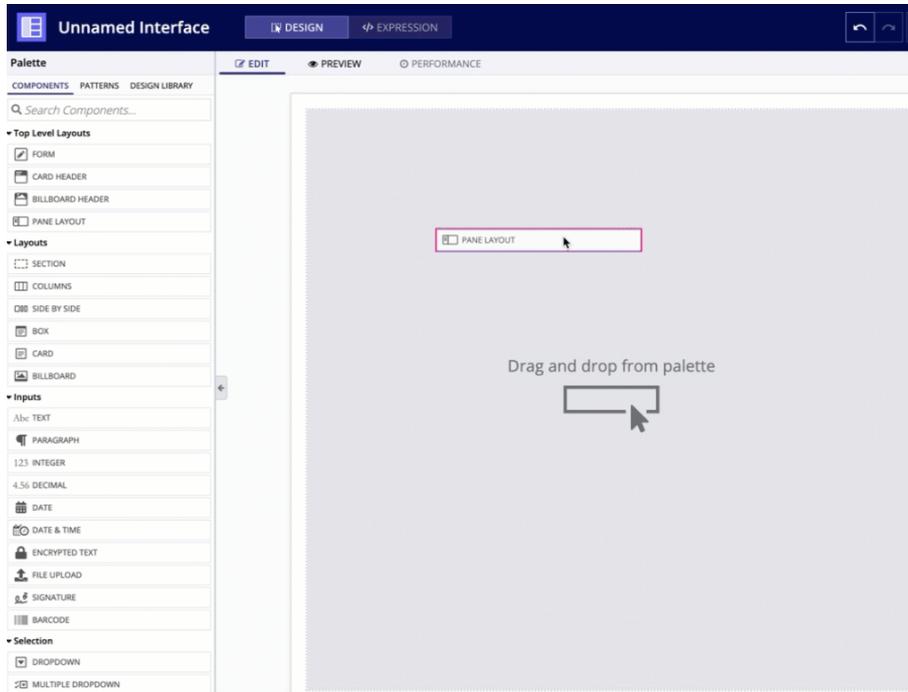


Figura 2.9. Funzionalità Drag-And-Drop Appian SAIL

- *Expression Mode*: mette a disposizione un editor di espressioni, le quali connettono le componenti dell'interfaccia utente, i dati e la logica applicativa in modo coerente. Un'espressione in Appian è un'istruzione composta da uno o più valori statici (stringhe, numeri interi o decimali, valori booleani), operatori, funzioni e variabili.

L'editor combina tre elementi fondamentali:

1. *Toolbar*: include strumenti per formattare e commentare il codice, oltre a opzioni per cercare, sostituire e navigare tra funzioni e domini. È possibile creare costanti, salvare espressioni selezionate e accedere rapidamente al *Query Editor*.

2. *Pannello di Editing*: la sezione vera e propria in cui scrivere l'espressione, sfruttando la funzionalità di *Auto-suggestions*, ovvero la capacità dell'editor di fornire suggerimenti in tempo reale sulla base di ciò che si sta digitando. L'*Expression Editor* suggerisce, infatti, funzioni, regole, parole chiave, costanti, tipi di dati o variabili che contengono i caratteri inseriti e corrispondono al dominio.

Durante la scrittura delle espressioni, vengono fornite, inoltre, delle linee guida di progettazione, ovvero modelli di *best practice* da implementare. Nel caso in cui vengano rilevati errori di sintassi, riferimenti errati o funzioni non supportate, vengono mostrate icone di *warning* (che devono essere risolti) o *recommendation* (che rilevano un modello di progettazione che va contro i modelli di *best practice*), con la possibilità di ottenere maggiori dettagli sulla criticità riscontrata.

3. *Documentazione*: selezionando un elemento all'interno dell'espressione viene mostrata la documentazione relativa.

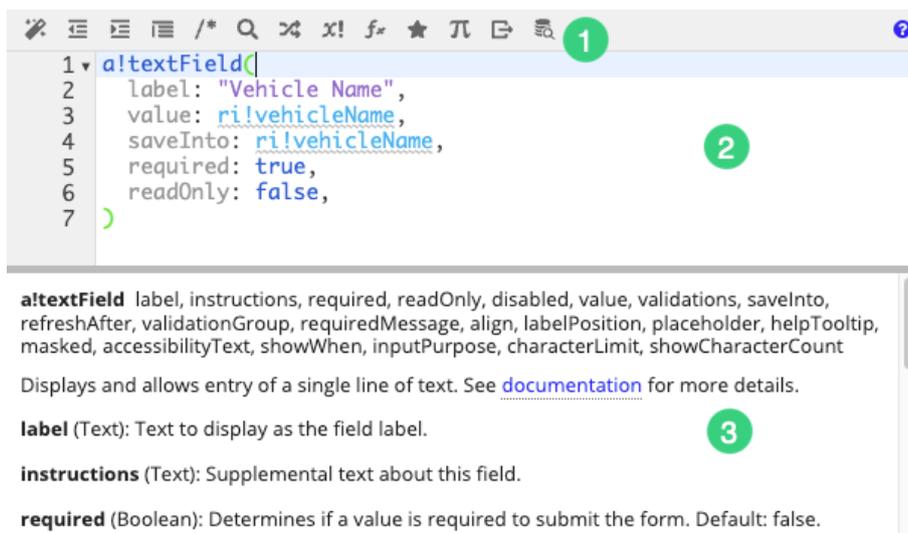


Figura 2.10. Expression Editor

Automazione dei Processi Aziendali

Una delle funzionalità distintive di Appian è la cosiddetta *end-to-end process automation*. Il punto di partenza per automatizzare i processi aziendali è rappresentato dal *Process Model Design Object*, strumento che permette di combinare persone, dati, IA e *robotic tasks*, in unico workflow, utilizzando la BPMN (*Business Process Model Notation*) per rappresentare graficamente i processi aziendali. [14](Appian Documentation, 2024)

È importante sottolineare, tuttavia, che questa rappresentazione non costituisce solamente il punto di partenza per costruire il processo, ma è il processo stesso. Appian traduce automaticamente i workflow aziendali in processi eseguibili.

I nodi del workflow comprendono *Human Tasks*, *Attività*, *Eventi* e *Gateway*. Le *Human Tasks* rappresentano attività che richiedono l'intervento umano, come l'assegnazione di un compito a un utente o a un gruppo. Le *Attività* vengono utilizzate all'interno del workflow per acquisire o elaborare dati aziendali. Gli *Eventi* consentono di avviare, interrompere o proseguire l'avanzamento del flusso, mentre i *Gateway* sono utilizzati per i controlli.

Una volta realizzato il modello di processo, Appian mette a disposizione una raccolta di nodi e *Smart Services* preconfigurati, che possono essere trascinati ed aggiunti al flusso, senza la necessità di dover scrivere alcuna riga di codice.

Gli *Smart Services* gestiscono l'integrazione con l'IA generativa (*AI Skill*), l'automazione dei processi attraverso la definizione di *robotic tasks* e l'integrazione con sistemi esterni tramite *API*.

- *AI Skill*: strumento che consente di creare, configurare e addestrare un agente di intelligenza artificiale (IA), permettendo di classificare testi, e-mail e documenti in base a tratti specifici stabiliti, estrarre informazioni, generare testo fornendo requisiti personalizzati e sintetizzare le informazioni chiave di

qualsiasi fonte. Inoltre, consente di rilevare informazioni personali identificabili (PII) per garantire una gestione sicura dei dati sensibili. [15](Appian Documentation, 2024)

- *RPA (Robotic Process Automation)*: l'obiettivo di Appian RPA è sostituire i task manuali con task "robotici" programmati. Tali "robot" interagiscono con le applicazioni allo stesso modo degli esseri umani, eseguendo compiti ripetitivi in base a regole predefinite e intervalli di tempo predefiniti.
- *Integrazione con Sistemi Esterni*: l'integrazione con sistemi esterni avviene dapprima tramite la creazione del *Connected System Object*, in cui memorizzare le informazioni di autenticazione e connessione (URL). Appian fornisce una raccolta di *Connected System Object* preconfigurati, verso i sistemi più comuni. In secondo luogo, dopo aver definito i dettagli della connessione, è necessario creare l'*Integration Design Object* in cui specificare l'URL completo, il metodo della richiesta HTTP (GET, POST...), il corpo, il relativo *content type* (JSON, XML...) e ulteriori parametri. In seguito alla definizione dell'integrazione, quest'ultima può essere facilmente aggiunta a qualsiasi modello di processo come *Smart Service*.

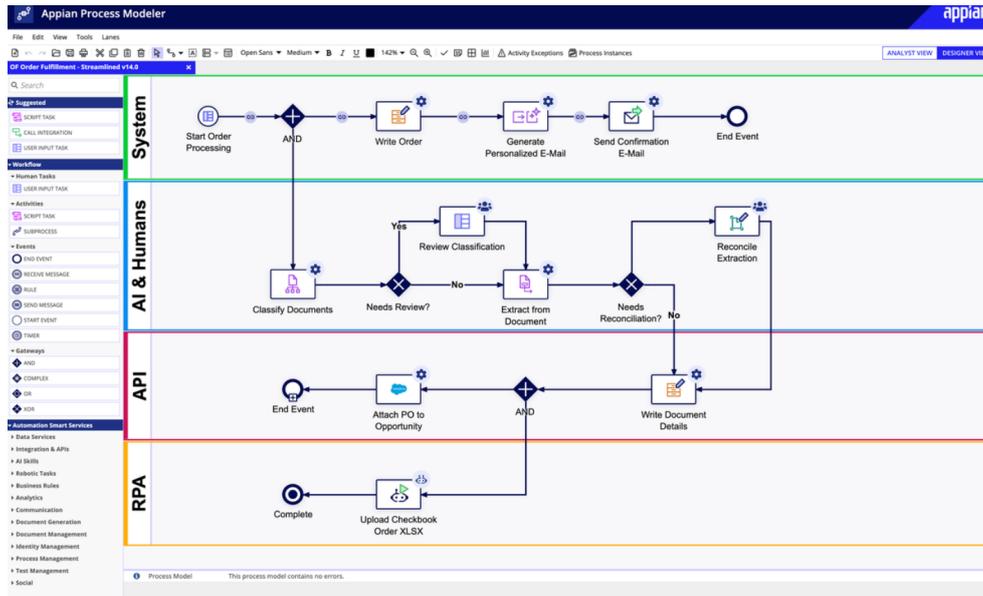


Figura 2.11. Appian Process Modeler Fonte: Appian

2.4 Microsoft Power Platform

2.4.1 Introduzione

Microsoft Power Platform è una piattaforma *low code* utilizzata per creare rapidamente soluzioni aziendali personalizzate *end-to-end*. Include attualmente cinque prodotti principali: Power Automate, Power Apps, Power BI, Power Pages e Copilot Studio (Ex Power Virtual Agents). Ognuno di questi strumenti può essere utilizzato singolarmente o in combinazione con gli altri, permettendo alle organizzazioni di analizzare i dati, automatizzare i processi, creare applicazioni personalizzate e migliorare le interazioni con i clienti.

Oltre ai prodotti principali sopra citati, Power Platform si basa su una serie di strumenti di supporto, fondamentali per il funzionamento della piattaforma:

- Connettori
- AI Builder

- Microsoft Dataverse
- Power FX

Un ulteriore aspetto da sottolineare è l'integrazione nativa con Microsoft Azure. Gli sviluppatori, infatti, possono sfruttare le competenze di sviluppo cloud per estendere Microsoft Power Platform con diversi servizi Azure [16](Microsoft Learn, 2024).

Power Automate, Power Apps e Power BI, ovvero i principali applicativi utilizzati per la realizzazione del progetto presentato nel Capitolo 3 e i relativi strumenti e servizi di supporto saranno oggetto di analisi di questo sottocapitolo.

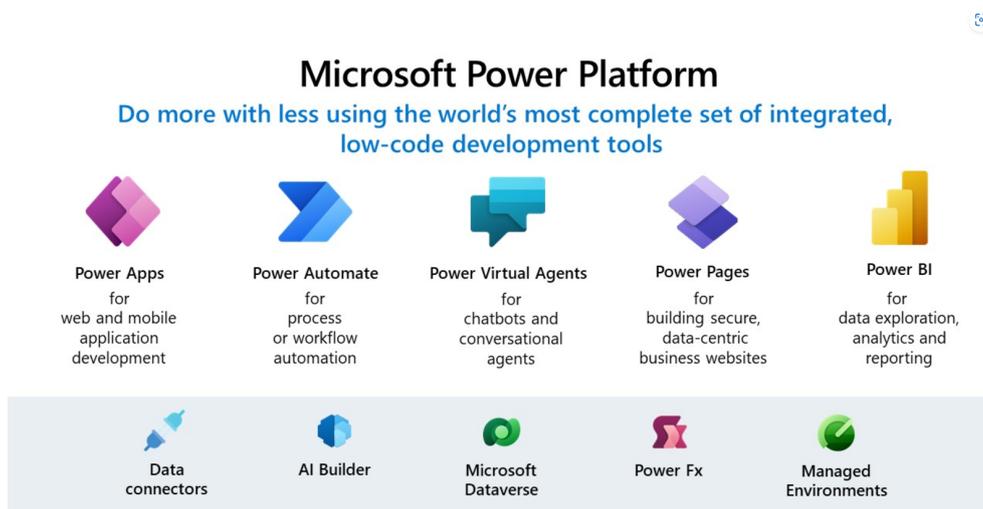


Figura 2.12. Panoramica Microsoft Power Platform

2.4.2 Power Automate

Microsoft Power Automate è una piattaforma *cloud-based* per l'automazione e la gestione dei flussi di lavoro, che consente agli utenti di automatizzare attività ripetitive e dispendiose in termini di tempo, creare integrazioni con applicazioni e servizi di terze parti, ottimizzare i processi aziendali e connettersi a diverse fonti

dati.

In virtù della propria interfaccia visiva *drag-and-drop*, Power Automate consente a chiunque, indipendentemente dalla preparazione tecnica, di creare flussi di lavoro personalizzati (chiamati "flows"), che possono essere attivati da eventi, pianificati o avviati manualmente. Power Automate si basa sull'utilizzo di connettori per interconnettere applicazioni e facilitare la comunicazione e l'esecuzione di attività. Attualmente sono disponibili circa 1000 connettori predefiniti, che oltre a comprendere applicativi Microsoft come Microsoft Teams, Sharepoint, Outlook e gli altri prodotti di Power Platform, includono Twitter, Dropbox, servizi Google, Office 365 e molti altri.

Questa piattaforma si presta, evidentemente, ad una notevole gamma di applicazioni, a partire dall'automazione di attività ripetitive, che consente di semplificare compiti quotidiani come l'invio di e-mail, notifiche ed avvisi, l'aggiornamento di file e l'immissione automatica di dati.

Inoltre, la possibilità di integrarsi perfettamente con altre applicazioni contribuisce ad ottimizzare processi quali approvazioni, gestione di progetti e documenti. Il funzionamento di Power Automate si basa su due entità: *trigger* e *azioni*. Un *trigger* avvia un flusso, ad esempio alla ricezione di un'e-mail nel proprio account Outlook. Un'*azione* è il passo successivo che viene eseguito una volta attivato il trigger, come pubblicare il messaggio di Outlook su Microsoft Teams.

Oltre alle capacità di automazione di Microsoft Power Automate, l'integrazione con Copilot, l'assistente AI di Microsoft, permette di creare flussi intelligenti ed efficienti. Copilot, analizzando il contesto del flusso in fase di creazione, fornisce suggerimenti per azioni, trigger e connettori.

Tipologie di Flusso

Microsoft Power Automate offre tre tipologie principali di flusso, ciascuno progettato per soddisfare esigenze specifiche di automazione: flussi cloud, flussi desktop e flussi per processi aziendali.

- *Cloud Flows*, ideali per automazioni che si attivano automaticamente, manualmente o secondo una pianificazione definita:
 - *Automated Cloud Flows*: permettono di creare un'automazione attivata da un evento specifico definito dall'utente. Per esempio, è possibile configurare un'automazione che scarica e salva automaticamente gli allegati delle e-mail ricevute in *Outlook* in una cartella specifica su OneDrive, eliminando la necessità di farlo manualmente.
 - *Instant Cloud Flows*: consentono di avviare l'automazione con un'azione manuale dell'utente, come l'invio di un promemoria.
 - *Scheduled Cloud Flows*: eseguono automazioni ricorrenti, come l'invio di riepiloghi giornalieri via e-mail o il caricamento settimanale di documenti.
- *Desktop Flows*: operano direttamente sul computer locale dell'utente, automatizzando attività come la gestione di file, l'apertura e chiusura di applicazioni, o l'invio di email. Questi flussi ampliano le funzionalità di automazione dei *robotic tasks* (RPA) offerte da Power Automate, permettendo di gestire processi ripetitivi basati sul desktop.
- *Business Process Flows*: rappresentano una soluzione avanzata per automatizzare intere sequenze operative, migliorando l'efficienza e la coerenza, riducendo le operazioni manuali e gli errori. Questi flussi possono gestire operazioni complesse

come l'onboarding dei clienti, la gestione delle vendite o l'approvazione di ordini di acquisto, fornendo un approccio strutturato e affidabile ai processi aziendali critici [17](Desk365, 2024).

Creazione di Un Flusso

Gli utenti possono scegliere se creare un nuovo flusso da zero, utilizzare modelli preesistenti o sfruttare Copilot per la generazione del flusso, a partire dalla descrizione del trigger e delle azioni necessarie. Per creare un flusso da zero è necessario specificare il trigger di partenza, manuale o automatico e poi aggiungere le azioni successive, ovvero i relativi connettori, compilando i parametri specifici richiesti.

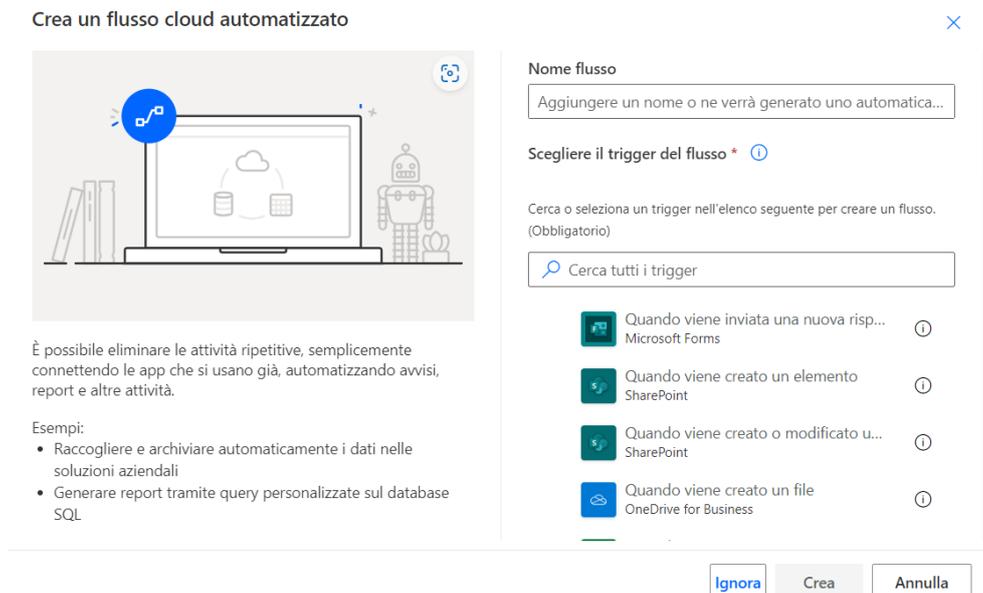


Figura 2.13. Scelta Trigger Cloud Flow

Qui di seguito i parametri richiesti per l'azione "Aggiorna Elemento" di una lista su Microsoft Sharepoint.

Aggiorna elemento

Parametri Impostazioni Code view Test Informazioni su

Indirizzo Del Sito *

Ad esempio: *https://contoso.sharepoint.com/sites/sitename*

Nome Lista *

Nome elenco SharePoint

ID *

Identificatore univoco dell'elemento da aggiornare

Parametri avanzati

Visualizzazione di 0 di 1

Mostra tutto Cancella tutto

Figura 2.14. Compilazione parametri azione Power Automate

Power Automate, tuttavia, permette anche di scegliere tra una vasta gamma di template predefiniti per la generazione dei flussi, permettendo di sfruttare le automazioni più utilizzate dagli utenti, fornendo lo schema di base del flusso.

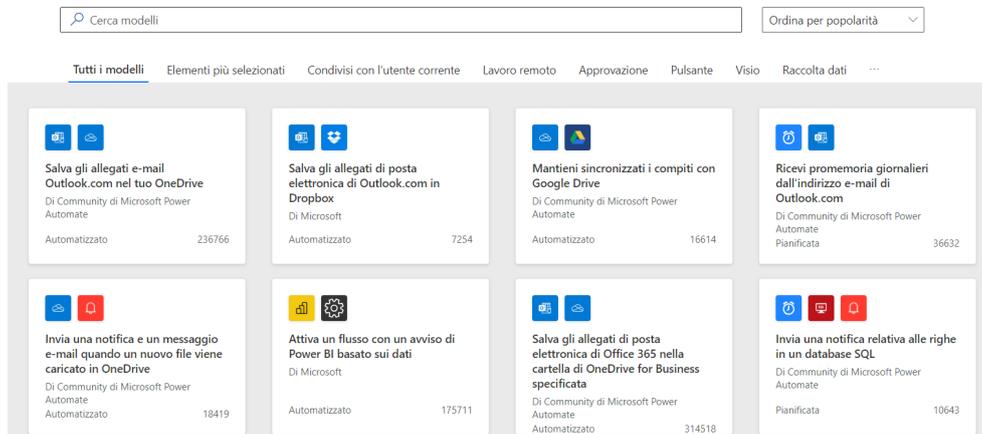


Figura 2.15. Modelli di Flusso Power Automate

Infine, come precedentemente sottolineato, è possibile generare un flusso semplicemente descrivendo a Copilot le automazioni richieste e gli applicativi coinvolti.

Quest'ultimo fornirà delle proposte da cui partire come base per la personalizzazione del flusso.

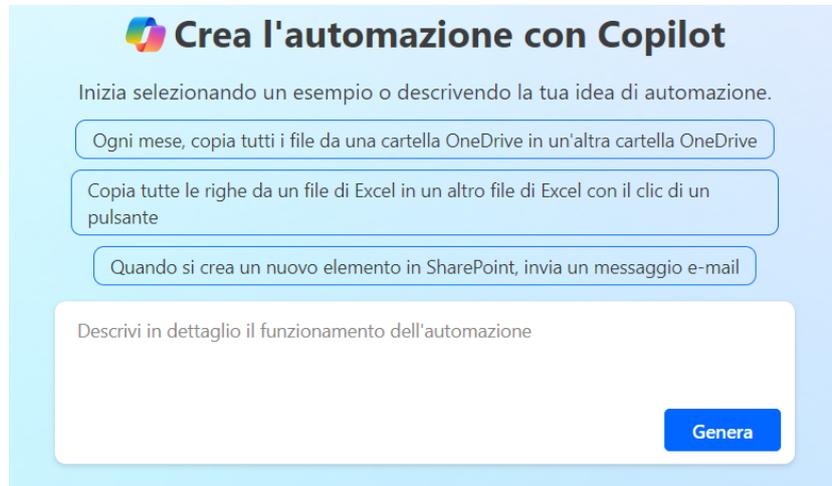


Figura 2.16. Generazione Flusso Con Copilot

2.4.3 Power Apps

Power Apps è un ambiente di sviluppo rapido per creare app personalizzate in base alle esigenze aziendali. Consente agli utenti di sviluppare applicazioni, accessibili sia su desktop che su dispositivi mobili, con il minimo utilizzo di codice, democratizzando il processo di sviluppo.

Utilizzando Power Apps, gli utenti possono creare app personalizzate che si connettono a dati archiviati nella piattaforma dati sottostante (Microsoft Dataverse) o provenienti da diverse fonti dati cloud e locali, come SQL Server, SharePoint ed Excel.

Creazione di un'Applicazione

La piattaforma offre la possibilità di creare un'applicazione da zero (*Blank Canvas App*) oppure a partire da modelli dati preesistenti (*Model-Driven App*).

- *Canvas Apps*

Dopo aver creato l'applicazione e aver definito il formato (tablet o dispositivo mobile), è possibile personalizzarla aggiungendo nuove schermate, controlli ed eventualmente collegandola a diverse fonti di dati.

Grazie ad un'interfaccia *drag-and-drop*, questo processo risulta semplice e intuitivo. È possibile inserire nell'applicazione elementi come pulsanti, icone, input di testo, form e tabelle, selezionandoli e adattandoli alle esigenze specifiche. Ogni elemento può essere configurato in modo dettagliato, modificandone proprietà come dimensioni, posizione e interazioni reciproche, per esempio disabilitando o rendendo invisibile un elemento alla selezione di un pulsante, alla selezione di un elemento in una combobox da parte dell'utente.

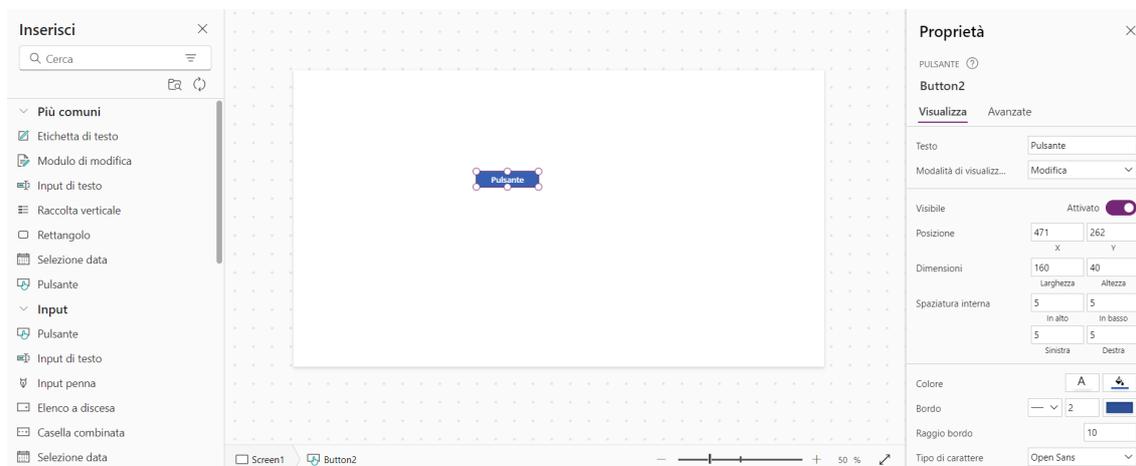


Figura 2.17. Interfaccia Drag-and-Drop Power Apps

Power Apps permette, inoltre, di inserire formule e funzioni all'interno di una barra delle formule, utilizzando *Power Fx*, un linguaggio intuitivo a basso utilizzo di codice tipizzato, dichiarativo e funzionale [18](Microsoft Learn, 2024).

Le formule consentono di manipolare gli elementi grafici in tempo reale, modificandone le proprietà e le interazioni. Possono essere applicate selezionando l'elemento desiderato, scegliendo la proprietà da modificare e scrivendo la funzione necessaria.

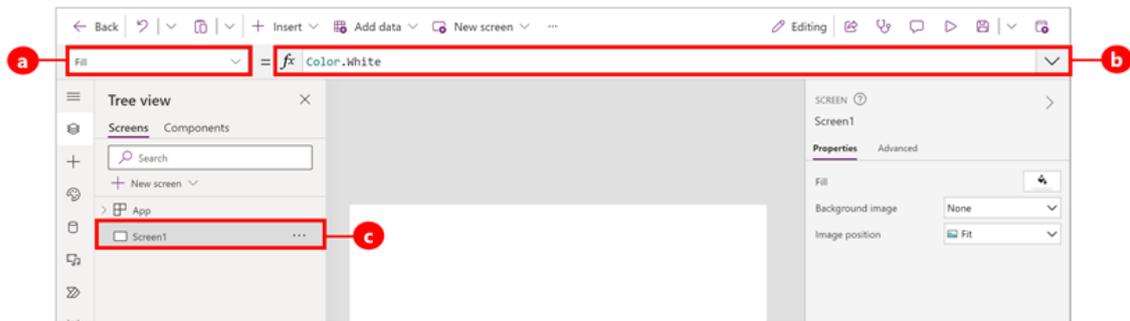


Figura 2.18. Barra delle Formule e Proprietà

- a) *Elenco delle Proprietà*: ogni elemento e schermata dispone di un insieme di proprietà selezionabili.
- b) *Barra delle Formule*: la formula per la proprietà selezionata è composta da valori, operatori e funzioni. Durante la digitazione, *Intellisense* (strumento di completamento automatico del codice, introdotto da Microsoft nel 1996) fornisce suggerimenti per formule, sintassi ed eventuali errori.
- c) *Elemento Selezionato*: nella barra delle formule è possibile visualizzare e modificare le proprietà dell'elemento selezionato o della schermata, se nessun elemento è selezionato. [19](Microsoft Learn, 2024).

