

POLITECNICO DI TORINO

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA GESTIONALE

TESI DI LAUREA MAGISTRALE

---

**Creazione di un sistema di integrazione  
per il trasferimento di un file di paghe e  
note spese da OneDrive a Dynamics 365  
for Finance and Operations tramite Logic  
Apps**



**Politecnico  
di Torino**

*Candidato:*  
Beatrice MAINI

*Relatrice:*  
Prof.ssa Tania CERQUITELLI  
*Tutor Aziendale:*  
Alessandra CORONA

Anno accademico 2021/2022



# Sommario

La seguente tesi si prefigge l'obiettivo finale di descrivere il progetto realizzato durante il periodo di tirocinio iniziato nel mese di marzo 2022 protrattosi fino ad oggi. L'attività di tirocinio è stata svolta presso l'azienda di consulenza informatica Alternativa SRL. La presente tesi è sviluppata in 7 capitoli, di cui viene di seguito proposta una disamina.

- Nel capitolo 1 è stata inserita una descrizione dell'azienda in cui si è svolto il tirocinio e la seguente stesura della tesi.
- Nel Capitolo 2 viene introdotto il concetto di Microsoft Azure Services per far comprendere il contesto in cui si è operato; viene poi affrontata una descrizione più dettagliata delle Azure Logic Apps analizzandone lo scopo, i vantaggi e il funzionamento.
- Nel Capitolo 3 viene discusso in un contesto generico il tema di Dynamics 365 for Finance and Operations e del corrispondente ambiente di sviluppo. Ne sono stati analizzati gli aspetti teorici necessari alla comprensione della realizzazione del progetto.
- Nel Capitolo 4 viene descritto il processo creativo alla base del progetto.
- Nel Capitolo 5 è stata presentata l'analisi dei dati iniziali e la progettazione della prima applicazione per la logica necessaria alla prima parte del processo. Viene poi trattata l'implementazione del codice all'interno di Visual Studio fondamentale alla successiva elaborazione dei dati. Infine, è stata descritta la realizzazione della seconda applicazione per la logica, funzionale alla concretizzazione della parte finale del processo.
- Nel capitolo 6 è stata realizzata una guida step by step per la messa in pratica del processo realizzato.
- Nel Capitolo 7 vengono tratte delle conclusioni sul lavoro svolto e suggeriti dei possibili sviluppi futuri.



# Ringraziamenti

Innanzitutto, vorrei ringraziare la Prof.ssa Tania Cerquitelli per avermi dato la possibilità di svolgere questa tesi e per gli utili consigli.

Un ringraziamento va anche alla mia tutor aziendale, Alessandra Corona, per l'aiuto e il sostegno ricevuto, che sono stati fondamentali per la riuscita di questo progetto.

Un ringraziamento speciale va alla mia famiglia che mi ha sempre sostenuta nei momenti di difficoltà e che mi ha permesso di raggiungere questo traguardo.

Ringrazio il mio fidanzato, che mi ha sopportata e supportata dall'inizio di questo percorso universitario fino ad oggi.

Infine, ringrazio le mie amiche e i miei colleghi per l'affetto e l'aiuto che mi hanno dato in questi lunghi anni.



# Contents

<b>Sommario</b>	<b>iii</b>
<b>Ringraziamenti</b>	<b>v</b>
<b>1 Introduzione</b>	<b>1</b>
1.1 Descrizione dell'azienda . . . . .	1
<b>2 Logic Apps</b>	<b>3</b>
2.1 Azure Integration Services . . . . .	3
2.2 Overview sulle Logic Apps . . . . .	4
2.2.1 I connettori di Azure Logic Apps . . . . .	6
Connettori integrati per servizi e sistemi specifici . . . . .	10
2.2.2 Pianificazione per trigger ricorrenti nei flussi di lavoro di App per la logica di Azure . . . . .	12
<b>3 Dynamics 365 for Finance and Operations</b>	<b>15</b>
3.1 Introduzione a Dynamics 365 for Finance and Operations . . . . .	15
La trasformazione digitale . . . . .	15
Finance and Operations apps . . . . .	16
3.2 Ambiente di sviluppo . . . . .	17
3.2.1 Visual Studio 2019 . . . . .	17
3.3 Stato dell'arte . . . . .	18
3.3.1 Panoramica delle organizzazioni e delle gerarchie organizzative Squadre . . . . .	18
3.3.2 Scritture contabili . . . . .	21
3.3.3 Le dimensioni finanziarie . . . . .	26
3.3.4 Business Events . . . . .	32
<b>4 Metodologia</b>	<b>35</b>
4.1 Overview del progetto . . . . .	35
<b>5 Sperimentazione</b>	<b>37</b>
5.1 Analisi dei file inviati dall'azienda cliente . . . . .	37
5.2 Creazione delle tabelle . . . . .	39
5.3 Creazione della prima logic app per l'importazione del file . . . . .	42
5.4 Implementazione . . . . .	52
5.4.1 Creazione SysOperationFramework . . . . .	52
All'interno della classe <i>LedgerImpPayrollService_ALTC</i> . . . . .	53
Metodo <i>createDefaultDimension</i> . . . . .	62
5.4.2 Creazione business event . . . . .	65
5.5 Creazione della seconda Logic App . . . . .	70
<b>6 Risultati Sperimentali</b>	<b>75</b>