È possibile utilizzare Copilot nella barra delle formule per spiegare le formule *Power Fx* in linguaggio naturale o generare formule *Power Fx* a partire da linguaggio naturale o commenti nel codice.

Un'ulteriore funzionalità consente la creazione di flussi da integrare con l'applicazione o l'aggiunta e modifica di quelli esistenti accedendo alla sezione relativa a Power Automate all'interno di *Power Apps Studio*.

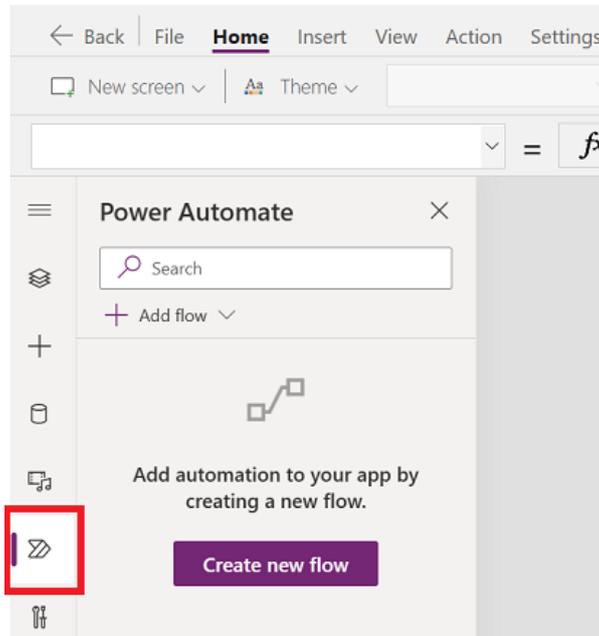


Figura 2.19. Power Automate Pane in Power Apps

- *Model-Driven Apps*

Le *Model-Driven Apps* vengono costruite a partire da un modello di dati, utilizzando un approccio *data-first*.

Questa tipologia di applicazioni costruisce un'interfaccia utente sopra Microsoft Dataverse (database relazionale e *data layer* di Power Platform), con il modello di dati sottostante che costituisce il fondamento dell'applicazione. Il primo passo essenziale nello sviluppo di queste applicazioni è strutturare correttamente i dati e i processi, mentre l'interfaccia utente e le funzionalità dell'app dipendono direttamente dal modello di dati, che guida l'intero processo di sviluppo.

L'approccio di sviluppo delle *model-driven apps* si concentra su tre aree principali: il modello di dati, la definizione dei processi e la composizione dell'applicazione. Il modello di dati si occupa di identificare i dati necessari e le loro relazioni,

mentre la definizione dei processi garantisce coerenza e uniformità operativa all'interno dell'app. Infine, la costruzione dell'applicazione vera e propria viene realizzata attraverso l'*App Designer*, che consente di aggiungere e configurare le pagine e i componenti dell'applicazione [20](Medium, 2023).

Confronto tra Canvas Apps e Model-Driven Apps

Le *canvas apps* e le *model-driven apps* rappresentano due approcci distinti alla creazione di applicazioni all'interno di Power Apps, ognuno con caratteristiche e vantaggi unici che le rendono adatte a specifici scenari.

Le *model-driven apps* si basano esclusivamente su Dataverse come piattaforma dati, mentre le *canvas app* offrono una maggiore flessibilità, consentendo di utilizzare non solo Dataverse ma anche altre fonti dati grazie ai connettori disponibili (Sharepoint, Excel, SQL...). Dal punto di vista dell'esperienza di progettazione, le *model-driven apps* si focalizzano su un design basato su componenti senza necessità di codice, mentre le *canvas apps* permettono una personalizzazione più approfondita tramite l'uso di espressioni *Power Fx* per manipolare le proprietà dei controlli.

Analizzando, invece, il controllo e la gestione dell'interfaccia utente, le *model-driven apps* offrono opzioni limitate di personalizzazione, poiché si basano principalmente sulle impostazioni predefinite dei componenti (approccio *data-first*). Al contrario, le *canvas apps* garantiscono un controllo completo, lasciando la libertà di definire ogni dettaglio dell'esperienza utente. Questa libertà, tuttavia, può influire sulla coerenza dell'applicazione, che, al contrario, tende ad essere garantita nelle *model-driven apps*, dove la struttura dipende principalmente dalle tabelle e dalle viste selezionate.

Il processo di migrazione tra ambienti risulta anch'esso differente. Per quanto riguarda le *model-driven apps* quest'ultimo è semplice, grazie alla struttura basata su Dataverse, mentre per le *canvas apps* può risultare più complesso, poiché potrebbe

essere necessario aggiornare le fonti dati. La velocità di creazione delle applicazioni, invece, varia in base al contesto: le *model-driven apps* consentono uno sviluppo più rapido grazie alla loro impostazione guidata, mentre il tempo necessario per creare una *canvas app* dipende dalla complessità del design.

Per quanto concerne la responsività, le *model-driven apps* sono automaticamente reattive e si adattano facilmente a diversi dispositivi, al contrario, nelle *canvas apps* la responsività deve essere progettata esplicitamente [21](Microsoft Learn, 2024).

2.4.4 Power BI

Microsoft Power BI è una piattaforma di visualizzazione dei dati utilizzata principalmente per scopi di *business intelligence*. Power BI o *Power Business Intelligence* comprende, infatti, un insieme di strumenti software e connettori che aiutano a trasformare dati provenienti da più fonti in informazioni utili e analizzabili.

Power BI è in grado di generare report e visualizzare i dati in una vasta gamma di stili diversi, tra cui grafici, mappe, diagrammi e grafici a dispersione.

Power BI è composto da diversi ambienti che è possibile utilizzare in sinergia in base alle proprie esigenze:

- *Power BI Desktop*: applicazione desktop per Windows.
- *Power BI service*: servizio online di tipo *software as a service* (SaaS).
- *App Power BI Mobile* per dispositivi Windows, iOS e Android.

Power BI Desktop

Power BI Desktop è un'app gratuita installabile in locale, che permette di connettersi ai dati, trasformarli e visualizzarli. Utilizzando questa applicazione è possibile connettersi a diverse fonti dati e combinarle in un modello unico, che consente di creare grafici, tabelle e altre visualizzazioni che si possono raggruppare in report, da condividere successivamente utilizzando Power BI Service.

Rappresenta uno strumento ideale per prendere dati grezzi, organizzarli, trasformarli ed infine rappresentarli graficamente.

In Power BI Desktop, l'esperienza di modellazione ruota attorno a tre sezioni principali, ognuna progettata per un aspetto specifico del processo di creazione e analisi.

1. *Report*: è il fulcro della creazione, ovvero dove gli utenti progettano report e visualizzazioni.
2. *Dati*: permette di esplorare tabelle, misure e altre informazioni del modello di dati, consentendo di trasformarli e ottimizzarli ai fini del report.
3. *Modello*: consente di gestire e visualizzare le relazioni tra le tabelle, offrendo un controllo preciso sulla struttura del modello di dati.

Il primo passo per utilizzare Power BI Desktop è connettersi ai dati. Il processo è semplice e supporta una vasta gamma di fonti, dai database ai file locali, fino ai servizi cloud.

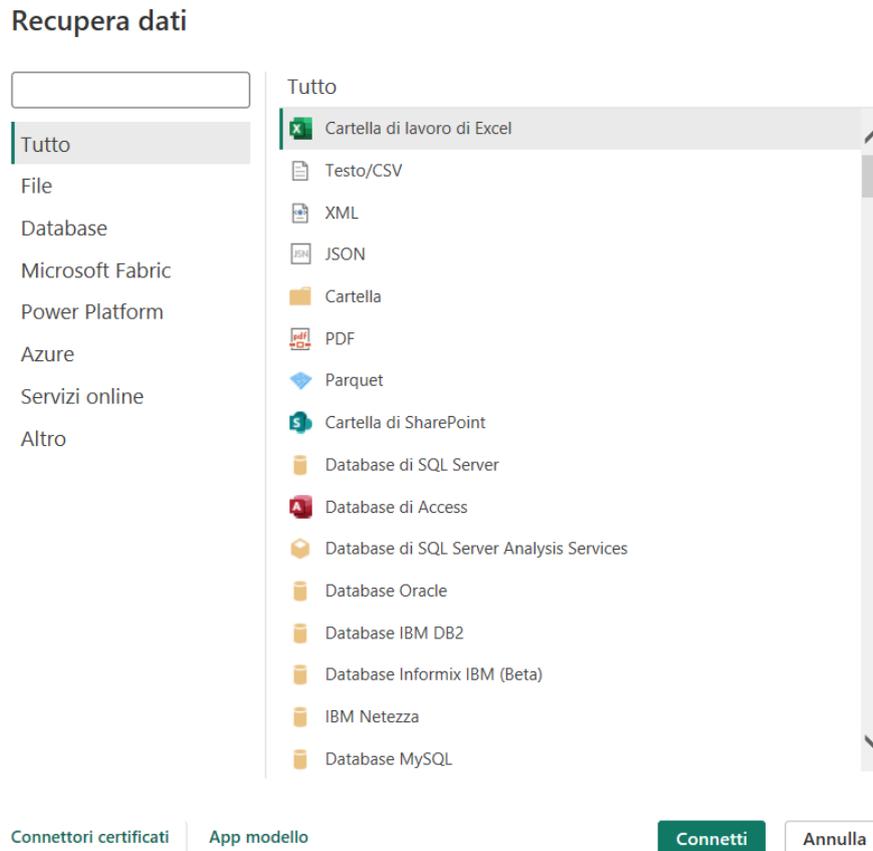


Figura 2.20. Connessioni alle Fonti Dati Power BI

Una volta stabilita la connessione, è possibile trasformare e pulire i dati utilizzando il *Power Query Editor*, uno strumento intuitivo che consente di modificare e strutturare i dati in modo che rispondano perfettamente alle esigenze del modello. Ogni modifica effettuata, come la rimozione di colonne o la combinazione di elementi da più fonti, viene registrata, garantendo che ogni aggiornamento mantenga le trasformazioni specificate.

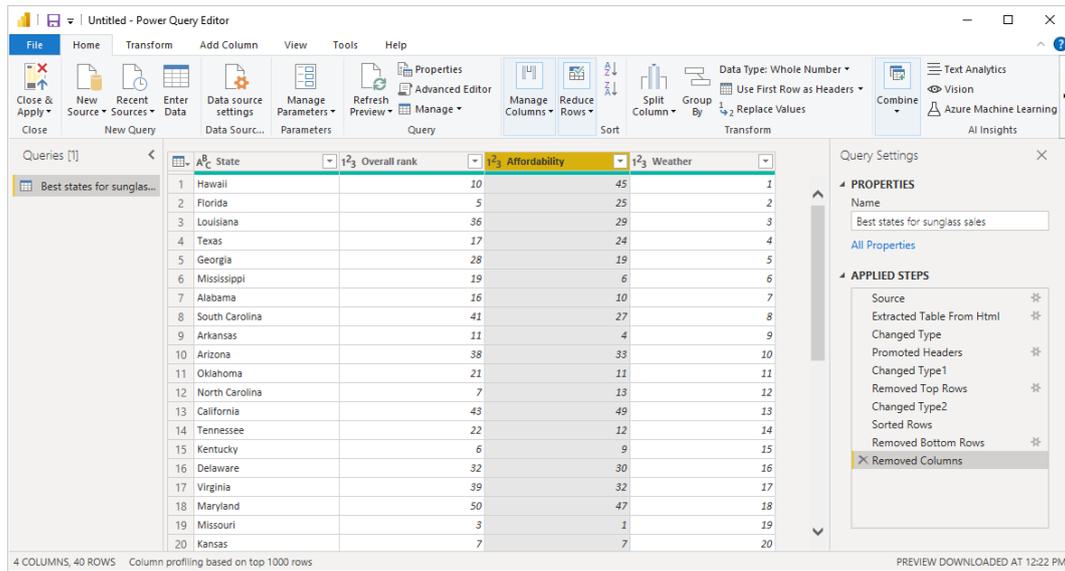


Figura 2.21. Power Query Editor

Dopo aver modellato i dati è possibile passare alla creazione di visualizzazioni. Trascinando i campi (come i campi di una lista Sharepoint o colonne di una tabella Excel) nell'area del report, è possibile generare rappresentazioni grafiche che rendono i dati più chiari e comprensibili.

La funzionalità principale, tuttavia, consiste nella creazione di una raccolta di visualizzazioni che mostrino vari aspetti dei dati utilizzati per costruire il modello in Power BI Desktop. Questa raccolta di visualizzazioni, contenuta in un unico file di Power BI Desktop, viene definita Report, il quale può comprendere una o più pagine, in modo analogo ai fogli di lavoro di un file Excel.

Dopo aver costruito il report, è possibile pubblicarlo su Power BI Service e renderlo disponibile a chiunque nella propria organizzazione possieda una licenza Power BI [22](Microsoft Learn, 2024).

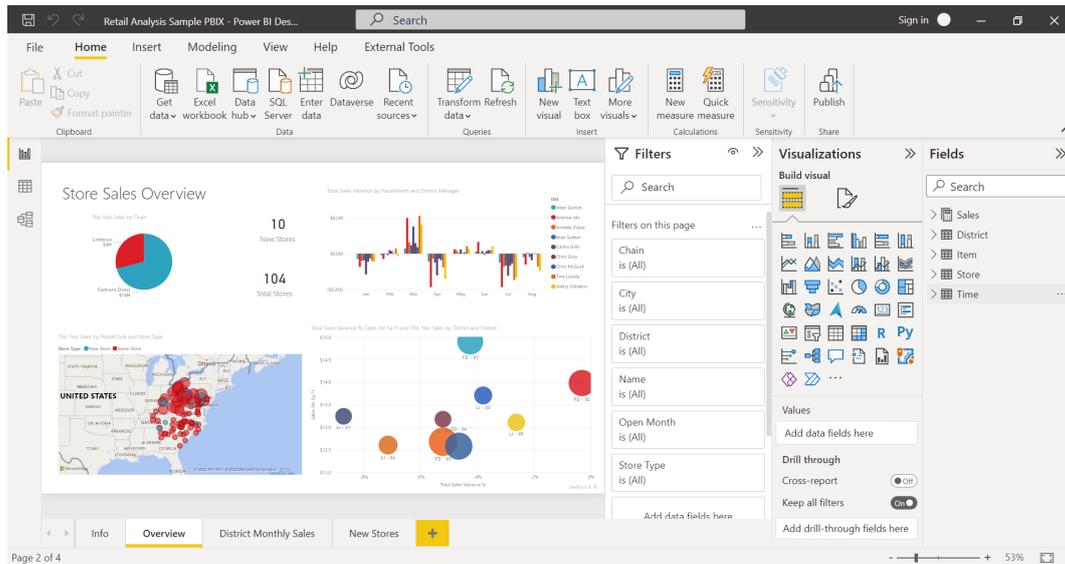


Figura 2.22. Report Power BI Fonte: Microsoft Learn

Power BI Service e Power BI Mobile Apps

Il Power BI Service rappresenta la componente *Software as a Service* (SaaS) di Power BI. Esso consente di visualizzare e interagire con i report pubblicati ed i singoli elementi visivi presenti all'interno, direttamente dal browser, supportando la collaborazione tra team e organizzazioni.

Sebbene sia possibile connettersi a diverse fonti dati anche in Power BI Service, le possibilità di modellazione sono limitate rispetto a Power BI Desktop. Power BI Service è infatti principalmente utilizzato per attività come la creazione di dashboard, la generazione e la condivisione di app, nonché per l'analisi e l'esplorazione dei dati al fine di scoprire insights utili per l'azienda.

Le *Power BI Mobile Apps* sono, invece, applicazioni mobili che consentono agli utenti di accedere e interagire con i propri report e dashboard di Power BI sul proprio dispositivo mobile.

Tali applicazioni offrono una serie di funzionalità avanzate che consentono agli

utenti di visualizzare e interagire con i report e dashboard in modo intuitivo, permettendo in primo luogo di esplorare diversi tipi di grafici, tabelle e visualizzazioni, facilitando l'identificazione di pattern, tendenze e insight significativi. In secondo luogo, gli utenti hanno la possibilità di filtrare ed esplorare i dati all'interno dei report e delle dashboard, applicando filtri, slicer e altre funzionalità interattive per approfondire specifici set di dati.

Una delle caratteristiche distintive delle *Power BI Mobile Apps* è la possibilità di aggiungere note, commenti o focus specifici direttamente sulle visualizzazioni, facilitando la collaborazione e la comunicazione tra i membri del team. È inoltre possibile condividere report e dashboard con altri utenti, sia all'interno dell'organizzazione che esternamente ed esportare le visualizzazioni dei dati in immagini o file PDF, semplificando l'inclusione di questi contenuti in presentazioni o report.

Infine, le *Power BI Mobile Apps* offrono la possibilità di configurare avvisi e notifiche, garantendo che gli utenti vengano informati quando vengono raggiunti determinati valori soglia o al verificarsi di condizioni specifiche. Questa funzionalità di monitoraggio aiuta gli utenti a rimanere aggiornati sui principali indicatori aziendali e a prendere decisioni tempestive quando necessario [23](Addend Analytics, 2024).

Il tipico flusso di lavoro in Power BI inizia con la connessione alle fonti dati in Power BI Desktop e la creazione di un report. Successivamente, il report viene pubblicato da Power BI Desktop su Power BI Service e condiviso, in modo che gli utenti aziendali su Power BI Service e sui dispositivi mobili possano visualizzarlo e interagire con esso. Questo flusso di lavoro è comune e mostra come i tre principali elementi di Power BI si complementino a vicenda.

2.5 Devops e Low-Code

Con il termine *DevOps* si fa riferimento ad un approccio moderno allo sviluppo software che integra e automatizza il lavoro dei team di sviluppo (Dev) e delle operations IT (Ops). Questo approccio promuove l'idea che una collaborazione stretta tra team, tradizionalmente separati, sia più efficace rispetto a lavorare in maniera isolata. L'obiettivo principale di *DevOps*, oltre a favorire una comunicazione trasversale tra team, è ridurre il ciclo di vita dello sviluppo software, senza però compromettere la qualità del software o la stabilità dei sistemi costruiti.

Nel contesto della metodologia *DevOps*, questo obiettivo viene raggiunto combinando processi, strumenti e mentalità diverse per garantire:

- *Continuous Integration (CI)*: pratica di sviluppo software in cui tutti gli sviluppatori effettuano le modifiche al codice portandole in un *repository* centralizzato e condiviso come, per esempio, *Git*, facilitando notevolmente il processo di integrazione delle nuove modifiche con il codice già esistente, rendendo più fluido lo sviluppo e la manutenzione delle applicazioni.
- *Continuous Delivery (CD)*: un approccio di ingegneria del software che predilige la produzione in cicli brevi, che garantiscano la possibilità di rilasciare il software in modo affidabile in qualsiasi momento. La *Continuous Delivery*, infatti, non vede le fasi di sviluppo, rilascio, feedback e gestione della qualità come irrimediabilmente separate, ma come un ciclo continuo applicabile ad ogni minima modifica che, come vuole la *Continuous Integration*, riceverà un feedback istantaneo e verrà rilasciata frequentemente.
- *Continuous Monitoring*: questo approccio prevede di iniziare a raccogliere feedback immediati già durante lo sviluppo e la distribuzione, riuscendo ad individuare e affrontare rapidamente eventuali problemi prima che si aggravino.

I cicli di feedback a lungo termine includono anche le informazioni raccolte nel tempo durante il periodo *post-deployment*.

La filosofia *DevOps*, oltre a fondarsi su un cambiamento culturale che promuove la collaborazione e la responsabilità condivisa tra i team di sviluppo e quelli operativi, si basa sull'automatizzare il più possibile le attività ripetitive, come il testing delle applicazioni, *deployment* e monitoraggio delle prestazioni.

Un ulteriore principio cardine della metodologia *DevOps* è l'attenzione verso la creazione di soluzioni software scalabili e affidabili, progettate per adattarsi a esigenze in costante evoluzione. Questo obiettivo viene perseguito attraverso un approccio strutturato e orientato alla continuità operativa, con la realizzazione di sistemi ridondanti e una particolare attenzione nella definizione di piani di *disaster recovery* [24](Akveo, 2024).

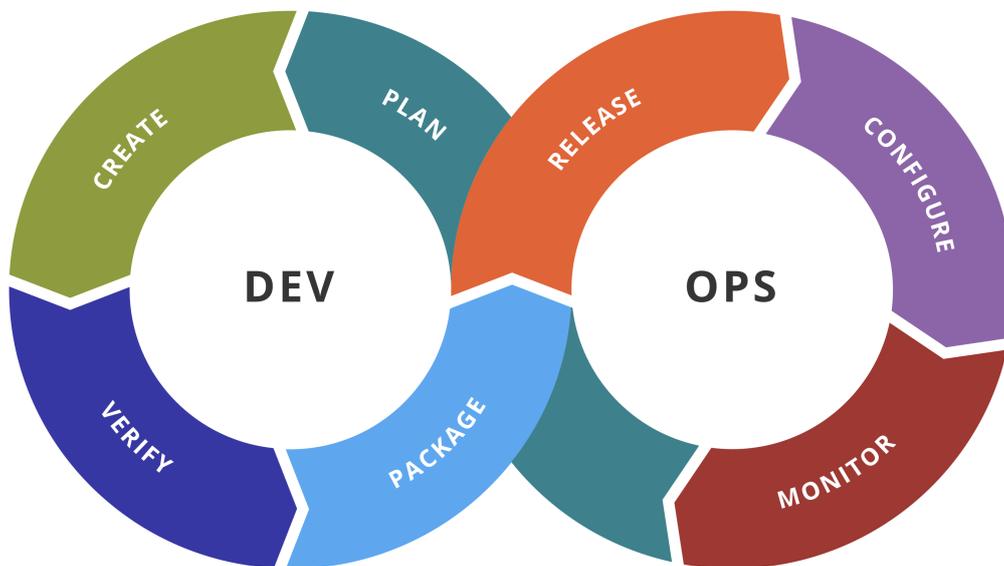


Figura 2.23. Ciclo di Vita Approccio DevOps

L'integrazione tra tecnologie *Low code* all'interno delle pratiche *DevOps* rappresenta un'opportunità significativa per migliorare l'efficienza e l'efficacia dello sviluppo

software. In particolare, una *pipeline CI/CD* ben definita (insieme di procedure che automatizzano il processo di integrazione, testing e rilascio delle modifiche al codice, sfruttando CI/CD, *Continuous Integration* e *Continuous Delivery/Deployment*), essenziale per il *DevOps*, può beneficiare delle caratteristiche distintive delle piattaforme *low code*, che semplificano il flusso di codice attraverso automazione e strumenti predefiniti.

In primo luogo, grazie ai connettori preconfigurati è possibile integrare i dati senza necessità di migrazioni di dati complesse, superando i tradizionali *silos* informativi (ambienti isolati e sconnessi tra loro). Inoltre, l'utilizzo di componenti riutilizzabili consente di risparmiare tempo e ridurre lo sforzo richiesto ai team. Questo approccio, combinato con la rapidità di implementazione di automazioni, tipica degli strumenti *low code*, consente di accelerare l'intero ciclo di sviluppo, obiettivo chiave della metodologia *Devops*.

Oltre a ciò, l'approccio *low code* agevola considerevolmente la collaborazione tra team. Attraverso la semplificazione del codice e rendendo visibile la struttura dell'applicazione, tutti i membri, inclusi gli utenti aziendali non tecnici, possono comprendere meglio i cambiamenti apportati e partecipare attivamente alla creazione e modernizzazione del software.

Un ulteriore elemento distintivo delle tecnologie *low code* applicate a *DevOps* riguarda il processo di testing. L'“approccio visivo” del *low code*, basato sull'utilizzo di interfacce grafiche intuitive, facilita l'identificazione degli errori e la loro risoluzione fin dalle prime fasi di sviluppo, riducendo il tempo e gli sforzi rispetto all'analisi manuale del codice. L'intuitività di queste tecnologie permette a tutti i membri del team, non solo agli esperti tecnici, di partecipare al controllo e all'aggiornamento del software, rafforzando la qualità del prodotto finale e garantendo software più sicuri e affidabili.

La gestione delle versioni e degli *incident* è un altro punto cruciale. Le piattaforme

low code integrate con l'approccio *DevOps* forniscono una documentazione centralizzata e affidabile, essenziale per workflow efficienti. Sfruttando strumenti *low code* che consentono di impostare alert per eventi critici, come cali di prestazioni o errori, è possibile risolvere tempestivamente le problematiche e mantenere la continuità operativa.

Infine, l'approccio *low code* offre strumenti per la gestione del ciclo di vita delle applicazioni in linea con i principi di *Continuous Integration e Delivery* di *DevOps*, permettendo di monitorare le prestazioni, ottimizzare i processi di testing e semplificare il debug. In questo modo, le piattaforme *low code* diventano veri e propri hub centralizzati per la gestione delle app, dando ai team una visione completa dei flussi e un maggiore controllo sul processo di trasformazione digitale [24](Akveo, 2024).

In sintesi, la combinazione tra *low code* e i principi *DevOps* accelera i cicli di sviluppo, agevola la collaborazione tra figure tecniche e non, migliora la qualità e l'affidabilità del software, supporta la scalabilità e garantisce maggiore flessibilità. A tutto ciò si aggiunge un impatto significativo in termini di costi, grazie alla riduzione del tempo di sviluppo e alla minimizzazione della dipendenza da competenze specializzate. Questo connubio rappresenta una soluzione ideale per le aziende che vogliono innovare rapidamente e rispondere con agilità alle esigenze di mercato.

Capitolo 3

Implementazione: Caso di Studio con Microsoft Power Platform

3.1 Scopo e Ambito del Progetto

Il progetto descritto in questo capitolo è stato sviluppato personalmente all'interno di *aizoOn Technology Consulting*, una realtà aziendale di consulenza, specializzata nel fornire soluzioni tecnologiche avanzate.

Nel contesto delle attività del gruppo di lavoro, è emersa l'esigenza di dotarsi di uno strumento centralizzato in grado di supportare il monitoraggio e la gestione dei progetti in modo sistematico ed efficace. L'obiettivo principale è quello di ottenere una visione complessiva dell'organizzazione e della pianificazione delle attività, valutando in modo oggettivo sia la copertura della capacità operativa delle risorse, sia l'avanzamento dei progetti in corso. Inoltre, la necessità di monitorare con regolarità la distribuzione delle attività rispetto alle risorse disponibili ha evidenziato la criticità di evitare situazioni di sopra o sotto allocazione, garantendo così un utilizzo ottimale delle competenze all'interno del team.

In azienda, il processo di gestione dei progetti si articola in diverse fasi. Esse vanno

dall'analisi di fattibilità e pianificazione iniziale fino all'esecuzione e al completamento delle attività. Per ogni progetto, si definiscono un budget, requisiti più o meno dettagliati e tempistiche specifiche. La pianificazione prevede l'assegnazione delle risorse più adeguate in termini di competenze e disponibilità, per garantire un corretto avanzamento nelle varie fasi operative. Durante l'esecuzione, è fondamentale mantenere un monitoraggio costante delle attività, con particolare attenzione al rispetto delle stime iniziali e all'eventuale gestione tempestiva di problematiche o ritardi.

Qui di seguito uno schema sintetico:

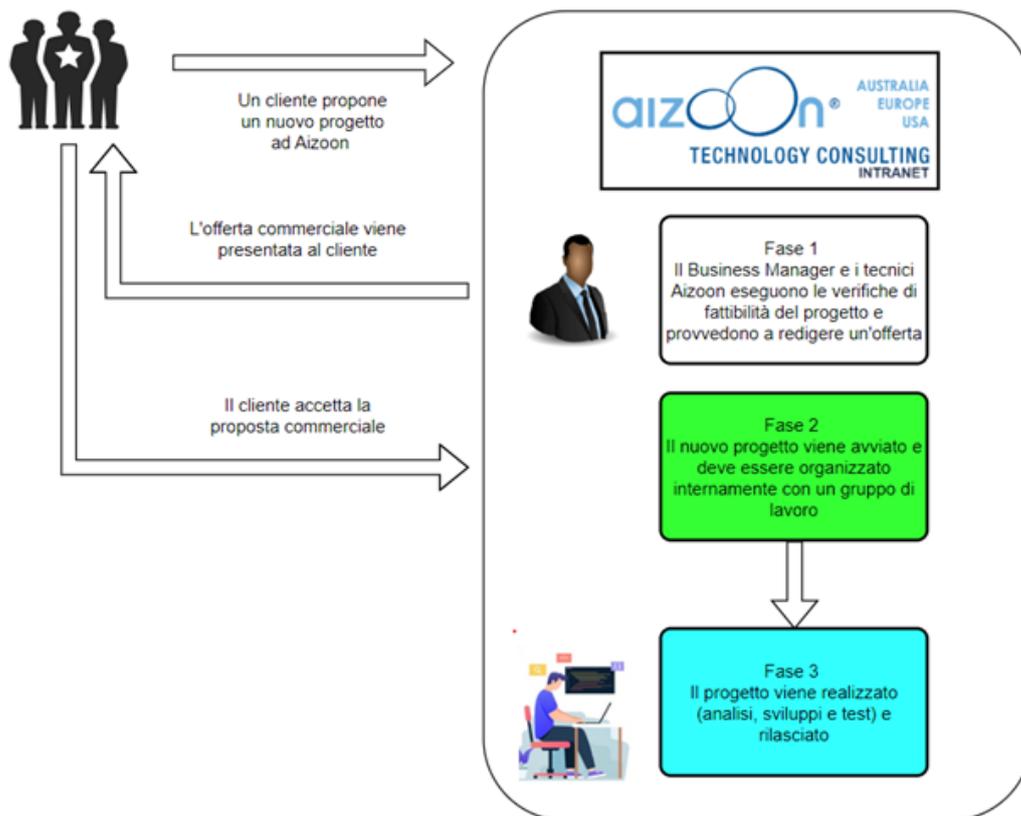


Figura 3.1. Fasi Gestione Progetti

Attualmente, viene utilizzato Azure DevOps come principale strumento di supporto per il censimento dei progetti e il monitoraggio delle attività. Tale piattaforma, tuttavia, risulta limitata in quanto copre principalmente la gestione della fase operativa (Fase 3 del processo). L'assenza di un'integrazione che permetta una visione complessiva delle risorse e delle competenze disponibili nel gruppo rappresenta una criticità, soprattutto durante le fasi di offerta e pianificazione (Fase 1 e Fase 2).

Il progetto nasce quindi con l'obiettivo di superare questi limiti, sviluppando uno strumento centralizzato in grado di:

- Fornire una visione unificata e completa dei progetti in carico al gruppo e delle risorse disponibili, con evidenza delle loro competenze specifiche.
- Supportare la pianificazione, valutando in tempo reale la capacità operativa e le competenze richieste per nuovi progetti.
- Monitorare in modo preciso l'avanzamento delle attività durante l'esecuzione dei progetti, mettendo in relazione i tempi effettivi con le stime iniziali per prevenire eventuali criticità.

Per l'implementazione del progetto, è stata utilizzata Microsoft Power Platform, sfruttando le licenze Microsoft già in possesso dell'azienda. Questo approccio ha permesso di creare una soluzione integrata e scalabile. Lo sviluppo si è limitato alla fase di gestione dell'offerta (Fase 1) e al principio della fase di pianificazione (Fase 2), ponendo le basi per un'evoluzione futura verso una gestione completa delle attività e delle risorse e del monitoraggio dei progetti.

Nelle sezioni successive, verranno analizzate le informazioni fondamentali da raccogliere per le entità "Progetto" e "Risorsa", che rappresentano i pilastri della soluzione proposta.

3.2 Liste Microsoft Sharepoint: Entità Progetto ed Entità Risorsa

3.2.1 Entità Progetto

L'entità Progetto contiene l'insieme dei progetti, sia quelli potenziali, definiti come opportunità, che quelli in corso di realizzazione.

Il progetto deve poter essere censito già nel momento della ricezione della proposta da parte del cliente e le informazioni, che inizialmente saranno minimali, dovranno essere integrate con tutti i dettagli necessari per il monitoraggio e l'avanzamento di progetto durante la fase di pianificazione e realizzazione. Per la memorizzazione e la gestione delle informazioni chiave relative a quest'entità, si è optato per l'utilizzo delle liste di Microsoft SharePoint. SharePoint è una piattaforma collaborativa offerta da Microsoft, progettata per supportare la gestione documentale, la condivisione di informazioni e la collaborazione tra team all'interno di un'organizzazione. Grazie alla sua flessibilità e all'integrazione con le applicazioni di Microsoft 365, come Teams e One Drive, SharePoint consente di centralizzare i dati e semplificare i processi di lavoro.

Le *Liste* di SharePoint rappresentano un elemento fondamentale per strutturare e organizzare i dati. Si tratta di contenitori simili a tabelle di database che permettono di archiviare informazioni in formato strutturato, utilizzando colonne che rappresentano i campi ed elementi che costituiscono i singoli record.

Le liste offrono funzionalità avanzate, come la personalizzazione delle colonne, la gestione delle versioni e l'integrazione con altre piattaforme per automatizzare flussi di lavoro (utilizzando Power Automate).

In questo contesto, ogni progetto viene rappresentato come un elemento all'interno della lista, con attributi chiave, che possono essere rapidamente aggiornati e monitorati.

Questa soluzione ha reso possibile non solo una gestione efficiente delle informazioni legate ai progetti, ma anche una base solida per eventuali integrazioni future con altri strumenti Microsoft, supportando la necessità di evolvere verso un sistema sempre più integrato e completo.

I campi principali da popolare, memorizzati nella lista sono:

- *Nome-Progetto*: identificativo dell'entità.
- *Descrizione-sintetica*: descrizione sintetica del progetto.
- *Descrizione-dettagliata*: descrizione di dettaglio del progetto.
- *Cliente*: ragione sociale del cliente a cui fa riferimento il progetto.
- *Owner-cliente*: nominativo del riferimento di progetto presso il cliente.
- *Owner-interno*: nominativo del riferimento interno.
- *Data-inizio*: data stimata di inizio in fase di proposta del cliente.
- *Data-limite-presentazione*: data limite per la presentazione dell'offerta al cliente.
- *Data-kick-off*: data di inizio del progetto effettiva, definita in fase di registrazione esito trattative.
- *Data-rilascio*: data in cui il progetto termina e si procede con il rilascio.
- *Data-fine-garanzia*: data successiva al rilascio per garantire l'assistenza di quanto rilasciato al cliente; entro questa data è necessario garantire supporto per eventuali bug.
- *Stato-avanzamento*: indica lo stato dell'opportunità, dell'offerta o del progetto in un determinato momento per definire le azioni necessarie per monitorare efficacemente l'attività.

Fase 1

- Opportunità in valutazione
- Opportunità rifiutata
- Offerta in preparazione
- Offerta inviata al cliente
- Offerta declinata

Fase 2

- Progetto da avviare
- Progetto avviato

Fase 3

- Progetto in analisi
- Progetto in sviluppo
- Progetto in System Test
- Progetto in UAT
- Progetto in garanzia
- Progetto chiuso

- *Costo-uomo*: valore medio di una giornata di lavoro per le risorse coinvolte nel progetto.
- *Budget-stimato*: budget stimato in fase di offerta.
- *Budget-finale*: accordato dal cliente all'avvio del progetto.
- *Marginalità*: percentuale di marginalità trattenuta sul budget di progetto.
- *Tipologia-entità*: distingue le entità in opportunità, offerte e progetti.

3.2.2 Entità Risorsa

Per poter pianificare i progetti e relative attività è indispensabile avere a disposizione l'elenco delle persone facenti parte del gruppo di lavoro con l'evidenza del ruolo, le competenze e la disponibilità.

Di seguito i campi ritenuti necessari per poter avere una visione di insieme delle risorse disponibili nel gruppo:

- *Identificativo-risorsa*: nominativo della risorsa.
- *Ruolo*: profilo della risorsa (es. Senior Developer, Middle Developer, Junior Developer, Functional Analyst, ecc.).
- *Fattore*: costo giornaliero modificabile della risorsa, caratterizzato da un periodo di validità.
- *Numero-progetti-a-carico*: numero di progetti a carico della risorsa.
- *Progetti-a-carico*: elenco dei progetti a carico della risorsa.
- *Owner-interno*: nominativo del riferimento interno.
- *Competenze*: competenze tecniche della risorsa (es. Java, Analyst, Low-code, UX Design, PHP, ecc.).

Oltre al censimento delle persone fisiche che lavorano all'interno del gruppo di lavoro, deve essere possibile censire anche delle risorse fittizie, per elaborare il *Capacity Plan Preventivo* durante la fase di offerta, quando non sono ancora definite le persone effettive che seguiranno il progetto.

Il censimento avviene popolando i seguenti campi di una lista "Risorse Fittizie":

- *Identificativo-risorsa*: (es. Analista 1, Sviluppatore Java1, Sviluppatore Java 2).

- *Ruolo*: profilo della risorsa (es. Senior Developer, Middle Developer, Junior Developer, Functional Analyst, ecc.).
- *Costo-giornaliero*: costo giornaliero stimato per la tipologia di risorsa.
- *Opportunità*: identificativo dell'opportunità in cui verrà impiegata.
- *Data-inizio*: data stimata di inizio impiego nell'opportunità.
- *Data-fine*: data stimata di fine impiego nell'opportunità.

Nel sottocapitolo successivo viene presentata una descrizione dettagliata del *Tool*, sviluppato personalmente, per la gestione integrata dei progetti e delle relative risorse attraverso le diverse fasi del loro ciclo di vita.

Lo sviluppo è stato realizzato sfruttando in sinergia i principali applicativi della Microsoft Power Platform, descritti nel capitolo precedente: Power Automate, Power Apps e Power BI.

3.3 Gestione Entità Progetto

3.3.1 Ciclo di vita dell'Entità

Nel ciclo di vita dell'entità "Progetto", essa assume denominazioni differenti in funzione della fase del processo che attraversa, riflettendo l'evoluzione delle attività e degli stati decisionali che la caratterizzano. Inizialmente, dal momento in cui il cliente esprime una richiesta, l'entità viene definita come "opportunità". Questo termine indica una fase preliminare in cui il responsabile interno stabilito, insieme ai tecnici competenti, valuta la fattibilità tecnica, economica e organizzativa della proposta.

Una volta completata questa analisi e qualora la richiesta venga considerata fattibile, l'entità cambia stato e viene denominata "offerta". In questa fase, viene

formalizzata una proposta dettagliata da sottoporre al cliente, includendo requisiti, tempistiche, costi e modalità operative.

Infine, qualora l'offerta venga accettata e le trattative con il cliente abbiano esito positivo, l'entità assume lo stato definitivo di "progetto". A questo punto il progetto è pronto per essere avviato ed è possibile in un secondo momento strutturare il lavoro in task, assegnare le risorse necessarie, definire le tempistiche e monitorare il progresso rispetto agli obiettivi definiti.

Questa transizione graduale da opportunità a offerta e, infine, a progetto, garantisce un monitoraggio chiaro e strutturato di ogni iniziativa, assicurando che ogni fase sia accuratamente supportata da decisioni consapevoli sullo stato del processo di censimento.

Si è deciso di strutturare il ciclo di vita dell'entità "Progetto" in cinque fasi principali, ognuna delle quali rappresenta uno stato chiave del processo di censimento, che accompagna l'entità dalla fase di proposta fino alla gestione operativa vera e propria. Questa suddivisione consente di organizzare e monitorare in modo sistematico l'evoluzione di ogni progetto, permettendo al team di progetto di tenere traccia dello stato di avanzamento, delle informazioni disponibili, delle approvazioni necessarie e delle risorse coinvolte.

1. *Inserimento di una Nuova Proposta*

La prima fase inizia con l'inserimento di una nuova proposta, che rappresenta la registrazione di un'opportunità presentata dal cliente. In questa fase, vengono raccolte le informazioni iniziali relative alla richiesta del cliente, quali la descrizione del progetto potenziale e il mercato di riferimento. La proposta è formalmente inserita nel sistema come punto di partenza per ulteriori valutazioni e analisi.

2. *Valutazione della Fattibilità*

Una volta registrata la proposta, si procede alla valutazione della fattibilità.

Questa fase è cruciale per determinare se l'opportunità può trasformarsi in un progetto realizzabile. La valutazione viene condotta dal responsabile interno, che analizza diversi aspetti tecnici, economici e organizzativi. Durante questo processo, si decide se procedere con la definizione dell'offerta oppure se l'opportunità non è praticabile.

3. *Caricamento dei Dettagli dell'Offerta*

Se la proposta supera la fase di valutazione, viene considerata fattibile e si passa alla fase di caricamento dei dettagli dell'offerta, come il budget stimato, il costo uomo e le tempistiche di realizzazione. Qui l'entità diventa ufficialmente un'offerta, e si lavora parallelamente alla stesura di un documento, da allegare in fase di caricamento, che include i requisiti, le funzionalità richieste, le tempistiche, i costi e le tecnologie da utilizzare. L'offerta rappresenta un passaggio formale e strutturato che sancisce l'impegno dell'organizzazione verso il cliente.

4. *Registrazione dell'Esito delle Trattative*

Conclusa la fase di offerta, si aprono le trattative con il cliente. Durante questa fase, si discute la proposta presentata per raggiungere un accordo. L'esito delle trattative viene quindi registrato nel sistema: un esito positivo trasforma l'entità in un progetto effettivo, pronto per essere avviato, mentre un esito negativo porta alla chiusura dell'opportunità.

5. *Gestione Progetti*

Con l'approvazione dell'offerta e la trasformazione in progetto, si entra nella fase operativa. In questa fase il progetto viene avviato e le risorse vengono assegnate in base alle competenze e disponibilità ai task definiti per ogni progetto. Il progetto viene quindi seguito fino alla sua conclusione, aggiornando coerentemente lo stato, previa autorizzazione dell'*owner* interno di progetto, ogniqualvolta vengono caricate le informazioni necessarie.

Tali fasi, oggetto delle successive sezioni dell'elaborato, sono state gestite attraverso lo sviluppo di un'unica applicazione su Power Apps, sfruttando l'integrazione nativa con i flussi di Power Automate, per automatizzare i processi ed integrare contemporaneamente i diversi applicativi Microsoft coinvolti.

L'home page dell'applicazione, mostrata qui di seguito, permette attraverso una barra di navigazione di spostarsi tra le diverse pagine, percorrendo logicamente l'intero ciclo di vita dell'entità "Progetto".



Figura 3.2. Home Page Applicazione

3.3.2 Inserimento di Una Nuova Proposta e Valutazione Fattibilità

Il processo di censimento di una nuova proposta inizia quando un cliente comunica all'azienda i dettagli preliminari relativi alla proposta.

Queste informazioni sono essenziali per creare la base della proposta e servono come input per le fasi successive.

Successivamente, un responsabile interno prende in carico queste informazioni e le inserisce nel sistema compilando un form di proposta tramite una pagina specifica (denominata "Opportunità") dell'applicazione su Power Apps.

Questo passaggio garantisce che le informazioni iniziali siano correttamente documentate e formalizzate in modo strutturato.

Dopo aver compilato ed inviato il form, dapprima viene inviata una notifica dell'inserimento della nuova proposta sul canale Teams del team di progetto e contestualmente, una richiesta di approvazione, con i relativi dettagli, viene automaticamente inoltrata al responsabile interno individuato, utilizzando il sistema di approvazioni integrato in Microsoft Teams, Approvals.

In questa fase, viene richiesto al responsabile di effettuare una valutazione della fattibilità e notificare l'esito tramite la richiesta ricevuta in *Approvals*. Questa richiesta, automatizzata sfruttando un flusso su Power Automate, assicura che ogni opportunità sia sottoposta a una revisione e approvazione formale prima di procedere effettivamente con la preparazione dell'offerta, in caso di esito positivo.

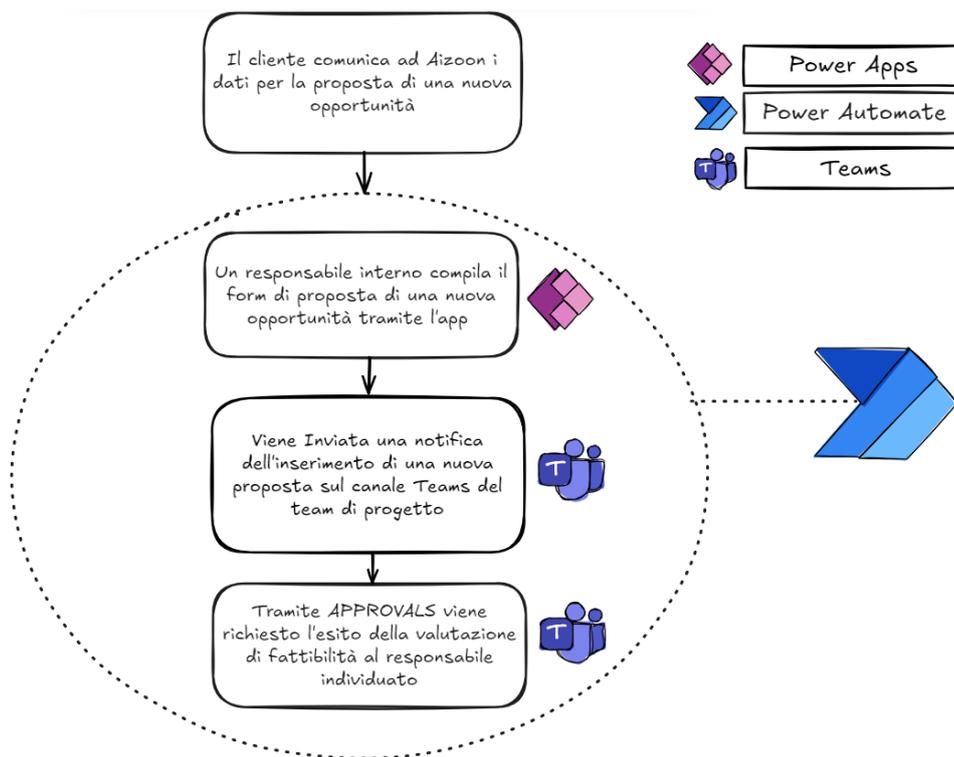


Figura 3.3. Diagramma Processo Inserimento Nuova Proposta

Qualora la proposta venga considerata fattibile, lo stato dell'entità, viene aggiornato automaticamente a "Offerta in Preparazione". Questa transizione segna l'inizio della fase operativa, in cui il team coinvolto viene notificato attraverso una comunicazione su Teams, assicurando che tutte le persone interessate siano informate e possano iniziare a lavorare sulla preparazione dell'offerta per il cliente. Questo meccanismo non solo velocizza il passaggio alla fase successiva, ma garantisce anche una sincronizzazione immediata tra i membri del team.

Nel caso in cui la proposta venga ritenuta non fattibile, lo stato dell'entità viene aggiornato a "Opportunità Rifiutata", e alcuni campi non più necessari vengono eliminati, mantenendo solo le informazioni essenziali, notificando automaticamente, anche in questo caso, la decisione sul canale Teams.

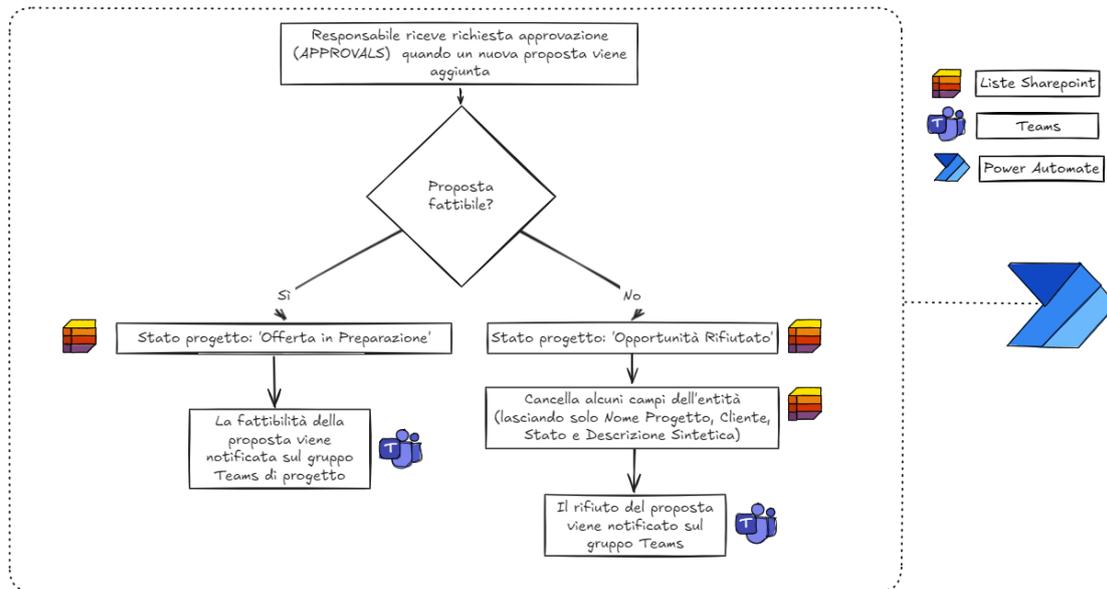


Figura 3.4. Diagramma Processo Valutazione Fattibilità

Pagina "Opportunità"

Figura 3.5. Pagina "Opportunità" dell'Applicazione Power Apps

Il form per il censimento di una nuova proposta è inserito all'interno della pagina

“Opportunità” dell’applicazione.

Per strutturare il form è stato necessario trascinare il componente nello *Screen*, sfruttando la funzionalità *drag-and-drop* e successivamente selezionare l’origine dati di riferimento per la selezione dei campi da compilare.

In questo caso, dopo aver collegato il form alla lista Sharepoint “Progetto”, Power Apps permette di selezionare i campi da mostrare all’utente, tra quelli definiti originariamente nella lista.

Per il censimento dell’opportunità sono stati definiti i seguenti campi, ciascuno caratterizzato da un tipo specifico, corrispondente al tipo di dato da gestire:

- *Nome*: *Text Input* in cui definire l’identificativo dell’opportunità.
- *Descrizione Sintetica*: *Text Input* (con una singola linea di testo).
- *Cliente*: *Text Input* in cui riportare la ragione sociale del cliente proponente.
- *Descrizione Dettagliata*: *Text Input* (multilinea).
- *Tipologia Opportunità*: *Dropdown*, popolato con un set di valori predefinito, specificato nella lista: Gara, Gara Pubblica, Gara AMS.
- *Mercato*: *Dropdown* per selezionare il mercato di riferimento.
- *Date*: controlli *Date Picker* per i campi "Data Inizio", "Data Limite Presentazione" e "Data Fine".

Per gestire la compilazione e l’invio del form sono stati inseriti i seguenti pulsanti, le cui azioni, definite dalle funzioni specificate nella *Formula Bar* di Power Apps, vengono richiamate alla selezione del pulsante (proprietà *Onselect*):

- “Cancella Campi”: permette di resettare il form allo stato di partenza, sfruttando la funzione *ResetForm(NomeForm)*.

- o “Carica Nuova Opportunità”: dopo aver effettuato un controllo sulla corretta compilazione dei campi richiesti, invia il form compilato con la funzione *SubmitForm(NomeForm)*, porta l’utente alla pagina che notifica il successo dell’operazione (funzione *Navigate(NomeScreen)*), resettando il form per una nuova compilazione.

```
If(IsBlank(DataCardValue3_1.Value) Or IsBlank(DataCardValue4_1.Value) Or IsBlank(DataCardValue8_1.Value) Or IsBlank(DataCardValue11_1.Value) Or IsBlank(DataCardValue12_1.SelectedDate) Or IsBlank(DataCardValue14.SelectedDate) Or IsBlank(DataCardValue15.SelectedDate) Or IsBlank(DataCardValue20.Selected) Or IsBlank(DataCardValue9.Selected);Notify("E' NECESSARIO COMPILARE TUTTI I CAMPI OBBLIGATORI!";NotificationType.Warning);  
SubmitForm(Form1);;  
Navigate(SUCCESSO_3);;  
ResetForm(Form1);;
```

Figura 3.6. Proprietà *OnSelect* pulsante “Carica Nuova Opportunità”

Flusso Power Automate per la Valutazione Fattibilità



Figura 3.7. Flusso Valutazione Fattibilità 1

Il flusso si attiva automaticamente "Quando viene creato un elemento", indicando che l'evento scatenante è l'inserimento di una nuova opportunità nella lista "Progetto" su SharePoint. Questo evento avvia l'intero processo automatizzato.

Dopo l'attivazione, il flusso esegue l'azione "Pubblica messaggio in una chat o un canale".

Questo step è configurato per inviare una notifica al team di progetto, su Microsoft Teams, informandolo dell'aggiunta della nuova opportunità, specificandone l'identificativo e il cliente di riferimento, dettagli estraibili dall'azione precedente, che sfruttando il connettore a Sharepoint, permette di accedere a tutti i campi dell'elemento appena creato, rendendoli disponibili per le azioni successive del flusso.

Il passo successivo è l'azione "Avvia e attendi un'approvazione", che coinvolge un approvatore designato nel processo decisionale, ovvero il responsabile di progetto.

All'interno della richiesta di approvazione su Microsoft Approvals, il responsabile può accedere tramite link (estraibile dall'azione iniziale del flusso) all'elemento Sharepoint appena creato, per visionare le informazioni preliminari disponibili e prendere una decisione circa la fattibilità della proposta, selezionando il pulsante corrispondente ("Approva" o "Rifiuta").

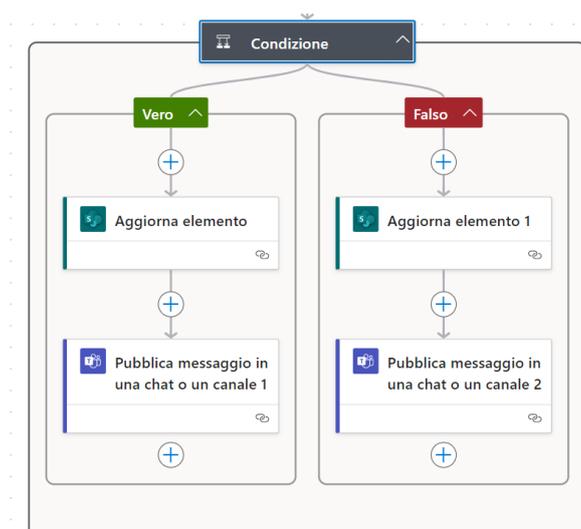


Figura 3.8. Flusso Valutazione Fattibilità 2

Una volta completata l'approvazione, il flusso passa a una struttura condizionale basata sull'esito della decisione. Qui vengono definite due diramazioni principali:

a) *Esito positivo ("Vero")*

- *Aggiorna elemento*: in caso di approvazione, l'elemento creato inizialmente viene aggiornato, modificando lo stato in "Offerta in Preparazione" e la tipologia dell'entità in "Offerta".
- *Pubblica messaggio in una chat o un canale 1*: viene inviata una notifica nel canale Teams dedicato, per informare gli interessati che l'approvazione è stata concessa.

b) *Esito negativo ("Falso")*

- *Aggiorna elemento 1*: in caso di rifiuto, lo stato viene aggiornato a "Opportunità Rifiutata" e vengono resettati i campi precedentemente compilati, che risultano poco significativi alla luce dell'esito negativo del processo di approvazione ("Descrizione Dettagliata", "Data Inizio", "Data Fine").
- *Pubblica messaggio in una chat o un canale 2*: viene inviata una notifica nel medesimo canale Teams, segnalando l'esito negativo dell'approvazione e specificando la necessità di informare il cliente.

Publica messaggio in una chat o un canale 2

Parametri Impostazioni Visualizzazione codice Testing Informazioni su

Team *

TEST

Canale *

Prova

Messaggio *

Normal Arial 15px B I U A E ↺ </>

L'Opportunità [Nome Progetto x] è stata valutata come non fattibile.
E' necessario comunicare a l'esito della valutazione a [Cliente x]

Figura 3.9. Azione "Pubblica Messaggio su Canale Teams" per notificare rifiuto Opportunità

3.3.3 Caricamento Offerta

Il processo di caricamento dei dettagli dell'offerta viene avviato da un membro del team di progetto che compila il modulo di offerta nella pagina "Caricamento Offerta" dell'applicazione. I campi inseriti, come il budget, il costo uomo e la marginalità, vengono automaticamente aggiornati sulla lista SharePoint, dove sono archiviati i campi relativi all'entità.