<b>7</b>	<b>Conclusione</b>	<b>81</b>
<b>8</b>	<b>Bibliografia</b>	<b>83</b>



# List of Figures

2.1	PowerApps data platform . . . . .	5
2.2	Connettori 1 . . . . .	9
2.3	Connettori 2 . . . . .	10
2.4	Connettori 3 . . . . .	10
2.5	Connettori 4 . . . . .	11
2.6	Connettori 5 . . . . .	11
3.1	Visual Studio [14]. . . . .	17
3.2	Persona Giuridica [16]. . . . .	18
3.3	Giornale contabile. . . . .	22
3.4	General Journal Processing [17]. . . . .	22
3.5	General Journals [17]. . . . .	22
3.6	Schema di stato patrimoniale. . . . .	24
3.7	Schema di conto economico. . . . .	24
3.8	Schema di conto. . . . .	25
3.9	Dimension Attribute . . . . .	27
3.10	Query SQL che indica le informazioni di base relative a ciascuna di- mensione . . . . .	27
3.11	Elenco di valori fornito da BusinessUnit . . . . .	28
3.12	Pagina Ledger contenente le account structure relative alla company ITCO. . . . .	28
3.13	Esempio di un account structure. . . . .	29
3.14	FinancialDimensions. . . . .	29
3.15	Tabella DimensionAttributeValueSetItem. . . . .	30
3.16	Casella Account. . . . .	30
3.17	Combinazione dimensioni finanziarie. . . . .	30
3.18	Main Account. . . . .	31
3.19	Main Account Type. . . . .	31
3.20	BusinessEventCatalog. . . . .	32
3.21	Business events activation. . . . .	33
4.1	Schema del processo. . . . .	36
5.1	File paghe/note spese da importare. . . . .	37
5.2	Anagrafica dei conti. . . . .	38
5.3	File Excel dati azienda cliente. . . . .	38
5.4	LedgerImpPayrollDecodeCompany_ALTC. . . . .	39
5.5	LedgerImpPayrollDecodeAnalytic_ALTC. . . . .	40
5.6	LedgerImpPayrollDecodeAccountSimpleList_ALTC. . . . .	40
5.7	LedgerImpPayrollStatusTable_ALTC. . . . .	41
5.8	Trigger OneDrive for Business. . . . .	42
5.9	Directory Files contenente il file di paghe e note spese . . . . .	42
5.10	Action per inizializzazione della variabile FileName. . . . .	43

5.11	Action per inizializzazione della variabile FileContent. . . . .	43
5.12	Lista valori da immettere. . . . .	43
5.13	Operazione di substring per estrazione del nome. . . . .	44
5.14	Substring applicata sul nome del file. . . . .	44
5.15	Action per inizializzazione della variabile External CompanyId. . . . .	45
5.16	LedgerImpPayrollDecodeCompany_ALTC. . . . .	45
5.17	Action per il recupero del codice della Company. . . . .	46
5.18	Action per inizializzazione della variabile Company ID. . . . .	47
5.19	LedgerImpPayrollStatusTable_ALTC. . . . .	47
5.20	Action per la creazione di un record della tabella LedgerImpPayroll- StatusTable_ALTC. . . . .	48
5.21	Action per inizializzazione della variabile Processing Status. . . . .	49
5.22	Action per verifica della condizione di uguaglianza. . . . .	49
5.23	Flusso in caso di risposta affermativa. . . . .	50
5.24	Flusso in caso di risposta negativa. . . . .	51
5.25	Flusso in caso di risposta negativa pt.2. . . . .	51
5.26	Esempio di testata del giornale contabile. . . . .	57
5.27	Esempio di General Journal. . . . .	64
5.28	Esempio di giornale. . . . .	64
5.29	Business Event . . . . .	69
5.30	Trigger creazione business event. . . . .	70
5.31	Schema JSON. . . . .	71
5.32	Action per inizializzazione del ProcessingStatus. . . . .	72
5.33	Condizione per il controllo degli errori. . . . .	72
5.34	Ramo del flusso in caso di risposta affermativa. . . . .	73
5.35	Ramo del flusso in caso di risposta negativa. . . . .	74
5.36	Ramo del flusso in caso di risposta negativa pt.2. . . . .	74
6.1	Step 1 - Creazione del file in OneDrive. . . . .	75
6.2	Step 2 - Esecuzione prima logic app. . . . .	75
6.3	Step 3 - Corretta esecuzione della prima logic app. . . . .	76
6.4	Step 4 - LedgerImpPayrollStatusTable_ALTC . . . . .	76
6.5	Step 5 - LedgerImpPayrollActionMenuItem. . . . .	77
6.6	Step 5 - LedgerImpPayrollActionMenuItem pt.2. . . . .	77
6.7	Step 6 - Corretta esecuzione della seconda logic app. . . . .	78
6.8	Step 7 - LedgerImpPayrollStatusTable_ALTC. . . . .	78
6.9	Step 8 - General Journals. . . . .	78
6.10	Step 6 - Giornale contabile realizzato. . . . .	79