Successivamente, una notifica viene pubblicata sul canale Teams dedicato al team di progetto, informando i membri dell'avvenuta compilazione dell'offerta. A questo punto, il processo prevede una fase decisionale in cui il responsabile verifica e decide se confermare l'invio dell'offerta tramite Microsoft Approvals.

Se il responsabile decide di non confermare l'invio, lo stato dell'offerta rimane impostato su "Offerta In Preparazione", un messaggio viene pubblicato su Teams per segnalare la necessità di ulteriori modifiche, includendo dettagli come il cliente e

l'identificativo dell'entità.

In caso contrario, se il responsabile conferma l'invio, lo stato viene aggiornato automaticamente a "Offerta Inviata", con la relativa notifica riguardante l'esito dell'approvazione.

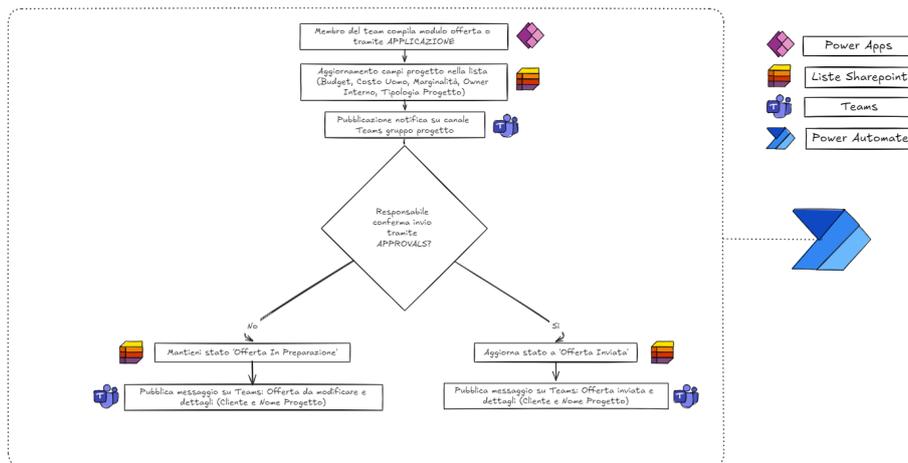


Figura 3.10. Diagramma Processo Caricamento Offerta

Pagina "Caricamento Offerta"

Lo screenshot mostra l'interfaccia utente della pagina "Caricamento Offerta". In alto, la barra di navigazione include "Home", "Opportunità/Offerte", "Opportunità", "Caricamento Offerta" (selezionato), "Registra Esito Trattative" e "Gestione Progetti". A sinistra, c'è un pannello di ricerca con "Cerca Opportunità" e un risultato per "Applicativo Cauzioni". Il campo principale è un modulo con i seguenti campi: "Nome" (Applicativo Cauzioni), "Tipologia di Progetto" (menu a tendina), "Owner Interno" (menu a tendina), "Budget Stimato" (campo di testo), "Costo Uomo" (campo di testo), "Marginalità" (campo di testo), "Data Kick-Off" (31 dicembre 2001), "Data Rilascio" (31 dicembre 2001), "Note Aggiuntive Offerta" (area di testo) e "Allegati" (Non sono presenti allegati). In basso a destra, ci sono i pulsanti "Cancella Campi" e "Aggiorna Dettagli Offerta".

Figura 3.11. Pagina "Caricamento Offerta" dell'Applicazione Power Apps

La sezione centrale della pagina è suddivisa in due pannelli principali. A sinistra, una tabella collegata alla lista Sharepoint “Progetto”, che mostra un elenco di opportunità disponibili, con la possibilità di effettuare una ricerca tramite un *ComboBox*. La ricerca è limitata alle Opportunità nello stato “Offerta in Preparazione”, utilizzando la funzione *Filter*, a cui passare come parametri l’origine dati (lista) e la colonna da filtrare (stato di avanzamento).

A destra si trova il modulo di caricamento dei dettagli dell’offerta, legato alla specifica opportunità selezionata nell’elenco dall’utente. Questo modulo è composto da una serie di campi strutturati per acquisire le informazioni chiave relative all’offerta:

- *Nome*: campo per il nome del progetto, in questo caso già compilato con "Applicativo Cauzioni".
- *Tipologia di Progetto*: un menu a discesa per specificare la categoria di contratto (TimeMaterial, Fixed Price...).
- *Owner Interno*: per selezionare il responsabile interno del progetto, tra i dipendenti dell’organizzazione.
- *Budget Stimato*: per inserire il valore economico stimato per il progetto.
- *Costo Uomo*: per indicare il costo per unità lavorativa.
- *Marginalità*: campo dedicato alla percentuale o al valore di marginalità associata al progetto.
- *Data Kick-Off*: selezione della data per l’inizio previsto delle attività del progetto.
- *Data Rilascio*: selezione della data stimata per il rilascio del progetto.
- *Note Aggiuntive Offerta*: una casella di testo libera per inserire eventuali commenti o informazioni supplementari.

- *Allegati*: campo per allegare all’offerta il *Calculation Model*, ovvero l’Excel con i dettagli relativi al calcolo dei costi stimati e delle risorse “fittizie” coinvolte per strutturare il *Capacity Plan Preventivo*.

Nella parte inferiore del modulo sono presenti i due pulsanti principali:

- “Cancella Campi”: per svuotare i dati già inseriti nei campi del modulo.
- “Aggiorna Dettagli Offerta”: per salvare o aggiornare le informazioni dell’offerta inserite.

Selezionando il secondo pulsante, il form viene resettato, l’elenco delle opportunità viene aggiornato e, soprattutto, sfruttando la funzione *Flusso.Run(Variabile)* viene innescato il corrispettivo flusso su Power Automate per la gestione del processo di caricamento offerta, precedentemente aggiunto nel pannello dedicato su Power Apps.

Flusso Power Automate per Caricamento ed Invio Offerta



Figura 3.12. Flusso Caricamento e Conferma Invio Offerta

Il flusso si attiva automaticamente "Quando Power Apps chiama un flusso", ovvero quando il flusso viene richiamato dall'applicazione sfruttando la funzione *Run (Variabile)*. In questo caso, viene passato come parametro della funzione una variabile, che memorizza l'id dell'elemento della lista "Progetto" selezionato nella pagina dell'applicazione per la compilazione del form di offerta.

Il trigger utilizzato, infatti, permette di specificare dei parametri di input, estratti dall'applicazione, da poter utilizzare nelle azioni successive del flusso.

L'id dell'elemento della lista risulta fondamentale per poter selezionare tale elemento, utilizzando l'azione Sharepoint "Ottieni Elemento", che richiede, infatti, tale informazione come parametro.

Tale azione, consente di estrarre i campi dell'elemento della lista per poter dettagliare la notifica sul canale Teams dell'avvenuto caricamento dei dettagli dell'offerta e la relativa richiesta di approvazione, con cui il responsabile può confermare l'invio dell'offerta al cliente o richiedere ulteriori modifiche.

In seguito alla ricezione dell'esito della richiesta di conferma di invio dell'offerta, il flusso prosegue con una struttura condizionale simile a quella mostrata per il processo precedente, in cui lo stato di avanzamento dell'entità passa ad "Offerta Inviata", in caso di esito positivo, mentre rimane invariato in "Offerta in preparazione" in caso contrario.

In entrambi i casi, il team di progetto viene informato con una apposita notifica su Microsoft Teams.

3.3.4 Registrazione Esito Trattative

Le trattative con il cliente proseguono esternamente al *Tool* e il processo di registrazione dell'esito si sviluppa all'interno della pagina "Registra Esito Trattative" dell'applicazione. Dopo aver confermato il raggiungimento di un accordo con il cliente, l'utente procede con la compilazione di un modulo, che formalizza la decisione finale sull'offerta presentata al cliente. Tale modulo consente di aggiornare i

dettagli definitivi dell'offerta accettata dal cliente.

Contestualmente, lo stato dell'entità associata viene modificato in "Progetto da Avviare", segnalando il passaggio alla fase operativa. A questo punto, viene creato un progetto su Azure DevOps (piattaforma Microsoft per la gestione dei progetti di sviluppo software), con due *Epic* (macroattività) che definiscono l'inizio e la fine del progetto e assegnando all'owner interno il ruolo di amministratore, integrandolo nel team di lavoro.

Successivamente, viene pubblicata una notifica sul gruppo Teams dedicato, che comunica che il progetto è stato aggiunto su Azure DevOps, con il relativo amministratore ed è pronto per essere avviato.

Nel caso in cui non venga raggiunto un accordo con il cliente, il processo segue un percorso alternativo. L'entità viene contrassegnata con lo stato "Offerta Declinata" e alcuni campi del progetto vengono cancellati, lasciando visibili solo le informazioni essenziali come il nome del progetto, il cliente e una breve descrizione.

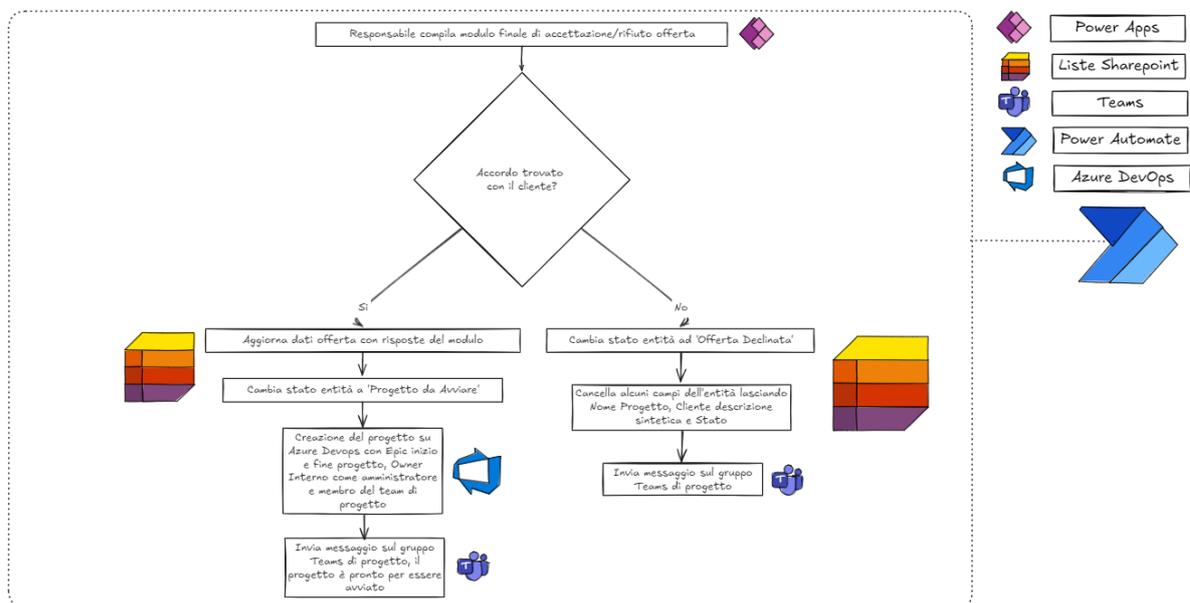


Figura 3.13. Diagramma Processo Registrazione Esito Trattative

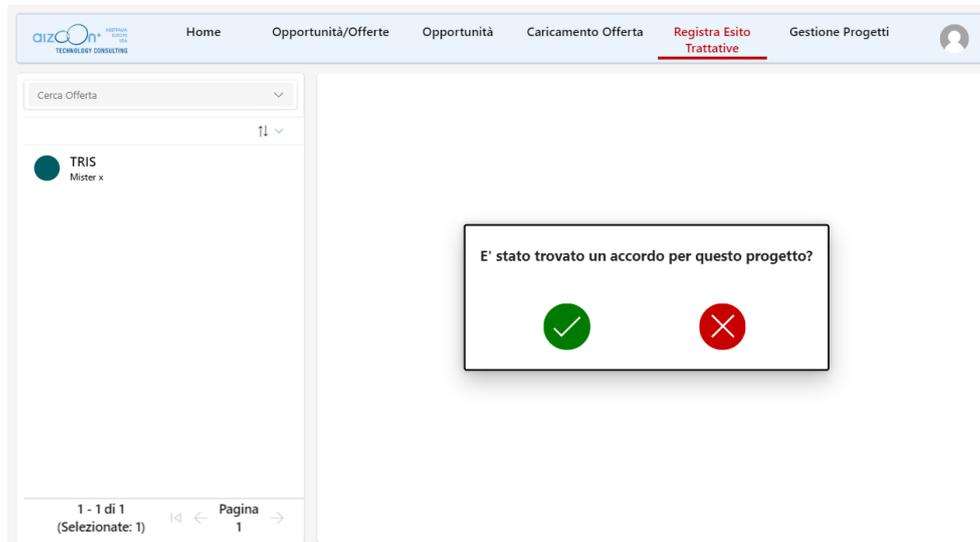
Pagina “Registra Esito Trattative”

Figura 3.14. Pagina “Registra Esito Trattative” Step 1

La struttura della pagina, similmente alla pagina “Caricamento Offerta” vede la suddivisione in due sezioni principali: a sinistra l’elenco delle offerte tra cui effettuare la selezione, con la possibilità di effettuare una ricerca tramite *ComboBox*, questa volta filtrando le entità in stato “Offerta Inviata”; a destra la sezione mostra temporaneamente un popup, visibile solo selezionando l’offerta dall’elenco (condizione definita nella proprietà *Visible* dell’elemento), che richiede l’esito delle trattative con il cliente.

a) *Accordo Non Trovato*

Nell’eventualità in cui le trattative non portino ad un accordo il popup richiede la conferma del rifiuto dell’offerta, come mostrato qui di seguito.

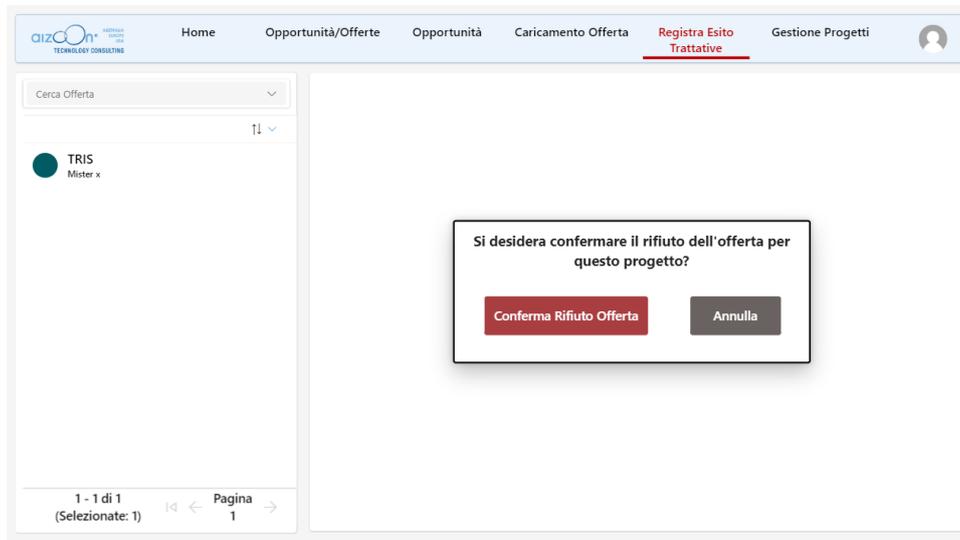


Figura 3.15. Conferma Rifiuto Offerta

La selezione del pulsante di conferma innesca un flusso su Power Automate, che esegue le azioni descritte in precedenza in caso di accordo non raggiunto tra le parti.

b) *Accordo Trovato* Qualora venga confermato il raggiungimento di un accordo

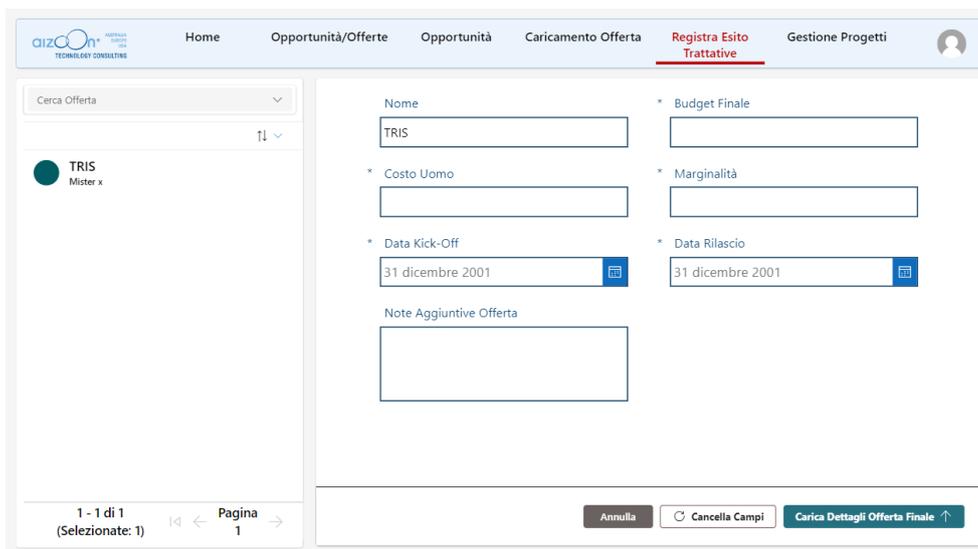


Figura 3.16. Caricamento Dettagli Offerta Finale

con il cliente, la sezione destra della pagina mostra il form di compilazione dei dettagli dell’offerta finale accettata dal cliente, simile a quello mostrato in fase di caricamento offerta, con l’unica differenza del campo “Budget Finale” in luogo del “Budget Stimato”.

Nella sezione inferiore della pagina è presente, oltre ai pulsanti “Cancella Campi” e “Carica Dettagli Offerta Finale”, il cui comportamento è analogo ai pulsanti presenti nelle pagine descritte in precedenza, il pulsante “Annulla”, il quale permette di mostrare nuovamente il popup riguardante l’esito delle trattative, consentendo la modifica della selezione in caso di errore.

La selezione del pulsante “Carica Dettagli Offerta Finale” avvia il relativo flusso su Power Automate, sfruttando nuovamente la funzione *Run (IdElemento)*, passando come parametro l’id dell’elemento selezionato.

Flusso Power Automate per Aggiornamento Offerta ed Inserimento Progetto su Azure DevOps

Il flusso risulta molto esteso, viste le difficoltà d’integrazione con Azure DevOps e può essere strutturato in 3 fasi distinte:

1. *Aggiornamento Stato e creazione progetto su DevOps.*
2. *Aggiunta Owner Interno come Project Administrator e membro del Default Team di progetto.*
3. *Inserimento Epic di inizio e rilascio del progetto.*

1. Aggiornamento Stato e creazione Progetto su DevOps



Figura 3.17. Flusso Aggiornamento Stato e creazione progetto DevOps

Il flusso viene chiamato contestualmente alla selezione del pulsante “Carica Dettagli Offerta Finale” nella Power App, mentre l’azione successiva “Aggiorna Elemento”, utilizzando l’id passato come parametro, aggiorna lo stato dell’elemento selezionato a “Progetto da Avviare”.

Vista l’assenza di un’azione nativa su Power Automate che permetta di aggiungere un nuovo progetto su Azure DevOps, è stata utilizzata l’azione che consente di inviare richieste HTTP ad Azure DevOps.

Qui di seguito viene mostrato il dettaglio dell’azione e i relativi parametri richiesti.

Invia una richiesta HTTP ad Azure DevOps

Parametri Impostazioni Visualizzazione codice Testing Informazioni su

Nome organizzazione *

TOOLS3

Metodo *

POST

URI relativo *

https://dev.azure.com/TOOLS3/_apis/projects?api-version=6.0

Parametri avanzati

Visualizzazione di 3 di 3

Mostra tutto Cancella tutto

Intestazioni

Authorization	Basic Om42emEyb2J2dGg0NGI3dno1anZ kemFwNnk3eWljYmw1dHY1NXhhd Ddub2szcDVuemc1NGE=	✕ 📄
Content-Type	application/json	✕

Figura 3.18. Aggiunta Progetto su *DevOps*

La sezione "Nome organizzazione" identifica l'ambiente specifico in cui verrà eseguita l'operazione, in questo caso l'organizzazione "TOOLS3". Questo parametro è essenziale poiché delimita il contesto operativo e garantisce che il progetto venga creato nel giusto spazio di lavoro.

Il metodo HTTP specifica il tipo di operazione che verrà eseguita sulla risorsa, in questo caso il metodo utilizzato è POST. Attraverso questo metodo, l'utente invia una richiesta che non modifica una risorsa esistente, ma ne genera una nuova all'interno della piattaforma Azure DevOps. L'URI relativo specificato punta direttamente all'endpoint progettato per gestire i progetti, utilizzando l'indirizzo `https://dev.azure.com/TOOLS3/_apis/projects?api-version=6.0`.

Questa URL rappresenta l'accesso all'API REST di Azure DevOps, con l'aggiunta della versione dell'API richiesta, in questo caso la 6.0, che assicura che la chiamata

sia conforme alle funzionalità più recenti disponibili per questa operazione.

L'autenticazione ricopre un ruolo fondamentale al fine di garantire che la richiesta sia autorizzata. In questo caso, il campo "Authorization" utilizza l'autenticazione Basic con una stringa codificata in Base64 che include un *token* di accesso personale temporaneo (PAT), generato all'interno del proprio account Azure DevOps. Questa stringa garantisce che solo utenti con i permessi corretti possano eseguire la richiesta, prevenendo accessi non autorizzati e mantenendo il sistema sicuro. La presenza dell'header "Content-Type" con il valore "application/json" indica che i dati inviati saranno formattati in JSON, un formato standard per l'interazione con API REST, garantendo così che Azure DevOps possa elaborare correttamente i parametri passati nel corpo della richiesta.

La struttura del corpo fornisce i parametri essenziali per definire il progetto che verrà generato. Al suo interno, il campo "name" corrisponde al nome del progetto, che viene dinamicamente valorizzato attraverso il parametro "Nome Progetto" ottenuto dall'azione precedente "Aggiorna Elemento". Allo stesso modo, il campo "description" rappresenta una descrizione sintetica del progetto, anch'essa inserita dinamicamente tramite il parametro "Descrizione Sintetica".

Il progetto viene creato con una visibilità impostata su "private", limitando l'accesso ai soli utenti autorizzati, garantendo così un livello di sicurezza adeguato per i contenuti del *repository* e della *pipeline*. All'interno della sezione "capabilities", viene specificato che il sistema di controllo versione sarà Git, una scelta ormai standard per la gestione del codice sorgente in ambienti *DevOps*.

Inoltre, la configurazione del progetto è determinata da un template di processo identificato da un "templateTypeId", che corrisponde a una particolare metodologia di gestione progetti già predefinita e associata a quel particolare identificativo univoco (*Waterfall, Agile...*).

Corpo

```

{
  "name": "Nome Progetto x",
  "description": "Descrizione Sintet... x",
  "visibility": "private",
  "capabilities": {
    "versioncontrol": {
      "sourceControlType": "Git"
    },
    "processTemplate": {
      "templateTypeid": "7ba7598f-57fd-486e-891e-4a5b5a3e3cb5"
    }
  }
}

```

Figura 3.19. Corpo Richiesta HTTP per Aggiunta Progetto

2. Aggiunta Onwer Interno come Project Administrator e membro del Default Team di progetto

Analogamente all'aggiunta di un nuovo progetto, non esiste al momento un'azione predefinita che permette di aggiungere un utente come amministratore o come membro del team di progetto, assegnandogli perciò i permessi derivanti in termini di modifica delle *Epic* e di aggiunta delle risorse.

L'unica strada percorribile si è rivelata essere una combinazione di richieste HTTP che permette, in primo luogo, di ottenere l'identificativo dei gruppi associati ai progetti caricati su *DevOps*, utilizzando il metodo GET e l'URI:

https://vssps.dev.azure.com/NomeOrganizzazione/_apis/graph/groups.

In questo modo è possibile ricercare, sfruttando un ciclo e una struttura condizionale, l'identificativo del gruppo dei *Project Administrators* e del team di progetto, creati contestualmente all'aggiunta del nuovo progetto su Azure DevOps.

In successione alla richiesta HTTP è stata utilizzata l'azione "Analizza JSON" per

poter analizzare il corpo della risposta (passato come parametro dell'azione, insieme al relativo schema), ottenuto dalla richiesta e utilizzare i relativi campi (come nome e id del gruppo) nelle azioni successive del flusso.

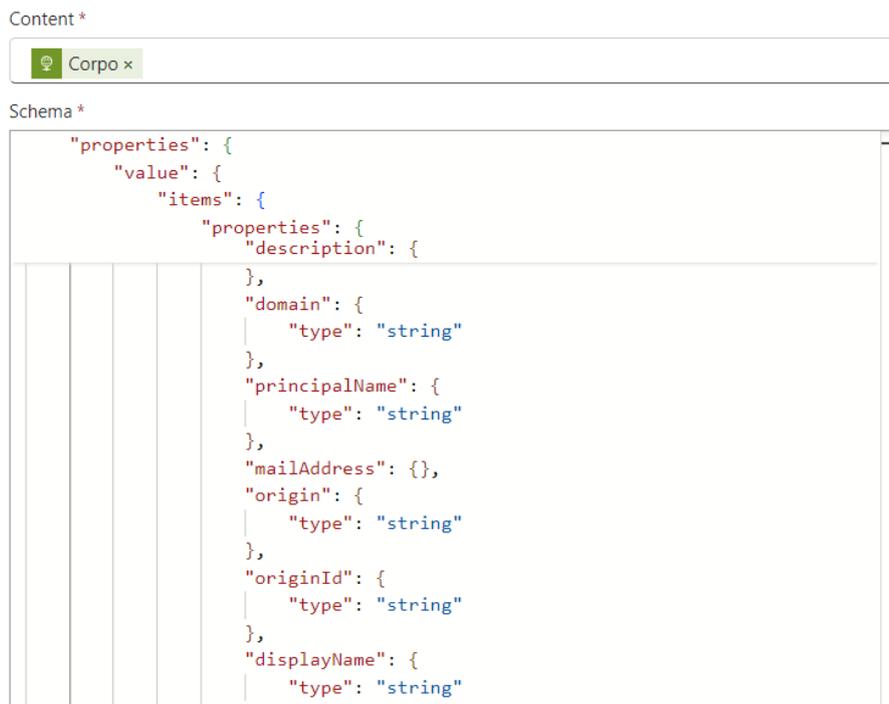


Figura 3.20. Esempio azione “Analizza JSON”

Come mostrato in figura, i campi descrittivi dei gruppi vengono salvati all'interno di un oggetto *item* e possono essere estratti ed utilizzati.

Dopo aver individuato e salvato, in variabili opportunamente definite, l'id e il “principalName” del gruppo degli amministratori e del team di default del progetto in questione, è necessario dapprima inserire l'*owner* interno designato come membro dell'organizzazione su Azure DevOps e in seguito, con una richiesta HTTP, estrarne l'identificativo. Tale azione, sfrutta nuovamente il metodo GET, l'autorizzazione tramite PAT e la formattazione in JSON, ma utilizza un'URI specifico per ottenere i membri:

https://vsaex.dev.azure.com/NomeOrganizzazione/_apis/memberentitlements?api-version=7.1-preview.2.

Tra i membri, per estrarre l'identificativo dell'amministratore da aggiungere, bisogna servirsi di un ciclo e di una condizione per verificare per ogni l'elemento la corrispondenza con l'*owner* definito nella lista Sharepoint.

Ottenuto l'id, è possibile inserire l'*owner* nel gruppo degli amministratori e nel *Default Team* con un'ulteriore richiesta, definita come riportato di seguito.

URI *

https://vsaex.dev.azure.com/TOOLS3/_apis/GroupEntitlements/Corpo//members/Bodyid?api-version=7.1

Method *

PUT

Headers

Authorization	Basic OnhucjU3YmcyaWp6anBxNTZ6ZHJpY Ww1NnVudmxiZ2h5bGdnZ2gyem5wb DY0ZTd6a25ucHE=
---------------	---

Queries

Content-type	application/json
api-version	7.1

Figura 3.21. Richiesta HTTP “Aggiunta Amministratore”

Il metodo adottato in questo caso è PUT, che indica che la richiesta mira a modificare o aggiungere risorse esistenti.

L'operazione avverrà all'interno dell'organizzazione TOOLS3 e interagirà con un gruppo identificato dinamicamente tramite il valore "Corpo", che rappresenta un parametro variabile, salvato in un'apposita variabile, derivante dalla prima richiesta HTTP, con la quale è stato possibile ottenere l'identificativo del gruppo degli amministratori.

Analogamente, all'interno dell'URI viene passato come parametro "Body id", ovvero l'id del membro dell'organizzazione, che diventerà amministratore.

Essendo il flusso innescato da un "Utente Tecnico" tramite l'applicazione su Power Apps, è stato necessario predisporre un'azione per poter rimuovere tale utente, aggiunto automaticamente all'aggiunta del progetto, dal gruppo amministratori e dal team di progetto, non essendo di fatto una risorsa effettiva.

La richiesta correlata sfrutta il metodo DELETE e l'identificativo dell'utente e del gruppo di riferimento come parametri dell'URI, estratti dalle richieste HTTP precedentemente descritte.

3. Inserimento Epic di inizio e rilascio del progetto.

Dopo aver ultimato l'inserimento dell'entità progetto su Azure DevOps, le azioni conclusive del flusso si concentrano sulla definizione delle *Epic* di inizio e fine progetto e sulla predisposizione della notifica sul canale Teams di progetto, che comunica l'aggiornamento dello stato a "Progetto da Avviare" e l'aggiunta dell'owner interno come amministratore.

Per quanto riguarda la creazione delle *Epic* all'interno del progetto appena aggiunto, è stata strutturata una richiesta HTTP, che utilizza il metodo POST e il seguente URI:

https://dev.azure.com/NomeOrganizzazione/NomeProgetto/_apis/wit/workitems.

Il corpo della richiesta, in formato JSON, è così definito:

```

{
  "op": "add",
  "path": "/fields/System.Title",
  "value": "Data Inizio"
},
{
  "op": "add",
  "path": "/fields/System.Description",
  "value": "Data d'inizio Del Progetto"
},
{
  "op": "add",
  "path": "/fields/Microsoft.VSTS.Scheduling.StartDate",
  "value": "Data Inizio x "
},
{
  "op": "add",
  "path": "/fields/Microsoft.VSTS.Scheduling.TargetDate",
  "value": "Data Inizio x "
}

```

Figura 3.22. Creazione *Epic* per “Data Inizio Progetto”

Nel corpo della richiesta, il contenuto JSON definisce un array di operazioni che aggiungono campi specifici al nuovo *Epic*. Ogni oggetto JSON rappresenta un’operazione *add* che inserisce valori nei campi del *work item*.

I campi fondamentali configurati includono il titolo dell’*Epic* (percorso */fields/System.Title*) con la stringa "Data Inizio", e la descrizione (percorso */fields/System.Description*). Oltre a questi, vengono definite la data di inizio (percorso */fields/Microsoft.VSTS.Scheduling.StartDate*) e di fine (percorso */fields/Microsoft.VSTS.Scheduling.TargetDate*), popolate con il medesimo valore dinamico (trattandosi di una milestone di progetto) estratto dalla lista SharePoint, identificato come “Data Inizio”.

3.3.5 Gestione Progetti

Le entità con stato di avanzamento “Progetto da Avviare” o successivo (“Progetto Avviato”, “Progetto in Analisi” ...) vengono gestite tramite un’apposita pagina dell’applicazione.

Il flusso inizia con un utente che accede alla lista dei progetti, visualizzando le iniziative in corso e lo stato di ciascuna. Da questa interfaccia, il responsabile può decidere se apportare modifiche ai dettagli del progetto o avanzare una richiesta di modifica dello stato. Nel caso in cui venga inviata una richiesta di approvazione per cambiare lo stato del progetto, il sistema passa alla fase successiva, in cui l’entità progetto viene collocata in uno stato temporaneo, identificato come "Modifica Stato in Corso".

A questo punto, l’*owner* interno, ossia il responsabile designato per la decisione finale, interviene per valutare la richiesta. Egli ha la possibilità di accettare o rifiutare la modifica dello stato del progetto. Se la richiesta viene approvata, il sistema aggiorna lo stato del progetto e pubblica automaticamente un messaggio sul canale Teams dedicato, informando tutti i membri del team della variazione avvenuta.

Nel caso in cui la richiesta venga invece rifiutata, lo stato del progetto rimane invariato, e il sistema invia comunque una notifica su Teams, specificando le motivazioni del rifiuto.

In entrambi i casi, in seguito alla decisione, il progetto esce dallo stato temporaneo "Modifica Stato in Corso" e torna al normale flusso operativo, consentendo nuovamente di apportare modifiche all’entità e di modificarne lo stato.

3.3 – Gestione Entità Progetto

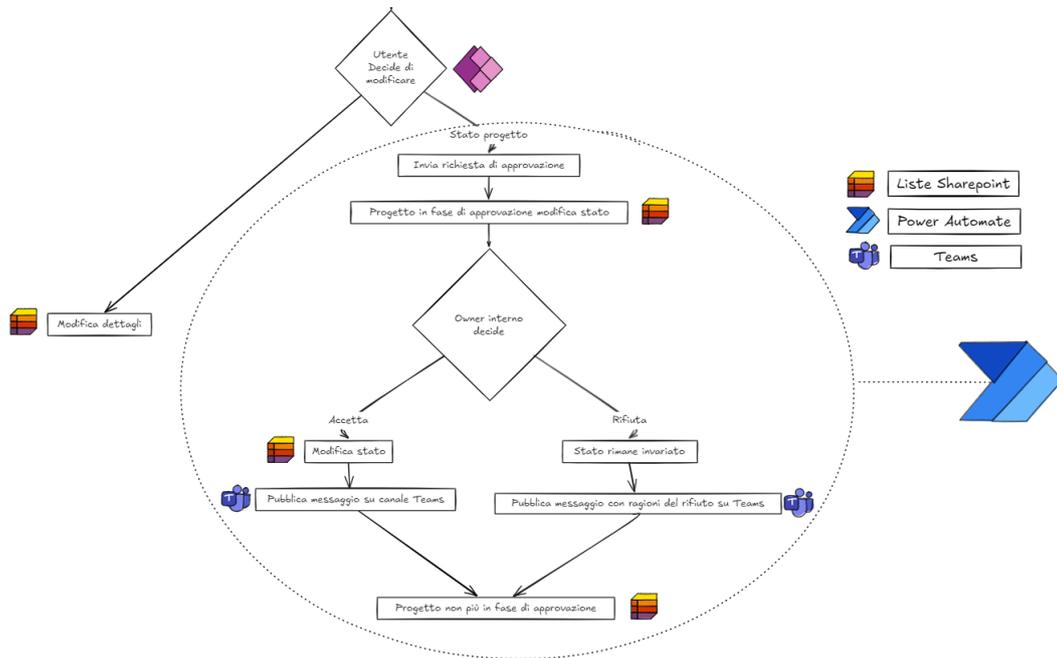


Figura 3.23. Diagramma Processo Gestione Progetti

Pagina "Gestione Progetti"

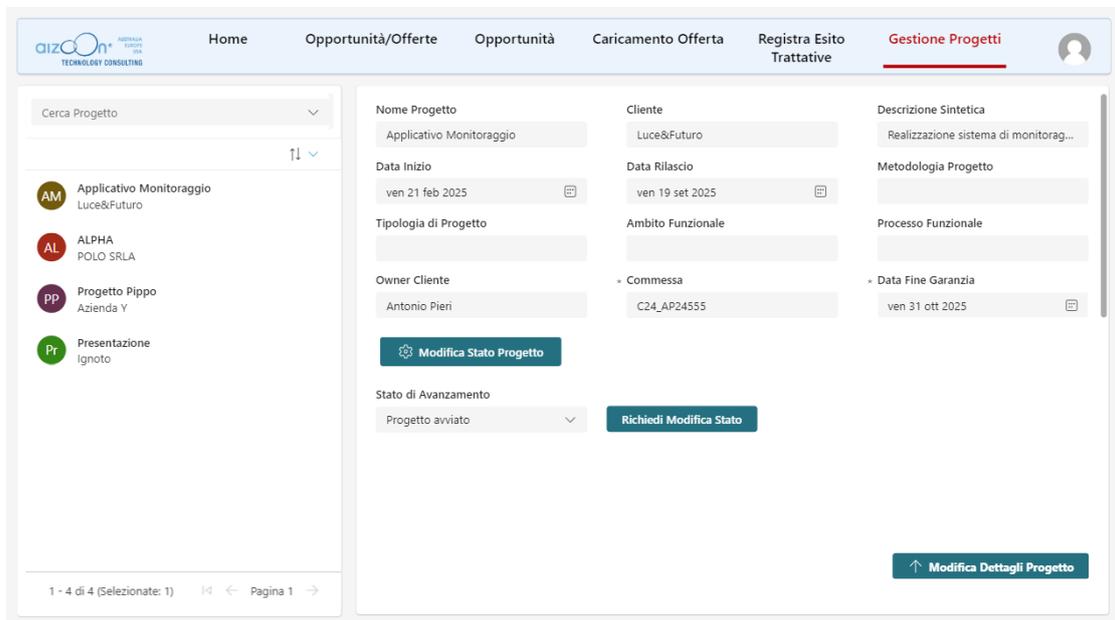


Figura 3.24. Pagina "Gestione Progetti"

La struttura della pagina riprende quella delle precedenti, con la possibilità di selezionare un progetto da una lista, effettuando una ricerca tra i progetti in stato “Progetto da Avviare” o successivo, escludendo i progetti per cui è già in corso una modifica dello stato di avanzamento.

L’utente può quindi decidere di modificare semplicemente alcuni campi dell’entità mostrati dal form, selezionando il pulsante “Modifica Dettagli Progetto”.

Scegliendo, invece, di richiedere la modifica dello stato di avanzamento, all’utente è richiesta la compilazione obbligatoria di 3 campi, necessari per l’avvio effettivo del progetto:

- *Owner Cliente*: referente o responsabile lato cliente del progetto.
- *Codice Commessa*: codice della commessa a cui il progetto è associato;
- *Owner Interno*: per selezionare il responsabile interno del progetto, tra i dipendenti dell’organizzazione.
- *Data Fine Garanzia*: data che indica il termine del periodo di garanzia associato al progetto.

Prima richiedere la modifica dello stato con il pulsante “Richiedi Modifica Stato”, è necessario selezionare, tramite *ComboBox*, lo stato di avanzamento a cui si vuole portare il progetto.

Gli stati selezionabili sono solamente quelli a partire da “Progetto da Avviare”, essendo questo processo utilizzato unicamente per la gestione delle entità diventate progetti effettivi.

Flusso Power Automate Per la Gestione Dei Progetti

Il flusso viene innescato tramite la pagina dell'applicazione, selezionando il pulsante "Richiedi Modifica Stato", passando come parametro della funzione *Run()*, l'id del progetto per cui è richiesta la modifica.

Dopo aver ottenuto i dettagli dell'elemento selezionato sulla lista Sharepoint con l'azione "Ottieni Elemento" e aver impostato lo stato temporaneo "Modifica Stato In Corso" con l'azione "Aggiorna Elemento", viene inviata una richiesta di approvazione della modifica al responsabile di progetto.

Tale richiesta specifica il passaggio di stato desiderato e incarica l'owner di verificare personalmente la presenza di tutte le informazioni necessarie, specificando nella sezione "Commenti" le motivazioni che hanno portato ad un eventuale rifiuto.



Figura 3.25. Richiesta Approvazione per Modifica Stato Progetto

In caso di esito positivo, lo stato viene aggiornato, mentre in caso di esito negativo, esso rimane invariato.

In entrambi i casi, come specificato in precedenza, il progetto esce dallo stato temporaneo “Modifica Stato in Corso” e viene inviata la relativa notifica sul canale Teams di progetto. La notifica, in caso di rifiuto, comprende anche le motivazioni riportate dal responsabile nella sezione “Commenti” nella richiesta di approvazione su *Approvals*.

3.4 Gestione Entità Risorsa

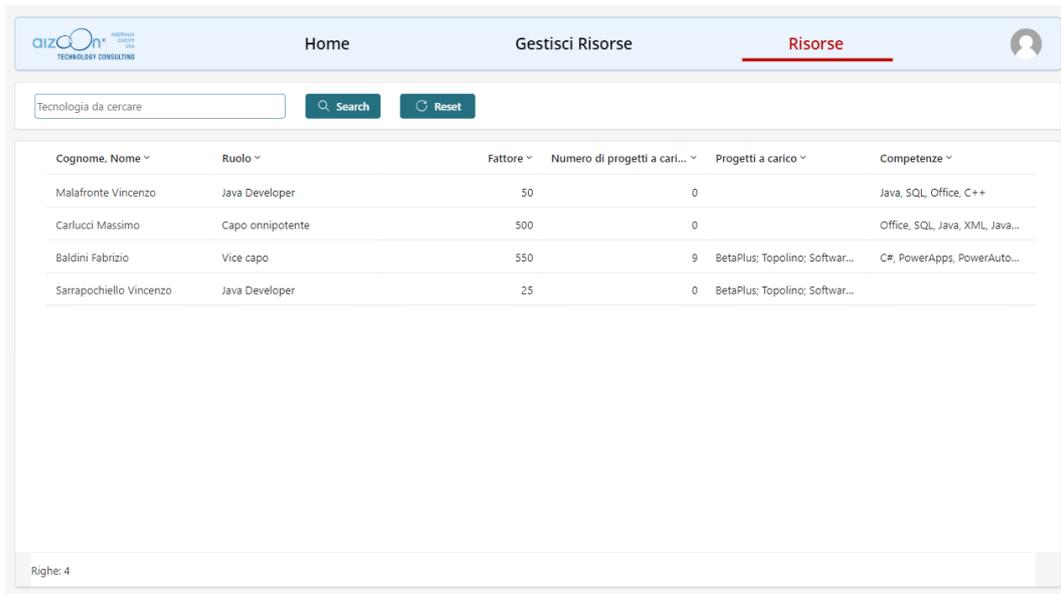
3.4.1 Censimento Risorse

Per censire le risorse aziendali è stata sviluppata un’ulteriore applicazione, che permette di definire le risorse e modificarne i dettagli, come il fattore (costo giornaliero) e le competenze tecniche.

È possibile allo stesso tempo tenere traccia dello storico dei fattori e della disponibilità della risorsa, determinata con il numero di progetti a carico, calcolato a partire dal numero di team di progetto in cui la risorsa è inserita su Azure DevOps.

L’applicazione è strutturata in due pagine distinte: la pagina “Risorse”, che permette la semplice visualizzazione delle risorse attualmente censite in azienda e la pagina “Gestisci Risorse”, che consente di selezionare una risorsa esistente e modificarne i dettagli o definire una nuova risorsa da zero, tra i dipendenti dell’organizzazione.

3.4 – Gestione Entità Risorsa

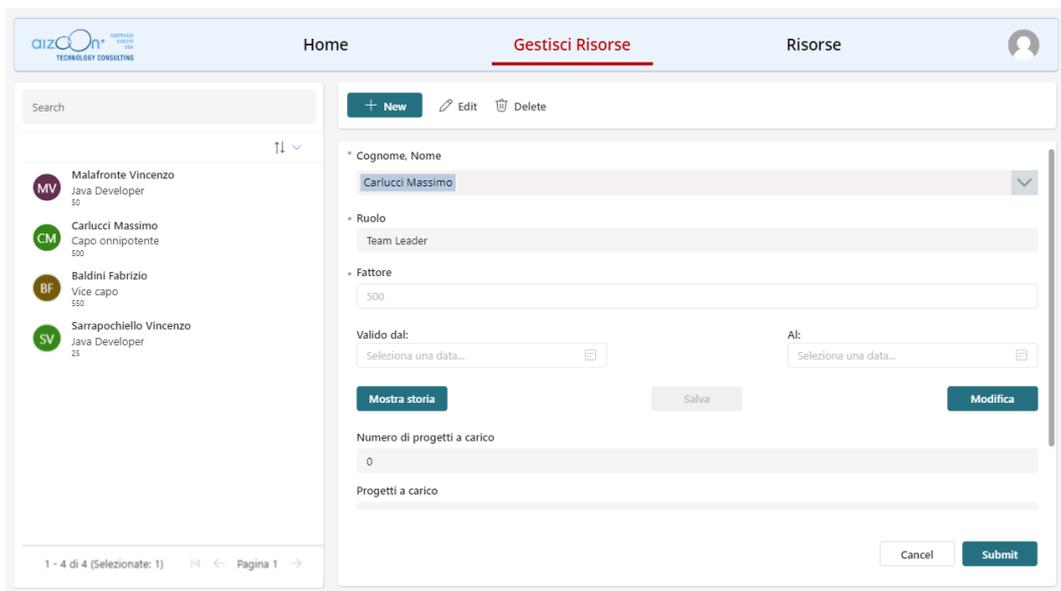


The screenshot shows the 'Risorse' page with a search bar and a table of resources. The table has columns for 'Cognome, Nome', 'Ruolo', 'Fattore', 'Numero di progetti a cari...', 'Progetti a carico', and 'Competenze'. The data is as follows:

Cognome, Nome	Ruolo	Fattore	Numero di progetti a cari...	Progetti a carico	Competenze
Malafrente Vincenzo	Java Developer	50	0		Java, SQL, Office, C++
Carlucci Massimo	Capo onnipotente	500	0		Office, SQL, Java, XML, Java...
Baldini Fabrizio	Vice capo	550	9	BetaPlus; Topolino; Softwar...	C#, PowerApps, PowerAuto...
Sarrapochiello Vincenzo	Java Developer	25	0	BetaPlus; Topolino; Softwar...	

Figura 3.26. Pagina “Risorse”

La pagina permette di ricercare le risorse che posseggono determinate competenze tecniche.



The screenshot shows the 'Gestisci Risorse' page with a search bar and a list of resources on the left. The main area shows a form for editing a resource. The form fields are:

- Cognome, Nome: Carlucci Massimo
- Ruolo: Team Leader
- Fattore: 500
- Valido dal: Seleziona una data...
- Al: Seleziona una data...
- Numero di progetti a carico: 0
- Progetti a carico: (empty)

Buttons: + New, Edit, Delete, Mostra storia, Salva, Modifica, Cancel, Submit.

Figura 3.27. Pagina “Gestisci Risorse”

Tramite la pagina “Gestisci Risorse” è possibile selezionare una risorsa esistente e, utilizzando la modalità “Edit”, modificare i campi relativi, come il ruolo e le competenze specifiche.

La funzionalità principale è rappresentata dalla definizione del fattore per la risorsa. Tale costo giornaliero, come mostrato in figura, necessita la definizione di un periodo di validità, determinato dai due *DataPicker* presenti, che devono essere obbligatoriamente compilati.

Dopo aver definito il nuovo fattore e il relativo periodo di validità, la pagina consente di salvare quanto stabilito e contestualmente visualizzare, tramite popup, lo storico dei fattori per risorsa, selezionando il pulsante “Mostra Storia”.

Flusso Power Automate Per il Calcolo Dei Progetti a Carico

Il flusso per il calcolo dei progetti a carico di una risorsa utilizza il trigger “Ricorrenza”, che lancia automaticamente l’esecuzione del flusso nei giorni e nelle ore specificate dall’utente.

Sfruttando le richieste HTTP descritte nella precedente sezione, il flusso verifica, per ogni risorsa presente nella Lista Sharepoint definita, la presenza o meno di quest’ultima tra i membri del *Default Team* di tutti i progetti dell’organizzazione.

Nel caso in cui tale presenza in un progetto venga verificata, viene incrementata la variabile di tipo *Integer* “ProgettiACarico” e viene concatenato alla variabile di tipo stringa “NomeProgetti” il nome del progetto in questione.

I campi della lista “Numero di Progetti a carico” e “Progetti a carico” vengono popolati con i valori delle due variabili utilizzando l’azione “Aggiorna Elemento” all’interno del ciclo che esegue il controllo su tutte le risorse censite, così da ottenere per ogni risorsa il numero totale di progetti e la stringa completa con i relativi identificativi.

① Numero di progetti... ▾	☰ Progetti a carico ▾
<hr/>	
0	
<hr/>	
2 Software Gestione Ordini; Applicativo Monitoraggio	
<hr/>	
9 BetaPlus; Topolino; Software Gestione Ordini; Progetto Pippo; Ubicazione Target; Presentazione; Applicativo Monitoraggio; Progetto Nexus; ALPHA	

Figura 3.28. Esempio Output Campi “Numero Progetti a carico” e “Progetti a carico”

3.5 Capacity Plan Preventivo

3.5.1 Estrazione Risorse Fittizie dal Calculation Model

Come specificato nella descrizione del processo di caricamento offerta, nella sezione “Allegati” è necessario caricare il *Calculation Model*, ovvero l’Excel contenente il dettaglio delle risorse fittizie e i relativi costi, determinati per l’opportunità in questione.

All’interno del *Calculation Model* risulta di particolare importanza il foglio di lavoro denominato “IA Progetti”, in cui sono definite le risorse fittizie ed il relativo costo e prezzo giornaliero, oltre alle macrotask stabilite per il progetto (da inserire successivamente su Azure DevOps), insieme alla loro durata e costo totale, calcolato alla luce dell’impegno di risorse richiesto da ciascuna.

Per la definizione del *Capacity Plan Preventivo*, ovvero di uno strumento di pianificazione strategica che permetta di stimare e organizzare le risorse necessarie, in fase di offerta, per affrontare le attività future di progetto, è necessario estrarre le

risorse riportate nel *worksheet*, con i costi giornalieri associati, e popolare successivamente una lista Sharepoint già predisposta.

Nella figura riportata di seguito vi è un esempio del *worksheet* sopra descritto, con le macroattività a sinistra e le risorse fittizie (Jr Servizi, Dev Servizi, An Servizi. . .), costi e prezzi in alto.

Macroattività								Giorni
	Costo	Jr Servizi	Dev Servizi	An Servizi	Sr Servizi	Expert Servizi	Tech. 1	
	Prezzo	€ 242,25	€ 273,63	€ 307,41	€ 341,89	€ 409,84	€ 300,00	
Assessment as-is				5				5
Analisi				8				8
Design		24			6			30
Creazione e assegnazione macro-gruppi per Facilitatori - BE			6					6
Creazione e assegnazione macro-gruppi per Facilitatori - FE		8			2			10
Gestione modifica e cancellazione macro-gruppi		5						5
Viste in sola lettura su analisi e percorsi		5						5
Export dati					2			2
Creazione gruppi Partecipanti - Dashboard - BE			6					6
Creazione gruppi Partecipanti - Dashboard - FE		12			3			15
Flow di tuning e gestione algoritmo per selezione clusterizzazione definitiva		3			2			5
Modifica manuale su gruppi creati da algoritmo. Arricchimento dati		8			2			10
Schermata di overview e collegamento con percorsi singoli e di gruppo		5						5
Assegnazione profili a gruppi e gestione target di transizione		6			4			10
Visualizzazione percorso - BE			2					2
Visualizzazione percorso - FE		2						2
QA		7						7
Doc				5				5

Figura 3.29. Foglio “IA Progetti” Calculation Model

L’obiettivo è perciò quello di estrarre la parte dello schema relativa alle risorse, costruendo una struttura dati importabile su SharePoint.

Per automatizzare l’estrazione e il popolamento della lista delle risorse fittizie, è stato costruito un flusso con Power Automate.

Flusso per l’Estrazione delle Risorse Dal Calculation Model

Il flusso viene attivato attraverso la selezione del pulsante “Aggiorna Dettagli Offerta” nella pagina per il caricamento dell’offerta nell’applicazione Power Apps.

A differenza dei trigger che sfruttano il connettore a Power Apps descritti precedentemente, in questo caso vengono passati come parametri di input l’identificativo del progetto selezionato nella pagina per il caricamento, ma anche il file, ovvero il *Calculation Model* caricato nella sezione “Allegati”.

Tale passaggio consente al flusso di operare dinamicamente sul *Calculation Model*

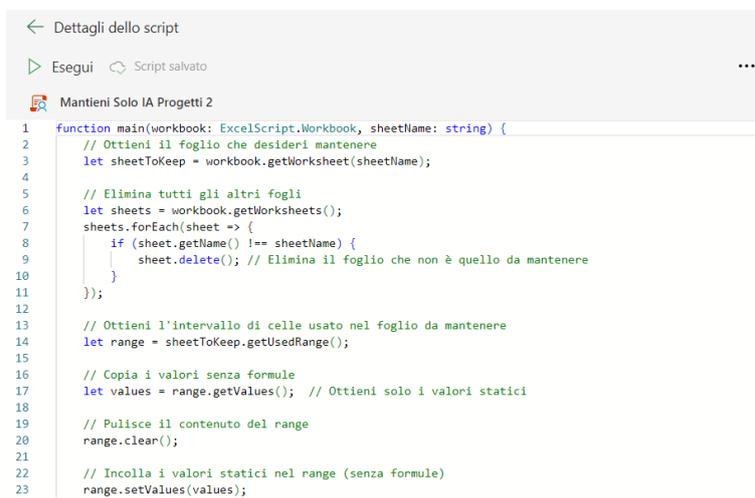
specifico di ogni progetto, senza dover impostare manualmente il file di partenza, ogniqualvolta si desidera caricare una nuova offerta.

Le azioni iniziali del flusso sono volte a creare una copia temporanea del file di partenza, che verrà eliminata al termine del flusso, in modo da poter modificare liberamente il file per l'estrazione delle risorse.

Per automatizzare le attività all'interno del file Excel sono stati utilizzati gli *Office Scripts* (accessibili tramite la sezione “Automatizza”), ovvero una funzionalità disponibile su *Excel Online*, progettata per automatizzare i fogli di lavoro direttamente nel cloud. Gli script vengono scritti in *TypeScript* (un linguaggio basato su JavaScript) e risultano particolarmente utili per l'integrare l'automazione con altri strumenti Microsoft come appunto Power Automate.

Tramite l'azione nativa “Esegui Script” in Power Automate, è infatti possibile lanciare l'esecuzione degli *script*, indicando il percorso, il file di riferimento ed eventuali parametri richiesti.

L'esempio di *script* mostrato di seguito permette di eliminare dal file Excel temporaneo tutti i fogli di lavoro ad eccezione di “IA Progetti”.



```
1 function main(workbook: ExcelScript.Workbook, sheetName: string) {
2   // Ottieni il foglio che desideri mantenere
3   let sheetToKeep = workbook.getWorksheet(sheetName);
4
5   // Elimina tutti gli altri fogli
6   let sheets = workbook.getWorksheets();
7   sheets.forEach(sheet => {
8     if (sheet.getName() !== sheetName) {
9       sheet.delete(); // Elimina il foglio che non è quello da mantenere
10    }
11  });
12
13  // Ottieni l'intervallo di celle usato nel foglio da mantenere
14  let range = sheetToKeep.getUsedRange();
15
16  // Copia i valori senza formule
17  let values = range.getValues(); // Ottieni solo i valori statici
18
19  // Pulisce il contenuto del range
20  range.clear();
21
22  // Incolla i valori statici nel range (senza formule)
23  range.setValues(values);
24 }
```

Figura 3.30. *Script* Excel

A questo punto, per poter popolare correttamente la lista Sharepoint, con i campi

“identificativo risorsa”, “costo” e “prezzo”, è necessario trasporre le celle contenenti queste informazioni, utilizzando uno *script* specifico e successivamente sfruttare l’azione “Crea Tabella” per creare una tabella all’interno del foglio, specificando l’intervallo di celle di destinazione.

Per inserire le singole risorse all’interno della lista, occorre utilizzare l’azione “Elenca le righe presenti in una tabella” in combinazione con un ciclo che crea un elemento Sharepoint per ogni riga estratta dal foglio Excel.

Infine, è stata utilizzata l’azione “Invia Richiesta HTTP a Sharepoint” per eliminare il file temporaneo.

Indirizzo del sito *

Formazione Progetto - https://aizoon365.sharepoint.com/sites/FormazioneProgetto

Metodo *

DELETE

URI *

_api/web/GetFileByServerRelativeUrl('/sites/FormazioneProgetto/Shared%20Documents/Rlso rse%20progetto.xlsx')

Parametri avanzati

Visualizzazione di 1 di 2

Mostra tutto

Cancela tutto

Intestazioni

Prefer

bypass-shared-lock

Figura 3.31. Richiesta HTTP a Sharepoint

Un elemento cruciale della configurazione è l’intestazione *Prefer: bypass-shared-lock*, che ha lo scopo di ignorare eventuali blocchi attivi sul file.