# List of Tables

3.1	Le diverse tipologie di unità operative, una loro descrizione e il loro obiettivo.[15]	20
-----	--	----



## Chapter 1

# Introduzione

### 1.1 Descrizione dell'azienda

Le attività di tirocinio e di stesura della tesi sono state svolte all'interno dell'azienda Alterna SRL.

Alterna SRL è una società di consulenza, System Integrator e Partener Microsoft facente parte della rete di società di consulenza aziendale a capo della quale vi è Altea Federation [1]. Quest'ultima ricopre il ruolo di direzione amministrativa e controllo delle venti aziende che fanno parte del gruppo.

Altea Federation collabora con i leader mondiali dell'innovazione tecnologica attraverso un modello di business omonico-virtuale che raccoglie sotto un unico marchio mondi diversi ma complementari che si avvicinano alle sfide dettate dai nuovi paradigmi in ambito Consulting, Technology, Digital e Operations. Altea offre servizi legali e di sviluppo delle risorse umane. Eroga servizi di consulenza e formazione per progetti in ambito sicurezza, ambiente e qualità. Supporta imprese federate e clienti nell'analisi, fruizione e gestione di finanziamenti e incentivi.

Alterna [2], supporta la crescita e la competitività delle imprese accompagnandole nel loro percorso di trasformazione digitale proponendo un portfolio di offerte che abbraccia tutte le soluzioni tecnologiche e applicative Microsoft. Le certificazioni Microsoft ottenute negli anni spaziano su tutta l'offerta del vendor. Tra le certificazioni principali si possono trovare le competenze Microsoft Partner in ambito ERP (Microsoft Dynamics 265 F&SCM e Microsoft Dynamics 365 Business Central), Microsoft Dynamics 365 Customer Engagement (CRM), Cloud Platform, Productivity, Collaboration & Content, Datacenter e ISV.

L'offerta di Alterna si articola in quattro macrogruppi: Dynamics 365, Data e AI, Microsoft Azure, Smart Working. Alterna lavora in un contesto data driven oriented dove diventa necessario analizzare una gran mole di dati. Oggi Microsoft Dynamics 365 rappresenta un nuovo approccio alle applicazioni aziendali intelligenti. Microsoft Dynamics 365 combina il meglio dell'offerta Microsoft ERP e CRM: integra applicazioni di business intelligenti e abilita le aziende a crescere, evolvere e trasformarsi attraverso nuove soluzioni aziendali quali: Business Central, Finance & Supply Chain Management, Sales, Customer Service, Marketing, Field Service, Project Service Automation e Talent. Sviluppata seguendo un approccio modulare, la nuova piattaforma cloud garantisce alle imprese massima flessibilità e personalizzazione sfruttando tutte le potenzialità dello stack tecnologico e applicativo Microsoft.

Con Dynamics 365 è possibile semplificare le operazioni di business, ridurre i costi e aumentare l'efficienza. Microsoft Dynamics 365, con una user experience familiare, integra molteplici applicazioni di business per rispondere in modo personalizzato alle esigenze di specifici ruoli professionali, funzioni e settori.