Questo è particolarmente utile in situazioni in cui il file potrebbe essere bloccato perché creato nel medesimo flusso, consentendo comunque l’eliminazione senza ostacoli.

3.5.2 Gantt e Gestione Risorse: Integrazione tra Power BI e Power Apps

Per la creazione del *Capacity Plan Preventivo* sono stati utilizzati gli strumenti forniti da Power BI per la visualizzazione dell'elenco delle risorse, per applicare filtri ai dati e per la costruzione del *Resource Gantt Chart*, ovvero un diagramma in cui il focus è rappresentato dalle risorse e dal loro impiego nel corso dell'intera durata del progetto.

Il punto di partenza è stato la creazione del report su Power BI Desktop (applicazione desktop di Power BI) e la connessione con le fonti dati necessarie, ovvero le liste Sharepoint relative ai progetti e alle risorse fittizie, estratte dal *Calculation Model*.

Durante l'importazione dei dati, utilizzando il *Power Query Editor* di Power BI è stato possibile applicare delle trasformazioni ai dati di partenza, tra cui la scelta delle colonne rilevanti ai fini della creazione del report e il merge di *query*, utile per creare una relazione tra le due tabelle, basata sull'identificativo dell'opportunità, che permette di utilizzare un'unica tabella "Risorse", con i campi utili, provenienti sia dalla lista di opportunità ("NomeProgetto", "Budget Stimato", "Marginalità"...) che dalla lista delle risorse fittizie.

Il report relativo al *Capacity Plan Preventivo*, pubblicato su Power BI Service, per permettere ad ogni membro dell'organizzazione di accedervi, è strutturato come mostrato in figura.

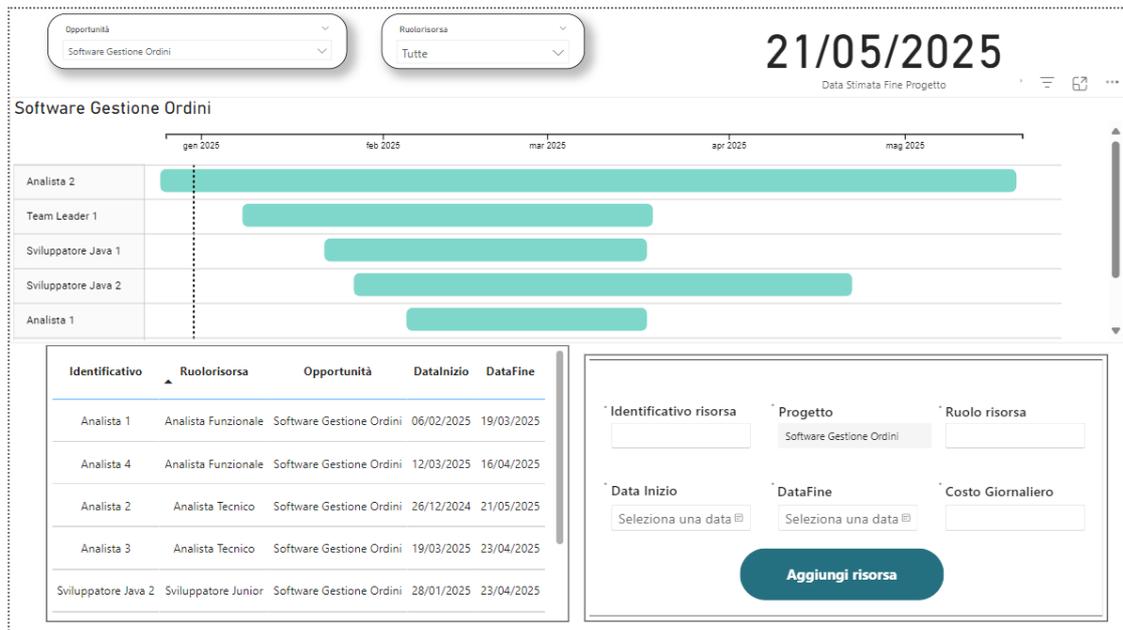


Figura 3.32. Capacity Plan Preventivo su Power BI

Al centro della dashboard si trova un diagramma di Gantt (importato dal Market-Place di Microsoft, non essendo una visualizzazione di base) che offre una visualizzazione chiara della distribuzione temporale delle attività delle risorse coinvolte. Le barre del diagramma identificano ciascuna risorsa, indicando il periodo in cui sono operative, con una rappresentazione visiva semplice e intuitiva delle sovrapposizioni e delle sequenze temporali.

La sezione superiore del report include la data stimata di fine progetto (ricavata dal Gantt) e due filtri interattivi fondamentali per personalizzare l'analisi. Uno consente di selezionare un'opportunità specifica, mentre l'altro permette di concentrarsi su un determinato ruolo delle risorse, come analisti, sviluppatori o altri. Questi filtri si integrano perfettamente con tutte le visualizzazioni della dashboard, aggiornando il Gantt, la tabella e gli altri elementi in base alle selezioni effettuate. Questo approccio offre agli utenti un controllo completo sull'analisi, adattandosi a esigenze specifiche e migliorando la comprensione dei dati.

Nella sezione inferiore, una tabella riassuntiva fornisce un livello maggiore di dettaglio, elencando le risorse con i relativi ruoli, i periodi di attività e opportunità di riferimento. Questa tabella, sincronizzata con i filtri, permette di accedere rapidamente ai dati chiave, offrendo un complemento analitico al Gantt e supportando un'esplorazione dettagliata.

Infine, nella parte inferiore destra del report, è presente un form dinamico creato con Power Apps e integrato all'interno di Power BI, selezionando l'apposita visualizzazione, che consente di scegliere un'app dal proprio ambiente di sviluppo o di crearne una nuova, passando come parametro un campo (per esempio l'id della risorsa o il nome dell'opportunità) della tabella "Risorse" creata su Power BI, per poterlo utilizzare nell'applicazione.

Sfruttando, infatti, la funzione *PowerBIIntegration*, questo form consente agli utenti di aggiungere nuove risorse direttamente dalla dashboard, specificando informazioni come ruolo, periodo d'impiego e costo giornaliero (il progetto viene precompilato alla selezione del filtro nel report), ma soprattutto di modificare le risorse esistenti, selezionandole nella tabella riassuntiva del report.

Alla selezione della risorsa, infatti, il form si popola automaticamente con i dettagli della risorsa, permettendo la modifica unicamente delle date d'impiego e del costo giornaliero.

The image shows a Power BI report interface. On the left is a summary table with columns: Identificativo, Ruolo/risorsa, Opportunità, DataInizio, and DataFine. The first row is highlighted. On the right is a form titled 'Modifica Parametri Risorsa' with fields for Identificativo risorsa, Progetto, Ruolo risorsa, Data Inizio, Data Fine, and Costo Giornaliero. A 'Modifica Parametri Risorsa' button is at the bottom of the form.

Identificativo	Ruolo/risorsa	Opportunità	DataInizio	DataFine
Analista 1	Analista Funzionale	Software Gestione Ordini	06/02/2025	19/03/2025
Analista 4	Analista Funzionale	Software Gestione Ordini	12/03/2025	16/04/2025
Analista 2	Analista Tecnico	Software Gestione Ordini	26/12/2024	21/05/2025
Analista 3	Analista Tecnico	Software Gestione Ordini	19/03/2025	23/04/2025
Sviluppatore Java 2	Sviluppatore Junior	Software Gestione Ordini	28/01/2025	23/04/2025

Identificativo risorsa	Progetto	Ruolo risorsa
Analista 1	Software Gestione Ordini	Analista Funzionale
Data Inizio	Data Fine	Costo Giornaliero
ven 7 feb 2025	gio 20 mar 2025	70

Modifica Parametri Risorsa

Figura 3.33. Modifica Parametri Risorsa su Power BI

Il report, sfruttando le funzionalità di Power Apps, non rappresenta, perciò, unicamente uno strumento di visualizzazione dei dati, ma diventa anche uno strumento operativo, permettendo di aggiornare e gestire le informazioni in tempo reale, operazioni che altrimenti non sarebbero possibili.

L'aggiornamento delle fonti dati (liste Sharepoint) utilizzate da Power BI non avviene automaticamente, ma richiede normalmente l'intervento dell'utente, tramite la selezione dalla funzionalità "Aggiorna".

Per ovviare a tale problematica è stato costruito un flusso su Power Automate, che attraverso l'azione nativa "Aggiorna un set di dati", che sfrutta il connettore a Power BI (specificando il report in Power Bi Service), aggiorna il set di dati alla selezione del pulsante "Aggiungi Risorsa" o "Modifica Parametri Risorsa" nel form sviluppato su Power Apps.



Figura 3.34. Flusso per l'Aggiornamento del Set di Dati in Power BI

Capitolo 4

Analisi dei Risultati Ottenuti

4.1 Introduzione

Nel capitolo precedente è stato illustrato il processo di progettazione e implementazione di una soluzione integrata per la gestione dei progetti e delle risorse, sviluppata mediante Microsoft Power Platform.

L'analisi che segue esplora i benefici tangibili e intangibili che l'adozione di questa piattaforma di sviluppo *low code* ha portato al progetto, mettendo in luce sia i punti di forza sia le limitazioni emerse durante il suo utilizzo.

Il capitolo si conclude con un confronto tra l'approccio offerto da una piattaforma *low code* come Microsoft Power Platform e le soluzioni cosiddette *custom*, nell'ambito del progetto descritto e in generale per realizzare sistemi CRM (*Customer Relationship Management*) personalizzati.

Il progetto in questione, infatti, presenta diverse caratteristiche che lo rendono assimilabile a un sistema CRM, pur non essendo un CRM tradizionale focalizzato sulla gestione delle relazioni con i clienti.

In primo luogo, il sistema gestisce e coordina risorse aziendali e il loro coinvolgimento nei progetti, un aspetto che in un CRM tradizionale è legato principalmente alla gestione dei contatti e dei clienti.

Inoltre, l'integrazione con Power BI consente un monitoraggio costante dei KPI aziendali, un aspetto comune nei CRM per analizzare e ottimizzare le performance, che in questo caso riguarda il progresso dei progetti e l'efficienza delle risorse.

L'automazione dei flussi di lavoro tramite Power Automate, che permette di semplificare attività ripetitive e garantire il rispetto delle tempistiche, rappresenta un altro elemento assimilabile ai processi automatizzati tipici dei CRM, in cui le automazioni riguardano i flussi di approvazione e le interazioni con i clienti. A questo proposito, un ulteriore punto di contatto con un sistema CRM è costituito dalla registrazione e l'inserimento automatico dei dati (in questo caso di progetti e risorse), che viene gestito in modo efficiente attraverso i flussi automatizzati, evitando errori e ottimizzando i diversi processi.

Infine, l'integrazione garantita da Power Platform favorisce l'interoperabilità con altre piattaforme aziendali, come ad esempio Azure DevOps, consentendo una gestione centralizzata delle operazioni e dei dati aziendali. Centralizzando i dati relativi alle risorse e ai progetti in un'unica piattaforma, similmente a come un CRM raccoglie e organizza i clienti e le relative informazioni, è possibile un facile accesso e aggiornamento da parte degli utenti.

Il confronto tra l'approccio *low code* e *custom* mira a fornire un quadro completo delle possibilità e delle sfide associate a entrambe le metodologie, evidenziando vantaggi e limitazioni in termini di flessibilità, costi, tempi di sviluppo e manutenzione.

4.2 Vantaggi dell'approccio Low-Code

4.2.1 Velocità di Implementazione

Uno degli aspetti più significativi emersi durante l'implementazione della soluzione con Microsoft Power Platform è stata la rapidità con cui è stato possibile sviluppare l'intero sistema operativo.

La velocità del processo di sviluppo può essere attribuita a quattro fattori determinanti: la semplicità nel design delle interfacce, la rapidità nella creazione di report e visualizzazioni, l’efficienza nella configurazione dei processi, l’integrazione automatizzata tra gli applicativi.

Design delle Interfacce

Nell’ambito del progetto descritto, la costruzione delle interfacce per la gestione delle opportunità, delle offerte e delle risorse è stata realizzata sfruttando Microsoft Power Apps. Questo strumento ha permesso di evitare completamente il design manuale delle UI, grazie all’utilizzo di componenti predefiniti (come moduli, tabelle e pulsanti) e alla possibilità di configurare layout in modalità *drag-and-drop*.

Le diverse pagine dell’applicazione presentano, infatti, una struttura standard di base e un processo di sviluppo molto simile. Sfruttando l’integrazione con le liste SharePoint come fonte dati, il sistema consente di selezionare rapidamente i campi da visualizzare e, in un secondo momento, e personalizzare i menu a discesa e definire i controlli necessari, come i pulsanti per l’invio e la cancellazione dei dati, senza la necessità di scrivere codice *front-end*.

Creazione Di Report e Visualizzazioni

Power BI ha permesso di costruire report dettagliati e visualizzazioni, come il *Resource Gantt Chart*, in tempi significativamente ridotti, grazie alle sue funzionalità di trasformazione dei dati. Attraverso il *Power Query Editor*, è stato possibile applicare modifiche e aggregazioni ai dati raccolti, eliminare colonne non necessarie, filtrare informazioni rilevanti e creare relazioni tra entità provenienti da diverse liste SharePoint.

Tutto ciò creando uno strato di trasformazione intermedio, che permette di lasciare

inalterate le fonti dati originali, mantenendo la loro integrità.

Ad esempio, i dati relativi a progetti e risorse, inizialmente frammentati in più liste, sono stati combinati per creare una tabella unica e strutturata. Questo approccio ha reso possibile una rappresentazione visiva immediata, dove ogni barra del Gantt mostra con chiarezza la disponibilità e l'allocazione delle risorse nei diversi progetti. La velocità con cui è stato possibile applicare trasformazioni alle strutture dati originali e configurare le visualizzazioni ha avuto un impatto diretto sul progetto, consentendo di disporre in breve tempo di strumenti di monitoraggio e previsione, che permettessero di realizzare una *What-if Analysis*.

Oltre a ciò, la possibilità di aggiungere filtri interattivi e definire misure (calcoli dinamici eseguiti sui dati a livello di modello o report) e colonne personalizzate, ha reso possibile rispondere rapidamente alle esigenze emerse in corso di sviluppo, adattando le visualizzazioni senza interventi sul back-end o sulle liste originali.

Tale strato di trasformazione aggiuntivo, combinato con la facilità d'uso di Power BI, ha migliorato notevolmente l'accessibilità ai dati e garantito una gestione più agile e flessibile, in linea con le tempistiche stringenti del progetto.

Configurazione dei processi

Un altro aspetto cruciale in cui la velocità di implementazione ha avuto un ruolo determinante riguarda la gestione dei flussi di lavoro.

Utilizzando Power Automate, è stato possibile, in primo luogo, configurare in pochi passaggi un sistema di notifiche e approvazioni, che informasse (utilizzando notifiche Teams) le risorse coinvolte e richiedesse l'intervento (tramite Approvals) di quelle designate, al verificarsi di determinati eventi.

Un caso emblematico è il flusso che automatizza il processo di approvazione delle nuove proposte di progetto. Grazie all'uso di trigger predefiniti (ad esempio, la creazione di un nuovo elemento in SharePoint), il flusso avvia automaticamente

una sequenza di azioni:

1. L’invio di una notifica su Teams con i dettagli dell’opportunità.
2. La generazione di una richiesta di approvazione tramite Microsoft Approvals.
3. L’aggiornamento dello stato dell’opportunità in base all’esito.

Un elemento chiave nella configurazione dei processi tramite Power Automate è quindi rappresentato dai trigger, che consentono di avviare automaticamente flussi di lavoro, in risposta a specifici eventi o condizioni.

Nel contesto del progetto, i trigger hanno giocato un ruolo cruciale per automatizzare attività ripetitive e migliorare l’efficienza operativa. In particolare, i trigger utilizzati nella costruzione dei flussi, descritti nel precedente capitolo, possono essere ricondotti a tre tipologie di eventi scatenanti:

- *Creazione di un elemento*: trigger basato sull’inserimento di un nuovo elemento in una lista SharePoint (progetto o risorsa).
- *Aggiornamento di stato*: trigger legato alla modifica di uno specifico campo su SharePoint.
- *Azioni manuali*: trigger attivati direttamente dall’utente tramite pulsanti in Power Apps.

Questo approccio ha reso i flussi di lavoro non solo più rapidi, ma anche più affidabili, poiché l’attivazione basata principalmente su eventi riduce il rischio di errori derivanti da azioni manuali mancanti o mal eseguite. I trigger hanno inoltre facilitato l’integrazione tra diversi sistemi, sincronizzando automaticamente i dati su piattaforme come SharePoint e Azure DevOps, garantendo una gestione coerente delle informazioni.

Infine, la flessibilità offerta da questi strumenti ha permesso di adattare i flussi alle esigenze specifiche del progetto. Per esempio, utilizzando un trigger basato su una

“Ricorrenza”, è stato possibile pianificare attività ricorrenti, come l’aggiornamento giornaliero dei progetti a carico delle risorse, senza richiedere interventi aggiuntivi da parte del team.

Integrazione nativa tra applicativi

L’utilizzo della Power Platform ha permesso di integrare nativamente diversi strumenti Microsoft, come SharePoint, Teams ed Excel, sfruttando un ampio catalogo di connettori predefiniti all’interno di Power Automate, senza necessità di sviluppare API o *middleware* personalizzati.

Questa integrazione nativa ha permesso di velocizzare ulteriormente i tempi di implementazione, eliminando la necessità di configurazioni tecniche complesse per l’autenticazione *cross-platform* o per la mappatura dei dati tra sistemi. Inoltre, i diversi processi relativi al censimento e alla gestione di progetti e risorse, che richiede la registrazione di dettagli su più piattaforme, è stato semplificato attraverso un unico *hub* di gestione su Power Apps, riducendo il rischio di errori e duplicazioni.

4.2.2 Velocità di Apprendimento

Un contributo significativo alla velocità d’implementazione del progetto è rappresentato dalla rapidità del processo di apprendimento per l’utilizzo della piattaforma. Power Apps, Power Automate e Power BI sono progettati per essere accessibili anche ad utenti non esperti, con interfacce intuitive e documentazione dettagliata. Inoltre, un aspetto da considerare è l’integrazione di questi applicativi con Copilot, l’assistente AI di Microsoft.

Per un utente che si avvicina per la prima volta a Power Automate e Power Apps, Copilot rappresenta uno strumento fondamentale per semplificare e velocizzare l’intero processo di apprendimento e sviluppo di applicazioni e automazioni.

In Power Apps, Copilot si integra perfettamente con il processo di sviluppo, facilitando la scrittura e la comprensione delle formule *Power Fx*, fondamentali per modificare le proprietà degli elementi grafici.

Grazie alla nuova funzionalità che permette di generare formule a partire da linguaggio naturale o da commenti di codice, Copilot ha consentito di concentrarsi sugli obiettivi dell’applicazione senza preoccuparsi eccessivamente della sintassi. Inoltre, la funzionalità di spiegazione delle formule aiuta a comprendere il funzionamento del codice esistente, offrendo descrizioni in linguaggio naturale di porzioni selezionate di formula, aspetto fondamentale per prendere familiarità col nuovo linguaggio. Nell’ambito della costruzione dei flussi in Power Automate, Copilot ne semplifica la creazione e la modifica. Utilizzando il linguaggio naturale, è infatti possibile descrivere il processo che si desidera automatizzare e Copilot genererà automaticamente il trigger e le azioni necessarie.

Inoltre, per quanto riguarda i flussi esistenti, Copilot è fondamentale per modificarli e migliorarli, suggerendo ottimizzazioni o aggiungendo passaggi utili. Un ulteriore

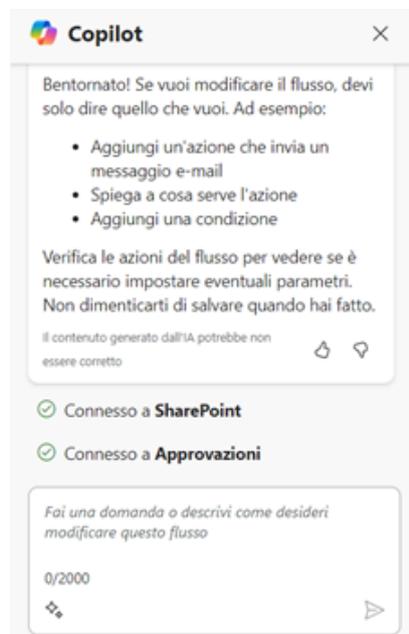


Figura 4.1. ChatBot Copilot integrato in Power Automate

aspetto significativo legato a Copilot è la sua capacità di offrire intelligenza contestuale. Esso è infatti in grado di analizzare le fonti dati integrate in un'applicazione Power Apps o i connettori utilizzati in un flusso, per fornire suggerimenti pertinenti e azioni basate sul contesto specifico.

Considerando il progetto esaminato, è stato possibile iniziare a utilizzare efficacemente gli strumenti forniti da Microsoft Power Platform in seguito a una formazione base iniziale di poche ore (basata sulla documentazione fornita da Microsoft), integrata dall'esperienza diretta con gli applicativi, le cui logiche risultano facilmente comprensibili e replicabili.

4.2.3 Prototipazione e Digitalizzazione dei Processi

L'adozione di Microsoft Power Platform si è rivelata particolarmente vantaggiosa per la prototipazione rapida e la digitalizzazione di processi aziendali che, in precedenza, venivano gestiti in modo manuale o frammentato. In particolare, il progetto ha beneficiato di questa capacità durante la digitalizzazione di attività strategiche come il caricamento delle offerte e il censimento delle risorse, che prima richiedevano un notevole impiego di tempo e risultavano esposte a errori umani.

L'approccio iterativo reso possibile dalla rapidità di creazione di applicazioni e flussi di lavoro, garantita piattaforma, ha permesso di coinvolgere gli utilizzatori finali (interni all'azienda) del *Tool*, già nelle prime fasi di sviluppo.

Durante progettazione dell'applicazione per la gestione dei progetti, il prototipo iniziale è stato testato direttamente dai responsabili, che hanno potuto fornire feedback immediati. Questo ciclo rapido di progettazione, test e revisione ha consentito di affinare il prodotto in tempi molto brevi, riducendo notevolmente il rischio di errori o insoddisfazione al momento del rilascio finale.

Un esempio significativo di digitalizzazione è rappresentato dal miglioramento dei processi di valutazione ed approvazione. Sfruttando l'integrazione con Power Automate, è stato possibile automatizzare flussi di approvazione che in precedenza

richiedevano numerosi passaggi manuali, come la revisione dei documenti e la comunicazione tra i diversi attori coinvolti.

La predisposizione di notifiche dettagliate ed automatizzate ai responsabili e ai team di progetto ha infatti assicurato che le operazioni richieste fossero ben definite e che venissero completate entro tempi prestabiliti (scadenza approvazioni). Questo approccio ha permesso di migliorare l'efficienza operativa e aumentare la trasparenza del processo, grazie alla possibilità di monitorare in tempo reale lo stato delle approvazioni e i relativi dettagli.

La digitalizzazione di processi, quali il censimento e la gestione di opportunità e risorse, ha ulteriormente dimostrato la versatilità della piattaforma.

L'implementazione di un'applicazione dedicata su Power Apps ha permesso di rivoluzionare il processo di censimento delle risorse aziendali, che in precedenza era gestito tramite l'utilizzo di fogli di calcolo Excel condivisi e quindi soggetti a frequenti errori o sovrapposizioni.

Gli utenti designati possono ora inserire e aggiornare i dati in un unico ambiente digitale centralizzato, accessibile da qualsiasi dispositivo, riducendo non solo i tempi necessari per la raccolta delle informazioni, ma ha anche eliminando ambiguità e duplicazioni, garantendo una maggiore accuratezza nella gestione dei dati.

Un importante passo avanti nella digitalizzazione dei processi aziendali è stato ottenuto nella gestione delle offerte e delle opportunità. In precedenza, questo processo coinvolgeva numerosi passaggi manuali, con comunicazioni via e-mail e interazioni tra più reparti, spesso causando ritardi e mancanza di visibilità sulle priorità.

Attraverso la creazione di un flusso automatizzato per la valutazione e l'approvazione delle offerte, ogni offerta viene notificata automaticamente ai decisori con i relativi dettagli, i quali possono approvarla o richiedere ulteriori modifiche con un'azione immediata su Microsoft Approvals.

In sintesi, la capacità di Microsoft Power Platform di agevolare la digitalizzazione e la rapida prototipazione dei processi ha costituito un elemento determinante per il

successo del progetto, permettendo di ottimizzare processi aziendali cruciali, accelerare i tempi di esecuzione, migliorare l'efficienza e garantendo maggiore precisione e trasparenza.

4.3 Limiti nell'utilizzo di Microsoft Power Platform

4.3.1 Customizzazioni Limitate ed Integrazione “Forzata”

Le piattaforme *low code* come Microsoft Power Platform, pur offrendo un approccio semplificato allo sviluppo, evidenziano alcune limitazioni significative quando si tratta di personalizzazioni avanzate. Questi limiti si sono manifestati durante il progetto descritto, soprattutto in relazione a Power Apps, influenzando sulla flessibilità e sulla capacità di adattare il sistema a esigenze specifiche.

Power Apps, pur essendo uno strumento potente per la creazione di interfacce utente, si scontra con vincoli che emergono chiaramente quando si cerca di sviluppare soluzioni personalizzate al di fuori delle funzionalità predefinite.

Un limite significativo riguarda le opzioni di personalizzazione dei componenti grafici. Sebbene sia possibile modificare proprietà base dei componenti (come colore, dimensione o visibilità), la mancanza di un controllo completo sull'aspetto grafico rende difficile soddisfare requisiti di branding o design specifici.

Nel progetto, ad esempio, si è rivelato complicato creare interfacce completamente in linea con il design aziendale senza scendere a compromessi, poiché i temi predefiniti di Power Apps non consentono una personalizzazione approfondita degli stili. Il design delle UI in Power Apps è quindi vincolato a un set di controlli e layout standardizzati. Le possibilità di personalizzazione, sebbene ampie, sono comunque limitate a ciò che è stato previsto dalla piattaforma (proprietà dei componenti), costringendo ad operare entro confini prestabiliti. In contesti che richiedono comportamenti altamente specifici o interfacce utente avanzate, questa limitazione può diventare un ostacolo significativo.

Le versioni moderne dei componenti di Power Apps hanno in generale portato miglioramenti significativi in termini di personalizzazione e gestione dei controlli. Ad esempio, per quanto riguarda la gestione dei DatePicker, i nuovi controlli permettono una personalizzazione più precisa, consentendo di impostare direttamente le proprietà *start date* ed *end date* per il selettore delle date, senza dover ricorrere a logiche complesse o espressioni aggiuntive. Questo miglioramento semplifica notevolmente la configurazione e l'esperienza dell'utente, rendendo la gestione delle date molto più fluida e diretta rispetto ai controlli precedenti, che non supportavano nativamente tali opzioni.

Tuttavia, in molti casi, i componenti moderni hanno limitato le opzioni di personalizzazione e reso lo sviluppo più complesso. Un esempio evidente riguarda i pulsanti e le proprietà *Hover*. Nei componenti "classici" era possibile personalizzare in modo avanzato gli effetti di *hover*, come il cambio di colore dello sfondo o del bordo (la proprietà *HoverFill* consente di cambiare il colore dello sfondo del bottone al passaggio del mouse, mentre *HoverBorderColor* permette di modificare il colore del bordo), permettendo un maggiore controllo sull'interfaccia. Con l'introduzione dei controlli moderni, questa capacità di personalizzazione è stata limitata, richiedendo l'uso di *Power Fx* per ottenere risultati simili, il che rende l'implementazione più complessa e meno immediata, soprattutto per chi non possiede competenze avanzate.

Un aspetto aggiuntivo in cui i componenti moderni hanno introdotto delle limitazioni, riguarda la validazione degli input. Nelle versioni precedenti dei componenti di input, risulta più semplice garantire che i dati inseriti in un campo di testo siano conformi al tipo previsto (ad esempio, numeri al posto delle lettere). Nei componenti moderni, sebbene ci siano opzioni per la gestione dei tipi di dati, l'approccio non è altrettanto rigido e richiede un intervento più manuale e una gestione esplicita tramite espressioni *Power Fx* per validare correttamente l'input. Questo ha reso più difficile mantenere un controllo completo sulla qualità dei dati inseriti,

aumentando la necessità di logiche personalizzate per garantire la validità delle informazioni.

Nell'ambito della progettazione del *Capacity Plan Preventivo*, uno dei problemi principali riscontrati è stato, invece, l'impossibilità di creare grafici e visualizzazioni interattive direttamente all'interno dell'applicazione Power Apps. Sebbene Power Apps consenta di visualizzare tabelle e dati collegati a liste SharePoint, non offre componenti nativi per generare grafici dinamici o rappresentazioni visive complesse. Questo ha limitato la possibilità di visualizzare immediatamente i dati relativi alla disponibilità delle risorse in base ai progetti pianificati, un requisito essenziale per il *Capacity Plan*. Per ovviare a questa mancanza, è stata necessaria un'integrazione con Power BI, che ha permesso di costruire report in cui inserire il dettaglio delle risorse e il relativo *Gantt Chart*.

Questa soluzione ha, però, introdotto un passaggio supplementare, obbligando gli utenti a consultare una piattaforma separata per ottenere una visione completa della disponibilità delle risorse, compromettendo l'esperienza unificata desiderata. Un secondo limite rilevante è rappresentato dall'impossibilità di creare tabelle temporanee e colonne "calcolate" all'interno di Power Apps, derivanti da altri campi delle liste collegate.

Questo problema si è manifestato chiaramente nella gestione delle risorse e nella pianificazione delle relative macroattività. Nella definizione delle macroattività di un progetto risulta necessario calcolare dinamicamente il costo e l'impiego totale delle risorse in base a campi come il costo giornaliero e le date previste d'inizio e fine impiego. I dati così calcolati devono poi essere aggregati per ottenere il costo totale della macroattività, la durata complessiva e infine i valori aggregati per l'intera opportunità.

Power Apps non supporta questa funzionalità in modo nativo, obbligando a sfruttare l'integrazione con Power BI, dove sono state create tabelle intermedie fittizie contenenti colonne calcolate per rappresentare i valori richiesti. Ad esempio, il costo

totale delle risorse è stato calcolato moltiplicando il costo giornaliero per il numero di giorni d'impiego, aggregando poi a livello di macroattività e opportunità.

Descrizione	Opportunità	Descrizione Risorsa	GG Risorsa	Costo Giornaliero Risorsa	Prezzo Giornaliero Risorsa	Costo Totale Risorsa	Prezzo Totale Risorsa	Costo attività	Prezzo attività	Giorni Attività
Attività 1	Applicativo Gestione Rifiuti	Analista Funzionale	10	200,00 €	270,00 €	2.000,00 €	2.700,00 €	3.200,00 €	4.260,00 €	22
Attività 1	Applicativo Gestione Rifiuti	Senior Developer	12	100,00 €	130,00 €	1.200,00 €	1.560,00 €	3.200,00 €	4.260,00 €	22
Attività 2	Applicativo Gestione Rifiuti	Analista Funzionale	10	200,00 €	270,00 €	2.000,00 €	2.700,00 €	2.000,00 €	2.700,00 €	10
Opportunità						5.200,00 €	6.960,00 €	8.400,00 €	11.220,00 €	54

Figura 4.2. Tabella Riepilogativa Risorse-Macroattività

In aggiunta a ciò, il *Capacity Plan Preventivo* descritto nel Capitolo 3 e sviluppato su Power BI ha richiesto l'integrazione con un form realizzato su Power Apps, per poter modificare effettivamente i dati ed aggiornarli in tempo reale.

Questa integrazione permette di estendere le capacità di Power BI, aggiungendo funzionalità di modifica e aggiornamento in tempo reale, ma introduce anche una certa complessità. È necessario, infatti, sviluppare e configurare una Power App separata, che rifletta le operazioni svolte dall'utente nel report (selezioni, filtri...) e che funga da interfaccia per la modifica dei dati, integrandola nel report Power BI tramite l'apposita visualizzazione.

A ciò si aggiunge la complicazione di dover aggiornare il report dopo aver modificato i dati tramite Power Apps. Di conseguenza, per garantire che i dati modificati vengano visualizzati correttamente nel report, è necessario implementare un processo di aggiornamento dei dati, che sfrutti Power Automate, come descritto precedentemente.

4.3.2 Limiti funzionali dei connettori Power Automate

Un'altra criticità significativa emersa durante il progetto riguarda l'integrazione complessa con Azure DevOps, descritta nel Capitolo 3. Il connettore nativo di Power Automate, sebbene utile per operazioni di base come la creazione di *work item* o il recupero di informazioni, si è rivelato limitato quando il progetto ha richiesto una gestione dinamica e complessa dei dati. Ad esempio, per creare automaticamente progetti specifici, basandosi sui dati inseriti in Power Apps, o per aggiungere o rimuovere un utente dal gruppo degli amministratori di progetto.

Questo ha reso necessario integrare richieste HTTP per invocare le API REST di Azure DevOps, aumentando la complessità del flusso e introducendo problemi di performance. Uno dei principali ostacoli è stato rappresentato dai ritardi nelle risposte delle API, che in alcuni casi hanno portato a timeout dei flussi o errori di sincronizzazione. Per mitigare questi problemi, è stato necessario introdurre ritardi programmati nei flussi, utilizzando l'azione "Delay" di Power Automate per attendere alcuni secondi tra un'operazione e l'altra.

Sebbene questa soluzione abbia ridotto la probabilità di errori, ha comportato un dilatamento delle tempistiche di esecuzione. Questo approccio, seppur efficace, ha introdotto un livello aggiuntivo di complessità nel progetto. Ogni operazione deve essere configurata manualmente, richiedendo di definire dettagli tecnici come le intestazioni di autenticazione, il *payload* dei dati da inviare e il metodo da utilizzare, elementi che hanno aumentato la difficoltà complessiva del processo.

L'implementazione delle richieste HTTP personalizzate richiede, infatti, competenze aggiuntive, per quanto riguarda la struttura delle API e il formato JSON, andando oltre le capacità normalmente necessarie in un ambiente *low code*. Questo ha avuto un impatto diretto sui tempi di sviluppo, rendendo l'intero processo significativamente più lento rispetto all'utilizzo delle azioni predefinite offerte dai connettori nativi.

Per effettuare tali richieste, è necessario, inoltre, generare un *Personal Access Token* (PAT) in Azure DevOps, che funge da metodo di autenticazione per accedere alle API REST. Il PAT, tuttavia, presenta diverse limitazioni.

In primo luogo, ha una durata limitata, poiché può essere configurato per un massimo di un anno, dopodiché deve essere rigenerato. Questo implica un onere amministrativo aggiuntivo, in quanto occorre monitorare e rinnovare i *token* prima della loro scadenza, per evitare l'interruzione dei flussi automatizzati.

A ciò si aggiunge che la validità del PAT è legata all'utente che lo ha generato, pertanto, tutte le richieste HTTP devono essere inviate nel contesto di tale utente. Questo vincolo riduce la flessibilità e complica la gestione in scenari in cui è richiesta l'esecuzione indipendente o collettiva dei flussi. Infine, per poter utilizzare il PAT, è necessario convertirlo in Base64 e inserirlo nelle intestazioni di autenticazione delle richieste HTTP.

Un ulteriore aspetto da considerare è la necessità di autenticazione dell'utente per avviare i flussi su Power Automate. Ogni flusso deve essere eseguito nel contesto di un utente autenticato e di conseguenza, in assenza di un'autenticazione attiva, i flussi non possono essere avviati o completati. Questo requisito ha limitato notevolmente l'autonomia del sistema.

Un esempio di questa complicazione in relazione al progetto riguarda il processo di creazione del progetto su Azure DevOps.

Quando il flusso viene avviato, l'utente che ha effettuato l'autenticazione viene automaticamente aggiunto come amministratore e parte del *Default Team* di progetto. In questo caso, l'utente autenticato potrebbe ricoprire un ruolo non previsto all'interno del progetto, richiedendo successivamente un passaggio supplementare per rimuoverlo da tali gruppi, tramite richieste HTTP separate.

Questo processo aggiuntivo risulta complesso e inefficiente, in quanto si rende necessario gestire manualmente l'eliminazione dell'utente dai gruppi di progetto. Se il flusso potesse essere avviato senza un'autenticazione attiva, utilizzando un utente

fittizio o un bot, questa operazione sarebbe evitata, semplificando notevolmente il processo e migliorando l'efficienza dell'automazione.

L'attuale vincolo di dover associare un flusso a un'autenticazione attiva rende necessario gestire manualmente tali circostanze con azioni supplementari e mantenere l'accesso agli applicativi costantemente disponibile.

4.3.3 Dipendenza dall'Ecosistema Microsoft e dal Cloud

Il progetto ha evidenziato una notevole dipendenza dall'ecosistema Microsoft, che rappresenta un aspetto cruciale da considerare quando si utilizzano strumenti come Microsoft Power Platform.

Questa piattaforma, pur offrendo una vasta gamma di funzionalità integrate e strumenti di sviluppo *low code*, crea una forte *lock-in* tecnologico, ossia un vincolo che lega fortemente i dati e i flussi di lavoro creati alla piattaforma stessa. Ciò significa che tutti i processi automatizzati, i flussi di lavoro (creati in Power Automate), le app (sviluppate con Power Apps) e i report (generati con Power BI) sono strettamente interconnessi con il sistema Microsoft. Una volta che questi flussi e applicazioni sono stati progettati e implementati, diventa estremamente difficile esportare o migrare i dati e i flussi verso altre piattaforme o sistemi esterni.

Le applicazioni della Microsoft Power Platform dipendono da specifiche connessioni e credenziali, che sono configurate in modo personalizzato per ogni ambiente Microsoft. Quando si tenta di esportare questi dati o flussi di lavoro verso altre piattaforme, è necessario riconfigurare tutte le connessioni e le autorizzazioni di accesso.

Inoltre, le licenze aziendali associate a Power Platform spesso includono configurazioni specifiche che non sono compatibili con soluzioni esterne dell'ecosistema Microsoft. Queste licenze includono l'accesso a funzionalità e servizi legati a prodotti Microsoft come Microsoft 365, Azure e Dynamics 365. L'azienda potrebbe trovarsi quindi di fronte alla necessità di ripensare o riconfigurare completamente

la propria infrastruttura tecnologica, nel caso di migrazione ad un altro ecosistema. Un esempio di come Power Platform sia strettamente legate all'ecosistema Microsoft è l'integrazione totale con Microsoft Teams, per quanto riguarda i processi di notifica ed approvazione precedentemente descritti, vincolati alle azioni dell'utente su Power Apps e quindi alle modifiche avvenute nelle liste Sharepoint. Risulterebbe estremamente complicato replicare o esportare tale integrazione al di fuori dell'ecosistema Microsoft.

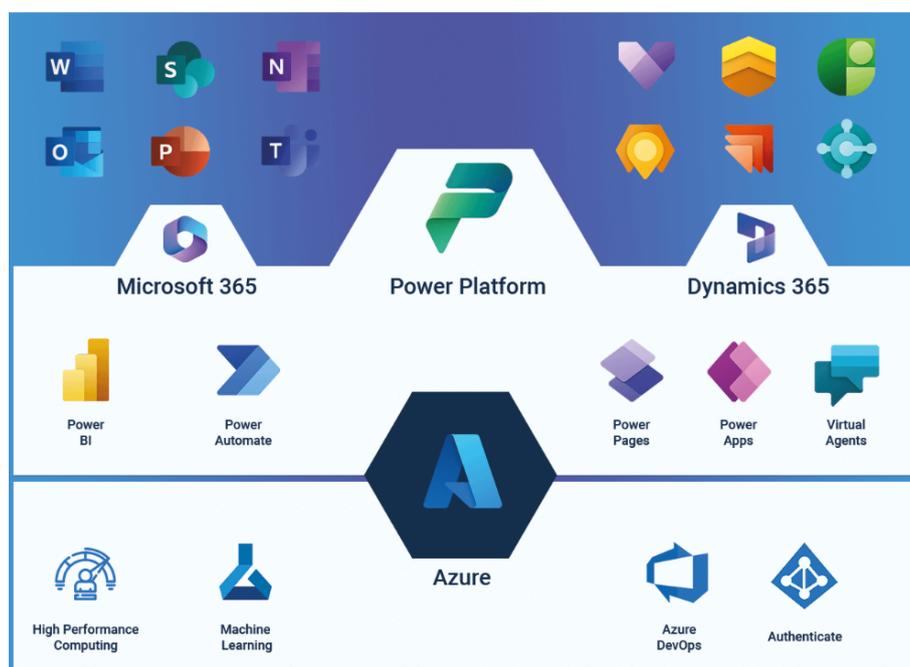


Figura 4.3. Ecosistema Microsoft Fonte: Yaveon

Un ulteriore vincolo significativo di Power Platform è costituito dalla dipendenza dal cloud, che implica che le applicazioni e i flussi di lavoro creati in Power Apps e Power Automate non possano funzionare offline.

Poiché queste piattaforme sono progettate per essere utilizzate nell'ambito di un'infrastruttura cloud, tutte le operazioni, la gestione dei dati e l'esecuzione dei flussi di lavoro avvengono in tempo reale attraverso il cloud Microsoft, che garantisce l'accesso ai dati e ai servizi necessari. L'ambiente di sviluppo integrato di Power

Platform (IDE) è accessibile solo tramite browser, il che implica che non è possibile sviluppare applicazioni o flussi senza una connessione a internet.

Un aspetto critico riguarda l'evoluzione del mercato tecnologico, in cui nuove soluzioni emergenti potrebbero non essere immediatamente compatibili con l'ecosistema Microsoft. In scenari del genere, l'azienda potrebbe trovarsi ad affrontare la scelta tra continuare a investire in un sistema chiuso e vincolato, o migrare a nuove tecnologie, affrontando però il rischio di dover riprogettare da zero una grande parte delle sue applicazioni e processi aziendali.

In questo contesto, l'investimento in Power Platform, sebbene inizialmente vantaggioso per la rapidità di sviluppo e implementazione, potrebbe rivelarsi problematico sul lungo periodo, soprattutto se si presentasse l'esigenza di evolvere verso soluzioni non vincolate a una piattaforma specifica.

4.3.4 Costi di Licenza

I costi di licenza possono rappresentare un limite di Microsoft Power Platform e in generale delle piattaforme *low code*, specialmente per le aziende che necessitano di soluzioni scalabili o che vogliono adottare la piattaforma su larga scala.

I costi possono diventare significativi a causa di diversi fattori, come la necessità di acquistare licenze premium per accedere a funzionalità avanzate, l'aumento dei costi con l'incremento del numero di utenti e la complessità delle operazioni aziendali.

Inoltre, la struttura di licenza di Power Platform può risultare poco trasparente e difficile da comprendere, soprattutto per le organizzazioni che non sono abituate a gestire licenze in modo dettagliato. Per esempio, i costi per funzionalità aggiuntive come il connettore premium, l'uso di Dataverse o l'integrazione di altre app Microsoft richiedono licenze separate, il che può aumentare esponenzialmente la spesa totale. La necessità di monitorare continuamente l'utilizzo delle licenze per evitare costi imprevisti potrebbe comportare un ulteriore impegno amministrativo.

Licenze Power Automate

Microsoft Power Automate offre diverse opzioni di licenza, pensate per soddisfare le necessità di diverse tipologie di business. I piani di licenza variano in base alle funzionalità richieste e al numero di utenti o flussi da gestire. L'opzione più comune consente, al costo di circa 12€-15€ per utente al mese, di utilizzare flussi cloud (DPA) e flussi desktop in modalità assistita (RPA), con una base di 50 MB di archiviazione dati.

Inoltre, sono inclusi 250 MB di capacità del database Dataverse e 2 GB di capacità di archiviazione dei file.

Per automatizzare i processi aziendali core con automazione RPA (*Robotic Process Automation*) non assistita, sono disponibili piani che prevedono una licenza per bot ad un costo di 120€-150€ al mese. Questi piani includono flussi cloud, flussi desktop in modalità non assistita e, ma con una capacità di archiviazione più limitata di 50 MB per il database e 200 MB per i file.

Esiste un ulteriore piano che offre una macchina virtuale gestita da Microsoft su infrastruttura Azure, rendendolo ideale per le aziende che necessitano di una gestione centralizzata e scalabile dei flussi di lavoro.

Per testare Power Automate è possibile esplorare la piattaforma gratuitamente per 30 giorni, con la possibilità di sperimentare i flussi cloud usando connettori standard[25](Microsoft, 2024).

Licenze Power Apps

Per quanto riguarda Power Apps, una delle soluzioni più interessanti è rappresentata dal piano per sviluppatori, disponibile gratuitamente. Questo piano consente di creare e testare un numero illimitato di applicazioni e flussi di automazione, offrendo tre ambienti dedicati per lo sviluppo e il supporto per connettori predefiniti, personalizzati e locali. Tale piano è stato utilizzato per lo sviluppo dell'applicazione Power Apps in questa fase di realizzazione del progetto. Non è possibile utilizzarlo per applicazioni o flussi in ambienti di produzione.

Per esigenze aziendali più complesse, è disponibile un piano che rappresenta una soluzione più completa.

Questo piano offre accesso illimitato alle applicazioni Power Apps e Power Pages per ogni utente. È incluso anche il supporto per connettori predefiniti, personalizzati e locali, oltre ai diritti Dataverse con una capacità di 250 MB per il database e 2 GB per i file, garantendo la possibilità di gestire dati in modo efficace e sicuro. Per le organizzazioni di grandi dimensioni, Microsoft offre una variante del piano precedente, ad un costo ridotto, mantenendo tutte le funzionalità standard. (Prezzi di Power Apps | Microsoft Power Platform)[26](Microsoft, 2024).

Un'ulteriore possibilità, per quanto riguarda le licenze Power Apps, è rappresentata dai piani per app (4,50€-5,50€ per utente/app/mese).

I piani per app di Power Apps offrono un modo flessibile per le aziende di utilizzare applicazioni specifiche senza dover acquistare licenze per ogni singolo utente. Questo tipo di piano è particolarmente utile per scenari aziendali in cui solo alcune applicazioni sono necessarie per determinati utenti o team. Con il piano per app, le aziende possono pagare per l'utilizzo di una singola applicazione o portale in un ambiente specifico, riuscendo così di ottimizzare i costi e le risorse.

Una volta acquistate le licenze, queste devono essere allocate agli ambienti tramite l'interfaccia di amministrazione di Power Platform. Questo processo consente alle

aziende di gestire facilmente le licenze e di assicurarsi che le applicazioni siano disponibili solo per gli utenti designati.

Licenze Power BI

Power BI offre una gamma di opzioni di licenza progettate per soddisfare esigenze diverse, con piani che spaziano dalla versione gratuita a opzioni più avanzate come le licenze Pro e Premium. Queste licenze variano in base al tipo di utilizzo previsto, alla capacità di condivisione e collaborazione, e all'accesso a funzionalità avanzate. Con la licenza gratuita, gli utenti possono connettersi ai dati e creare report all'interno di un'area di lavoro personale, ma non hanno la possibilità di condividere contenuti con altri né di pubblicarli in aree di lavoro comuni. Si tratta di un'opzione limitata, ideale per chi vuole esplorare le funzionalità base di Power BI senza un investimento economico.

La licenza Pro, invece, è una soluzione a pagamento pensata per utenti che necessitano di funzionalità collaborative. Ad un costo tra i 9€ e i 10€ al mese per utente, consente di condividere dashboard e report con altri utenti dotati della stessa licenza e di pubblicare contenuti in aree di lavoro condivise. Tuttavia, i limiti di capacità elaborativa e archiviazione restano ancorati a una dimensione condivisa e non esclusiva.

Per un accesso più completo e avanzato, la licenza Premium per utente (PPU) rappresenta un'opzione più potente, ad un costo compreso tra 15€ e 20€ al mese per utente. Questa licenza estende le funzionalità della versione Pro, consentendo l'accesso a modelli di dimensioni maggiori (fino a 100 GB), aggiornamenti più frequenti dei dataset, report impaginati, *pipeline* di distribuzione, intelligenza artificiale avanzata (come Auto-ML).

Tuttavia, i contenuti creati con una licenza PPU possono essere condivisi solo con altri utenti dotati della stessa licenza, salvo l'utilizzo di un'area di lavoro ospitata

su una capacità Premium.

La licenza Premium per *capacity*, infine, rappresenta una soluzione scalabile per grandi organizzazioni, con un costo variabile tra i 4500€ e i 5000€ al mese per singola *capacity*. Questa licenza elimina la necessità di licenze personali per ogni utente, rendendo i contenuti accessibili anche agli utenti con licenza gratuita[27](Datamaze, 2023).

Tipo di licenza	Funzionalità quando l'area di lavoro è nella capacità condivisa	Funzionalità aggiuntive quando l'area di lavoro è in capacità Premium
Fabric (gratuito)	Accesso al contenuto creato per se stessi.	Utilizzare il contenuto condiviso con loro dagli utenti Pro o PPU
Power BI Pro	Publicare contenuti in altre aree di lavoro, condividere i dashboard, sottoscrivere dashboard e report, condividere con gli utenti che hanno una licenza Pro	Distribuire il contenuto agli utenti con licenze gratuite o PPU
Power BI Premium per utente (PPU)	Publicare contenuti in altre aree di lavoro, condividere i dashboard, sottoscrivere dashboard e report, condividere con gli utenti che hanno una licenza Pro	Distribuire il contenuto agli utenti con licenze gratuite e Pro

Figura 4.4. Confronto tra funzionalità in base al tipo di licenza

È importante sottolineare che i prezzi indicati sono indicativi e possono variare in base alla dimensione dell'azienda, alle partnership esistenti con Microsoft (licenze già possedute) e alla possibilità di acquistare licenze da rivenditori autorizzati che spesso offrono sconti su volumi significativi.

Inoltre, le licenze di Power Apps e Power Automate possono essere già incluse in alcuni pacchetti Microsoft, rendendo i costi più contenuti per aziende che già utilizzano tali servizi. Microsoft offre una gamma di piani progettati per soddisfare le esigenze delle grandi imprese, che includono le licenze per Power Apps, Power

Automate e altri applicativi Microsoft (prezzo indicativo: tra i 7€ e 8€ per utente/mese).

Questi piani risultano particolarmente utili per le aziende che già possiedono le licenze Microsoft di base (Word, Excel, Power Point, Teams), in quanto consentono di integrare funzionalità aggiuntive, come gli applicativi della Power Platform, a costi contenuti.

Da sottolineare che, in alcuni pacchetti specifici, è inclusa anche la licenza Power BI Pro, che permette alle aziende di accedere alla piattaforma di *Business Intelligence* senza costi aggiuntivi per questa specifica licenza.

Stima dei costi di licenza per il progetto

Nel contesto del progetto in questione, l'azienda ha potuto sfruttare in modo strategico le licenze di SharePoint e Teams già in suo possesso, ottimizzando l'integrazione con gli strumenti di Microsoft Power Platform. Essendo il *tool* sviluppato per la gestione e il monitoraggio dei progetti destinato all'uso interno da parte di un numero ristretto di utenti, le licenze richieste per il suo funzionamento sono limitate e i relativi costi contenuti.

Alla luce di tali considerazioni, è possibile stimare i costi mensili di licenza per l'utilizzo di Microsoft Power Platform nell'ambito del progetto.

Innanzitutto, sono previste due licenze Microsoft per grandi imprese, con un costo indicativo compreso tra i 7€ e 8€ per utente al mese (dato che l'azienda soddisfa i requisiti per essere considerata grande impresa). Queste licenze includono l'accesso a Power Apps Premium, che permette di creare e distribuire applicazioni senza limiti e integrare fino a sei flussi di lavoro di Power Automate per applicazione, utilizzando il trigger "Quando Power Apps chiama il flusso". Inoltre, forniscono l'accesso alla versione Premium di Power Automate, consentendo di automatizzare processi aziendali avanzati.

Oltre a questi pacchetti, sarà necessario acquistare anche un piano per applicazione Power Apps, al costo compreso tra 4,50€ e 5,50€ per utente al mese. Questo piano consente di concedere la licenza a utenti selezionati, che possono quindi accedere e modificare l'applicazione Power Apps, inizialmente divisa in due sezioni (gestione progetti e risorse) e successivamente unificata.

In aggiunta, sono necessarie 6 licenze Power BI Pro, con un costo compreso tra 9€ e 10€ per utente al mese, per consentire ai 6 responsabili di accedere, condividere e visualizzare i report contenenti il Capacity Plan e i grafici di monitoraggio dei KPI di progetto. Considerando che il numero di utenti destinati a visualizzare i report è limitato, non sarà necessario acquistare licenze Premium per Capacity, poiché la licenza Pro è sufficiente per le esigenze del progetto.

In sintesi, i costi mensili stimati per le licenze saranno i seguenti: due licenze Microsoft per grandi imprese, con un costo totale compreso tra 14€ e 16€ al mese (7€-8€ ciascuna); una licenza Power Apps per applicazione, con un costo mensile compreso tra 4,50€ e 5,50€; e sei licenze Power BI Pro, con un costo totale compreso tra 54€ e 60€ al mese (9€-10€ ciascuna). Pertanto, il costo complessivo mensile stimato per le licenze necessarie al progetto sarà compreso tra 72,50€ e 81,50€, tenendo conto di tutte le componenti.

Licenza	Prezzo Mensile per Utente (Range)	Descrizione
Licenza Microsoft per grandi imprese	€7,00 - €8,00	Include Power Apps Premium e Power Automate Premium, fornendo gli strumenti per la creazione e la gestione delle applicazioni e dei flussi.
Power Apps (Piano per App)	€4,50 - €5,50	Permette agli utenti selezionati di accedere e modificare l'app Power Apps integrata (gestione progetti e risorse, con futura unificazione dell'app).
Power BI Pro	€9,00 - €10,00	Necessaria per i responsabili per condividere e visualizzare i report Power BI, come Capacity Plan e KPI, con supporto per l'uso di filtri avanzati.
Totale mensile per licenze	€72,50 - €81,50	Totale mensile per 2 licenze Microsoft per grandi imprese, 1 piano per app e 6 licenze Power BI Pro.

Figura 4.5. Stima Costo Licenze Progetto

4.4 Confronto tra Low-Code e Custom Development

Lo sviluppo di applicazioni aziendali richiede una scelta strategica tra soluzioni *low code*, come Microsoft Power Platform, e approcci personalizzati. Lo sviluppo personalizzato, *custom development*, si riferisce alla creazione di software progettato e sviluppato interamente su misura per soddisfare esigenze specifiche. Questo approccio consente alle organizzazioni di avere il controllo completo su ogni aspetto del sistema, dalle funzionalità alla gestione dei dati, fino all'integrazione con altre applicazioni.

Un esempio di una soluzione *custom* è un sistema CRM su misura, in grado di sincronizzarsi nativamente con strumenti come Azure DevOps o SAP, garantendo un'esperienza totalmente adattata alle specifiche esigenze aziendali.

Nel confronto con le piattaforme *low code*, che offrono rapidità e semplicità d'uso grazie a componenti predefiniti, lo sviluppo *custom* si distingue per la flessibilità e la libertà di personalizzazione, pur richiedendo tempi e costi iniziali più elevati. Questa sezione analizza i vantaggi e svantaggi di entrambi gli approcci, con particolare riferimento al progetto descritto nel Capitolo 3.

Le dimensioni chiave prese in esame sono:

1. Velocità di Sviluppo
2. Flessibilità e Personalizzazione
3. Costi di Sviluppo, Manutenzione e Licenza
4. Integrazione con Sistemi Esistenti
5. Scalabilità e Performance
6. Diritti di Proprietà e Controllo

7. Coinvolgimento degli Utenti Non Tecnici

1. Velocità di Sviluppo

Le piattaforme *low code*, come Power Apps e Power Automate, offrono un'elevata velocità di sviluppo. Nel progetto, è stato possibile creare rapidamente un'applicazione per il caricamento delle offerte e la gestione iniziale delle risorse. Utilizzando strumenti *drag-and-drop* e componenti predefiniti, l'implementazione di base ha richiesto solo poche settimane. Questo approccio si è dimostrato particolarmente utile nelle fasi iniziali, consentendo di avviare rapidamente le operazioni e testare il sistema con un *Minimum Viable Product* (MVP).

Tuttavia, quando sono state necessarie personalizzazioni più complesse, come la gestione delle relazioni multi-a-molti tra risorse e opportunità, i tempi di sviluppo si sono allungati a causa della necessità di configurare soluzioni come tabelle intermedie in SharePoint o integrare strumenti esterni come Power BI.

Al contrario, uno sviluppo personalizzato avrebbe richiesto più tempo, con una fase iniziale di analisi, prototipazione e sviluppo che avrebbe potuto protrarsi per diversi mesi.

Tuttavia, una volta completata, una soluzione *custom* avrebbe offerto un sistema ottimizzato e completamente adattato alle esigenze del progetto, eliminando la necessità di *workaround* e integrazioni complesse. Per esempio, le relazioni multi-a-molti e le colonne calcolate avrebbero potuto essere gestite direttamente nel database personalizzato, garantendo maggiore efficienza e semplicità di utilizzo.

2. Flessibilità e Personalizzazione

Le soluzioni *low code* offrono un livello di flessibilità limitato rispetto allo sviluppo personalizzato. Nel contesto del progetto, Power Apps ha mostrato restrizioni

significative, come l'impossibilità di creare grafici avanzati per il *capacity plan* direttamente nell'applicazione e di gestire colonne calcolate in tempo reale. Inoltre, i controlli predefiniti non permettono un'estetica completamente personalizzabile o l'implementazione di logiche avanzate direttamente all'interno dell'interfaccia.

Ad esempio, per rappresentare visivamente la disponibilità delle risorse, è stato necessario integrare Power BI per creare dashboard esterne, aggiungendo complessità e complicando l'esperienza utente. Questi limiti dimostrano come le piattaforme *low code* siano ottimali per esigenze standard ma meno adatte a scenari complessi. Lo sviluppo personalizzato offre totale libertà nella progettazione di funzionalità e logiche. Con un approccio *custom*, sarebbe stato possibile integrare grafici interattivi, colonne calcolate e una gestione nativa delle relazioni multi-a-molti direttamente nell'applicazione.

Inoltre, la personalizzazione visiva avrebbe permesso di rispettare completamente le linee guida aziendali, offrendo un sistema in linea con le aspettative.

3. Costi di Sviluppo, Manutenzione e Licenza

L'approccio *low code* ha consentito di contenere significativamente i costi iniziali del progetto, sfruttando licenze aziendali già disponibili, che includono gli applicativi base del pacchetto Office, oltre a Teams e Sharepoint.

Questo approccio ha permesso di ridurre al minimo l'investimento iniziale, fornendo al contempo strumenti efficaci per lo sviluppo e la gestione del *tool* interno, sfruttando la sinergia con gli applicativi di Microsoft Power Platform.

Nel caso del progetto in questione, i costi di licenza sono stati contenuti grazie al numero limitato di utenti coinvolti e all'utilizzo interno del *tool* (non devono essere acquistate licenze per i clienti) e alla possibilità di sfruttare i vantaggi garantiti da Microsoft alle grandi imprese.

Tuttavia, è fondamentale evidenziare che, nel caso in cui il numero di utenti coinvolti aumentasse significativamente e le applicazioni e i flussi dovessero essere condivisi con i clienti, i costi delle licenze crescerebbero considerevolmente.

Questo scenario è particolarmente critico per le piccole e medie imprese (PMI), che spesso non possiedono le licenze Microsoft di base o accordi di partnership con l'azienda e sono costretti quindi ad implementare l'intero ecosistema da zero. Per di più, quest'ultime non possono sfruttare i vantaggi offerti dai pacchetti Microsoft per le grandi organizzazioni e i relativi sconti legati a volumi significativi. In questi casi, l'acquisto di licenze aggiuntive, come quelle necessarie per utenti esterni o per l'accesso a funzionalità premium, potrebbe incidere pesantemente sul budget complessivo, rendendo meno vantaggioso l'approccio *low code*.

D'altro canto, una soluzione completamente personalizzata comporterebbe costi iniziali significativamente più elevati, richiedendo personale tecnico specializzato per la progettazione e lo sviluppo. Contestualmente, le tempistiche necessarie per implementare una soluzione su misura avrebbero ulteriormente incrementato l'investimento iniziale.

Per le soluzioni *custom*, una volta ultimato lo sviluppo, non vi sono costi di licenza ricorrenti e ciò potrebbe rappresentare un vantaggio economico nel lungo periodo. Ciononostante, queste soluzioni comportano inevitabilmente costi di aggiornamento e manutenzione, che devono essere gestiti regolarmente per garantire il corretto funzionamento del sistema e l'adeguamento a nuove esigenze aziendali o tecnologie. Questi costi, non previsti per le piattaforme *low code*, possono diventare significativi nel corso del tempo.

Un altro fattore da prendere in considerazione riguarda i costi di licenza, che nel caso di Microsoft Power Platform includono anche quelli relativi all'infrastruttura necessaria per eseguire le applicazioni, i flussi e i report, inclusi i server, la capacità di calcolo e lo storage necessari per l'elaborazione dei dati e il funzionamento delle app e dei flussi.

In una soluzione personalizzata, tali costi rappresenterebbero una spesa separata e rilevante, sia in termini di implementazione iniziale che di gestione continua. Questo aspetto rende la piattaforma *low code* di Microsoft particolarmente vantaggiosa, soprattutto per progetti di medio termine e ad utilizzo interno.

4. Integrazione con Sistemi Esistenti

Power Platform ha offerto una rapida integrazione con strumenti Microsoft come SharePoint e Teams. L'integrazione con Azure DevOps ha evidenziato, tuttavia, limiti significativi.

L'utilizzo di richieste HTTP personalizzate ha richiesto la generazione di *Personal Access Token* (PAT) con durata limitata, vincolando le richieste all'utente che ha generato il *token* e aumentando la complessità tecnica.

D'altra parte, una soluzione *custom* avrebbe offerto maggiore controllo sulle integrazioni, eliminando la necessità di dipendere da connettori predefiniti o *token* temporanei. Con un approccio personalizzato, sarebbe stato possibile gestire in modo nativo le connessioni a sistemi esterni come Azure DevOps, sfruttando strumenti di integrazione avanzati e *plugin* specifici (componenti software non autonomi, che aggiungono funzionalità a un programma esistente, permettendo di estendere o personalizzare le sue capacità). Questa soluzione avrebbe garantito una scalabilità maggiore, capace di supportare operazioni più complesse su larga scala, e avrebbe ridotto i rischi legati alla gestione dell'autenticazione.

5. Scalabilità e Performance

Nel caso specifico del progetto, l'utilizzo di SharePoint come fonte dati non ha generato problemi rilevanti di scalabilità e performance, poiché i volumi di dati

gestiti erano limitati e il numero di utenti coinvolti non ha comportato carichi eccessivi sul sistema.

Le operazioni di aggiornamento dati e i flussi Power Automate non hanno quindi evidenziato rallentamenti significativi. L'adozione degli applicativi di Power Platform e SharePoint ha consentito di mantenere performance soddisfacenti durante l'intero ciclo di sviluppo del progetto, senza necessità di ottimizzazioni avanzate o soluzioni di *scaling* (tecniche utilizzate per garantire che un'applicazione o un sistema possa crescere senza compromettere l'efficienza operativa o la qualità delle prestazioni).

Per quanto riguarda progetti caratterizzati da un maggiore coinvolgimento di utenti e un volume di dati più elevato, l'utilizzo di Power Platform integrato con SharePoint potrebbe presentare problemi di performance e scalabilità.

Le applicazioni sviluppate con Power Apps, quando connesse a fonti di dati come SharePoint, possono subire cali di prestazioni all'aumentare del numero di record e di operazioni richieste. La velocità di caricamento, l'interazione con i dati e la reattività dell'applicazione possono essere compromesse, specialmente quando le *query* su grandi dataset non sono ottimizzate. La funzione *Search ()*, utilizzata per ricercare tramite *ComboBox* un elemento di una lista, risulta non ottimizzata per dataset di grandi dimensioni, richiedendo perciò all'utente di limitare il campo di ricerca utilizzando dei filtri.

In relazione ai flussi Power Automate, anche in questo caso la scalabilità e le performance possono essere influenzate dal volume dei dati elaborati o dal numero di flussi in esecuzione contemporaneamente.

A causa delle limitazioni delle API o della capacità dei connettori di gestire operazioni simultanee su larga scala, il sistema può rallentare, con impatti sui tempi di esecuzione e sull'affidabilità dei flussi. Nel caso di soluzioni personalizzate, l'implementazione di un sistema scalabile e ottimizzato fin dall'inizio consente di affrontare carichi elevati e grandi volumi di dati in modo più efficiente.

Sfruttando database relazionali progettati per gestire quantità significative di informazioni, le soluzioni *custom* offrono maggiore controllo sulle prestazioni e sulla gestione delle risorse. In questo modo, è possibile progettare l'architettura in modo da supportare operazioni complesse e ottimizzare i tempi di risposta, riducendo i rischi di rallentamenti o problemi di scalabilità che potrebbero emergere in un ambiente con una crescente mole di utenti o di dati.

6. Diritti di Proprietà e Controllo

Il diritto di proprietà è un aspetto cruciale nella scelta tra sviluppo personalizzato e soluzioni *low code*. Nel caso dello sviluppo personalizzato, uno dei principali vantaggi è il completo controllo su tutti gli aspetti del sistema creato. Le aziende che seguono tale approccio di sviluppo tradizionale non dipendono da soluzioni esterne e hanno la possibilità di personalizzare ogni singolo dettaglio del software. Anche quando si utilizzano *framework* o librerie esterne, il prodotto finale rimane uno sviluppo proprietario, che può essere protetto tramite diritti di proprietà intellettuale, come i brevetti o i diritti d'autore, e può essere utilizzato esclusivamente dall'azienda o venduto come strumento esclusivo. Questo garantisce una totale autonomia nel gestire il software nel lungo termine.

D'altra parte, le soluzioni *low code* offrono comunque un sufficiente livello di personalizzazione, ma sono limitate dalla tecnologia e dai moduli predefiniti messi a disposizione dalla piattaforma. Sebbene sia possibile creare applicazioni completamente funzionali con tali strumenti, la possibilità di scaricare il codice sorgente del prodotto finale e trasferirlo su altre piattaforme o *hosting* dipende dalle politiche del fornitore della soluzione.

In molti casi, le piattaforme *low code* non permettono un trasferimento completo del codice sorgente, limitando la libertà dell'azienda di prendere il software e implementarlo su un'altra infrastruttura o adattarlo a necessità future senza dipendere

dal fornitore della piattaforma stessa. Di conseguenza, il controllo sul prodotto finale è meno flessibile rispetto allo sviluppo personalizzato, e le aziende potrebbero trovarsi vincolate ai limiti imposti dalla piattaforma *low code* utilizzata.

Tali vincoli appaiono evidenti nel caso specifico di Microsoft Power Platform, come evidenziato nella sezione 4.3.3.

7. Coinvolgimento degli Utenti Non Tecnici

Nelle soluzioni sviluppate su misura, il coinvolgimento degli utenti aziendali è generalmente limitato, in quanto il processo di sviluppo è principalmente gestito dal team tecnico. Gli utenti finali spesso partecipano solo in fase di raccolta dei requisiti iniziali e nel test finale del sistema.

Questo approccio garantisce funzionalità affidabili e versatili, ma può comportare un disallineamento tra la soluzione finale e le reali esigenze degli utilizzatori finali, in quanto la comunicazione tra team tecnico e aziendale risulta indiretta e centralizzata.

Al contrario, nelle soluzioni *low code*, come nel caso presentato, basato sulla Power Platform di Microsoft, gli utilizzatori finali (anche non tecnici) ricoprono un ruolo molto più attivo.

Poiché le piattaforme *low code* sono progettate per essere facilmente utilizzabili anche da utenti non tecnici, i responsabili aziendali possono partecipare direttamente allo sviluppo delle applicazioni, contribuendo con le loro conoscenze e intuizioni sui processi aziendali. Questo coinvolgimento diretto consente di raccogliere in modo più preciso le esigenze aziendali, migliorando l'accuratezza e la personalizzazione della soluzione.

Inoltre, è possibile un'implementazione più agile delle modifiche, poiché i feedback degli utenti finali sono più immediati e facilmente integrabili. In questo contesto, il coinvolgimento degli utenti diventa una risorsa fondamentale per l'efficacia della

soluzione, poiché la piattaforma consente iterazioni rapide e un allineamento costante tra tecnologia e business [28](Medium, 2021).

Qui di seguito viene riportata una tabella di confronto tra l'approccio *custom* e *low code* per la realizzazione di un sistema CRM [29] (Medium, 2024).

Confronto	CRM Tradizionale	CRM Low-Code/No-Code
Tempo di sviluppo e distribuzione	Cicli di sviluppo lunghi, solitamente diversi mesi o più, ma con sistemi più completi e altamente personalizzati.	Tempi di distribuzione più brevi, solitamente completati in giorni o settimane, ideale per progetti da avviare rapidamente.
Investimento economico	Costi iniziali di sviluppo e manutenzione elevati, adatti a imprese con budget sufficienti e obiettivi di stabilità a lungo termine.	Costi inferiori, adatti a PMI o team con budget limitati, riducendo la barriera tecnica.
Capacità di personalizzazione	Offre un alto grado di personalizzazione e integrazione profonda con le esigenze aziendali specifiche, ma richiede supporto da sviluppatori professionisti.	Personalizzazione flessibile, che consente agli utenti di apportare modifiche rapide senza programmare, anche se le personalizzazioni complesse potrebbero essere limitate dalla piattaforma.
Coinvolgimento degli utenti	Principalmente gestito dal team tecnico con un coinvolgimento limitato del team aziendale, ma con funzionalità professionali solide.	I team aziendali possono partecipare direttamente allo sviluppo, migliorando l'accuratezza e l'implementazione delle esigenze aziendali.
Scalabilità e integrazione	Forti capacità di scalabilità e integrazione, adatte ai requisiti complessi delle grandi imprese, con un'ampia gamma di opzioni di integrazione di terze parti.	Scalabilità raggiunta attraverso <i>plugin</i> integrati e strumenti di integrazione, sebbene il supporto per operazioni complesse su larga scala possa essere limitato.
Aggiornamenti e manutenzione	Stabilità del sistema elevata, adatta a imprese che cercano manutenzione e stabilità a lungo termine, ma con costi di aggiornamento e manutenzione elevati.	Aggiornamenti e manutenzione semplici, con piattaforme che spesso si aggiornano automaticamente, rendendolo adatto per esigenze di iterazione rapida, sebbene i cicli di aggiornamento della piattaforma possano imporre limitazioni.

Figura 4.6. Confronto CRM *custom* e CRM *Low-code/No-code*

Capitolo 5

Conclusioni

Lo sviluppo e l'adozione delle piattaforme *low code* rappresentano un cambiamento significativo nel panorama tecnologico contemporaneo, offrendo soluzioni che uniscono semplicità d'uso e rapidità di implementazione.

In un contesto caratterizzato dalla necessità di rispondere rapidamente alle sfide del mercato, queste tecnologie si configurano come strumenti strategici per accelerare la digitalizzazione e ridurre il *time-to-market* delle applicazioni aziendali, offrendo un accesso più ampio alla creazione di soluzioni digitali.

Nel processo di sviluppo vengono infatti coinvolti non solo gli sviluppatori qualificati, ma anche i cosiddetti *citizen developers*, ossia utenti con competenze tecniche limitate, che possono comunque contribuire al processo di digitalizzazione dei processi aziendali.

Piattaforme come Microsoft Power Platform, Appian e Mendix stanno evolvendo rapidamente, integrando l'intelligenza artificiale generativa ad ambienti dedicati alla semplificazione del processo di design delle interfacce e al potenziamento dell'automazione dei flussi di lavoro.

Parallelamente, grazie all'introduzione di API predefinite, connettori e strumenti di integrazione nativi, tali piattaforme hanno ampliato il proprio raggio di azione, semplificando l'integrazione con i sistemi aziendali esistenti.

Lo sviluppo del *Tool* integrato per la gestione dei progetti e delle risorse mediante la Microsoft Power Platform ha rappresentato un'occasione per esplorare questo approccio alternativo rispetto al tradizionale sviluppo *custom*.

Il progetto descritto nell'elaborato ha permesso di mettere in luce i principali vantaggi delle soluzioni *low code*, tra cui la rapidità di apprendimento, di implementazione e la riduzione dei costi iniziali, fattori che si sono rivelati cruciali per una rapida prototipazione e la digitalizzazione dei processi aziendali.

La possibilità di sfruttare applicativi come Power Apps e Power Automate ha accelerato la realizzazione delle funzionalità di base, permettendo un'interazione frequente con gli utilizzatori finali e adattamenti rapidi in corso d'opera.

In particolare, Power Apps ha consentito la creazione di applicazioni personalizzate utilizzando un'interfaccia *drag-and-drop*, senza necessità di scrivere codice complesso, facilitando così la progettazione di interfacce utente intuitive.

Power Automate, grazie a un ampio catalogo di connettori predefiniti, ha migliorato l'efficienza dei processi di caricamento, aggiornamento dei dati ed approvazione, automatizzando i flussi di lavoro e integrando facilmente applicativi Microsoft disponibili in azienda, come SharePoint e Teams.

Power BI è stato invece utilizzato per la definizione del *Capacity Plan* e per la creazione di dashboard interattive per il monitoraggio di progetti e risorse, consentendo di applicare filtri avanzati per analizzare e visualizzare i dati in modo dettagliato.

È importante evidenziare, inoltre, che Power Apps, Power Automate e Power BI sono nativamente integrabili tra loro, creando un ecosistema coeso che facilita la gestione dei dati, l'automazione dei processi e la visualizzazione dei KPI d'interesse. Tuttavia, nonostante i vantaggi sottolineati, l'utilizzo diretto di Power Platform ha posto in evidenza anche alcuni limiti.

La personalizzazione può risultare limitata, impedendo di adattare completamente le soluzioni a esigenze specifiche o a design particolarmente complessi. Le proprietà modificabili dei componenti grafici in Power Apps sono, in alcuni casi, limitate,

mentre le azioni che non sono previste dai connettori predefiniti di Power Automate comportano difficoltà significative nella configurazione, richiedendo soluzioni alternative come la definizione di API per interagire con piattaforme come Azure DevOps.

In aggiunta, l'integrazione con applicativi esterni e l'esportazione delle soluzioni al di fuori dell'ecosistema Microsoft può risultare complessa, essendo Power Platform progettata per funzionare in modo ottimale all'interno di tale ambiente, sfruttando le connessioni con altri strumenti come Office 365, SharePoint e Dynamics 365.

Alla luce di quanto sopra, l'approccio *low code* risulta particolarmente adatto a progetti di piccola e media complessità, in cui è fondamentale ottenere risultati rapidamente e con budget contenuti. Questo lo rende ideale per startup, piccole e medie imprese, e team aziendali che desiderano digitalizzare processi operativi o automatizzare attività standard, senza dover investire pesantemente in infrastrutture IT, risultando particolarmente efficace per creare prototipi, *Minimum Viable Product* (MVP) o applicazioni per la gestione di flussi di lavoro e dati condivisi, come moduli per la raccolta di informazioni, processi di approvazione o automazioni di base.

D'altro canto, questo approccio si dimostra meno adatto nei contesti in cui il numero di utenti, coinvolti nell'utilizzo delle soluzioni implementate, è significativo, visto l'impatto crescente dei costi di licenza sul bilancio complessivo del progetto. Nell'ambito di tali progetti, le piccole e medie imprese difficilmente possiedono le licenze di base e possono beneficiare degli sconti sui volumi o dei pacchetti di licenze pensati dai *vendor* per le grandi organizzazioni, costituendo così una barriera all'ingresso per l'adozione delle tecnologie *low code*.

Ad ogni modo, le limitazioni di personalizzazione e scalabilità rendono questo approccio meno indicato per organizzazioni con requisiti tecnici avanzati o che necessitano di supportare ampie basi di utenti e volumi significativi di dati, dove soluzioni più flessibili e su misura risultano più sostenibili nel lungo termine.

Le soluzioni *custom* garantiscono una flessibilità praticamente illimitata, permettendo di realizzare applicazioni su misura che rispondono perfettamente a esigenze specifiche e garantiscono il pieno controllo su ogni aspetto tecnico e funzionale.

Tale approccio risulta ideale per progetti complessi e di lunga durata, dove la scalabilità, le performance e l'integrazione avanzata sono requisiti fondamentali. In compenso, i tempi di sviluppo dilatati e i costi elevati di sviluppo, aggiornamento e manutenzione lo rendono meno adatto a realtà con budget limitati o che necessitano di soluzioni rapide e facilmente adattabili.

L'analisi condotta evidenzia come la scelta tra *low code* e sviluppo *custom* non sia una decisione assoluta, ma dipenda da una valutazione approfondita delle specifiche caratteristiche e priorità aziendali.

Le piattaforme *low code* sono ideali per le aziende che possiedono un ecosistema di applicativi facilmente integrabile con esse e nei contesti che richiedono rapidità, iterazione continua e costi iniziali contenuti.

Al contrario, lo sviluppo personalizzato si rivela più adatto a progetti strategici con requisiti tecnici avanzati, volumi elevati di dati, necessità di ampia personalizzazione, integrazione su larga scala e controllo totale sulle soluzioni sviluppate.

In conclusione, le piattaforme *low code* rappresentano una risposta efficace alle esigenze di digitalizzazione moderna, ma richiedono un approccio strategico per sfruttarne appieno il potenziale, minimizzandone le criticità. In alcuni casi, un approccio ibrido, che combini la rapidità e l'accessibilità del *low code* con la flessibilità e la robustezza dello sviluppo personalizzato, può rappresentare la soluzione ottimale.

Questo equilibrio consente di rispondere in modo agile alle sfide immediate, garantendo al contempo la sostenibilità e la scalabilità delle soluzioni a lungo termine, in un panorama tecnologico in continua evoluzione.

Sitografia

- [1] Oraichain Academy. *"The Brief History of Code Languages: Road to No-code"*.
URL:<https://academy.orai.io/road-to-no-code>, 2024.
(Ultimo Accesso 23/11/24).
- [2] Clay Richardson. *"New Development Platforms Emerge For Customer-Facing Applications"*.
URL:<https://www.forrester.com/report/microservices>, 2014.
(Ultimo Accesso 25/11/24).
- [3] Alvaro Llobet (Finboot). *"Blockchain Technology and Low-Code Platforms, a Perfect Match"*.
URL:<https://www.finboot.com/post/blockchain-and-low-code>, 2022.
(Ultimo Accesso 25/11/24).
- [4] kovaion.com. *"How AI-Powered Low-Code Platforms Are Transforming Innovation?"*.
URL:<https://www.kovaion.com/blog/ai-powered-low-code-platforms>, 2024.
(Ultimo Accesso 26/11/24).
- [5] Forbes Bernard Marr. *"The 10 Best Examples Of Low-Code And No-Code AI"*.
URL:<https://www.forbes.com/10-examples-of-low-code-and-no-code-ai>, 2022.
(Ultimo Accesso 26/11/24).

- [6] Fortune Business Insights. *"Low Code Development Platform Market Size, Share [2032]"*.
URL:<https://www.fortunebusinessinsights.com/low-code-market>, 2024.
(Ultimo Accesso 26/11/24).
- [7] Gartner. *"Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 20% in 2023"*.
URL:<https://www.gartner.com/forecasts-low-code-technologies>, 2022.
(Ultimo Accesso 27/11/24).
- [8] Mendix. *"What Is Mendix? How Does It Support the App Development Lifecycle?"*.
URL:<https://www.mendix.com/what-is-mendix>, 2024.
(Ultimo Accesso 03/12/24).
- [9] Pretius. *"Mendix platform 101: What is it, why is it interesting, and how much does it cost?"*.
URL:<https://pretius.com/blog/mendix-platform/>, 2024.
(Ultimo Accesso 04/12/24).
- [10] Mendix. *"Mendix Low-Code Platform Features - Low-Code App Development Tools"*.
URL:<https://www.mendix.com/platform/>, 2024.
(Ultimo Accesso 03/12/24).
- [11] Low code Guru. *"The Evolution of Appian: A Journey Through the History of a Low-Code Pioneer"*.
URL:<https://low-code.guru/the-evolution-of-appian->, 2023.
(Ultimo Accesso 04/12/24).
- [12] Appian Documentation. *"Introduction to Application Building in Appian - Appian 24.3"*.

- URL:<https://docs.appian.com/24.3/application-building>, 2024.
(Ultimo Accesso 05/12/24).
- [13] Appian Documentation. *"Integration Tutorial - Appian 24.3"*.
URL:<https://docs.appian.com/24.3/Integration-Tutorial>, 2024.
(Ultimo Accesso 05/12/24).
- [14] Appian Documentation. *"Automation - Appian 24.3"*.
URL:<https://docs.appian.com/24.3/Automation>, 2024.
(Ultimo Accesso 04/12/24).
- [15] Appian Documentation. *"AI Skill Object - Appian 24.4"*.
URL:<https://docs.appian.com/24.4/ai-skill-object>, 2024.
(Ultimo Accesso 04/12/24).
- [16] Microsoft Learn. *"Introduction to Microsoft Power Platform for developers"*.
URL:<https://learn.microsoft.com/en-us/power-platform/developer>, 2024.
(Ultimo Accesso 05/12/24).
- [17] Desk365. *"A Beginner's Guide to Microsoft Power Automate in 2024"*.
URL:<https://www.desk365.io/blog/microsoft-power-automate-guide>, 2024.
(Ultimo Accesso 05/12/24).
- [18] Microsoft Learn. *"Microsoft Power Fx overview - Power Platform"*.
URL:<https://learn.microsoft.com/en-us/power-platform/power-fx>, 2024.
(Ultimo Accesso 04/12/24).
- [19] Microsoft Learn. *"Get started with formulas in canvas apps"*.
URL:<https://learn.microsoft.com/power-apps/canvas-apps/formulas>, 2024.
(Ultimo Accesso 04/12/24).

- [20] Medium. *"Power Apps: Model-driven App Overview and Tutorial"*.
URL:<https://medium.com/microsoft-power-apps/model-driven-app>, 2023.
(Ultimo Accesso 05/12/24).
- [21] Microsoft Learn. *"Overview of building a model-driven app with Power Apps"*.
URL:<https://learn.microsoft.com/power-apps/model-driven-apps>, 2024.
(Ultimo Accesso 05/12/24).
- [22] Microsoft Learn. *"What is Power BI Desktop?"*.
URL: <https://learn.microsoft.com/power-bi/power-bi-desktop>, 2024.
(Ultimo Accesso 06/12/24).
- [23] Addend Analytics. *"All You Need to Know About Power BI Mobile App"*.
URL: <https://addendanalytics.com/blog/power-bi-mobile>, 2024.
(Ultimo Accesso 06/12/24).
- [24] Akveo. *"What Low-Code and DevOps Can Deliver Together: Key Benefits for Your Business"*. URL: <https://www.akveo.com/blog/low-code-devops>, 2024.
(Ultimo Accesso 6/12/24).
- [25] Microsoft. *"Prezzi di Power Automate"*.
URL:<https://www.microsoft.com/power-automate/pricing>, 2024.
(Ultimo Accesso 10/01/25).
- [26] Microsoft. *"Prezzi di Power Apps"*.
URL:<https://www.microsoft.com/power-apps/pricing>, 2024.
(Ultimo Accesso 10/01/25).
- [27] Datamaze. *"Costi e licenze di Power BI"*.
URL:<https://www.datamaze.it/blogs/post/costi-e-licenze-di-power-bi>, 2023.
(Ultimo Accesso 08/01/25).

[28] Medium. *"Low-Code vs Custom Development: What Are the Real Differences?"*.

URL:<https://www.medium.com/low-code-vs-custom-development-differences>,
2021.

(Ultimo Accesso 13/01/25).

[29] Medium. *"Build CRM in Hours: Top No-Code/Low-Code Tools You Need to Know"*.

URL:<https://www.medium.com/build-crm-low-code-custom>, 2024.

(Ultimo Accesso 13/01/25).

Bibliografia

- [1] Vincent Bushong, Amr S Abdelfattah, Abdullah A Maruf, Dipta Das, Austin Lehman, Eric Jaroszewski, Michael Coffey, Tomas Cerny, Karel Frajtak, Pavel Tisnovsky, and Miroslav Bures. *"On Microservice Analysis and Architecture Evolution: A Systematic Mapping Study"*. *Appl. Sci. (Basel)*, 11(17):7856, 2021.