La piattaforma di Cloud Computing Microsoft Azure è la soluzione cloud di Microsoft che soddisfa le richieste delle imprese più esigenti e offre un ambiente potente e flessibile per guidarle nel proprio percorso di trasformazione. Potendo contare su un modello Hyperscale con oltre 100 datacenter nel mondo, interconnessioni ad alta velocità e costi a consumo, Microsoft Azure rappresenta una scelta di sicuro interesse per tutti i tipi di aziende, dalla startup a una Fortune 500. La proposta a valore ha diversi aspetti, che vanno da risparmi significativi rispetto a un approccio tradizionale, alla capacità di creare rapidamente applicazioni robuste e resilienti, che possono scalare in base alle reali esigenze di ogni impresa. Per quanto riguarda lo Smart Working YouSMART è la nuova piattaforma di Smart Working realizzata da Alterna e basata su tutto lo stack tecnologico Microsoft, capace di sfruttare le potenzialità di Office365. Grazie a Microsoft SharePoint e OneDrive si possono organizzare le informazioni e condividerle.

## Chapter 2

# Logic Apps

### 2.1 Azure Integration Services

Sempre più aziende negli ultimi anni stanno investendo in nuove infrastrutture e servizi cloud-computing. Questa scelta è guidata prima di tutto da motivazioni economiche in quanto, tramite i servizi cloud, queste smettono di investire nella costruzione di costose infrastrutture fisiche e devono preoccuparsi soltanto del costo delle risorse consumate.

I servizi di cloud-computing ovviamente sono resi possibili dagli sviluppatori che, programmando in maniera efficace, riescono a rendere disponibili le risorse.

In questo scenario di evoluzione del cloud-computing, uno dei protagonisti principali è stato sicuramente Microsoft, che ha creato il suo servizio cloud chiamato *Azure* ricco di servizi e piattaforme messe a disposizione dei propri utenti.

Quando si parla di un'architettura serverless viene da sé che uno dei principali vantaggi è sicuramente il fatto di potersi concentrare maggiormente sull'innovazione nel settore e sul proprio business, ma non solo; infatti, tramite questa tipologia di servizio è possibile:

- realizzare dei pattern reattivi asincroni che permettono, ad esempio, di mettere in esecuzione un'applicazione al verificarsi di un evento;
- scrivere codice con un linguaggio a scelta dell'utente senza restrizioni di alcun tipo;
- ridurre il consumo delle risorse, minimizzando i costi dell'azienda grazie ad un modello dei prezzi basato sul consumo;
- coprire un'area di business molto ampia, riducendo i problemi di user experience e soddisfacendo le esigenze e le richieste di utenti posizionati in diverse aree del mondo;
- essere sempre aggiornati sui cambiamenti e le innovazioni all'interno del mercato grazie alle ottime capacità di monitoraggio;

All'interno di *Azure Integration Services* vi sono diverse piattaforme come Azure Logic Apps, Service Bus, Azure Event Grid, e API Management. Quindi, Azure mette a disposizione un ventaglio di servizi cloud che pongono al centro l'utente e coprono un range di settori che spaziano dall'ambito sanitario e automobilistico, fino ad arrivare a quello governativo e assicurativo.[3]

## 2.2 Overview sulle Logic Apps

*Azure Logic Apps*, come detto nell'introduzione, fa parte del pacchetto di Azure Integration Services, questa piattaforma permette di creare dei *flussi di lavoro* automatizzati scrivendo poche righe di codice.

Le app per la logica [4] consentono inoltre di:

- automatizzare attività ripetitive come l'immissione di dati;
- fornire funzionalità di integrazione con applicazioni Web e mobile, e API;
- creare un'integrazione con applicazioni locali e aziendali;
- creare un'integrazione con app e servizi di terze parti tramite connettori.

Un esempio di attività che le applicazioni per la logica riescono ad automatizzare, potrebbe essere l'invio di una mail tramite Office 365 ogni qual volta si verifica un determinato evento, come può essere la creazione di un file.

Azure Logic Apps presenta diversi vantaggi tra cui:

- il fatto di possedere degli strumenti di progettazione facili da usare, che permettono di progettare applicazioni ricche di connettori e di azioni;
- rende semplici le connessioni con un gran numero di sistemi, che richiederebbe altrimenti l'implementazione di codice altamente complesso ed elaborato;
- nel caso in cui un connettore non sia disponibile tra quelli standard è possibile creare un'applicazione API e utilizzarla come connettore.

Le logic apps hanno un sistema di prezzi basato sulla logica del consumo, perciò non hanno costi correnti o costi di infrastruttura e vengono addebitati solo i costi delle azioni che vengono eseguite [5].

All'interno della Logic App è possibile utilizzare trigger e azioni predefinite per poter progettare un flusso di lavoro, collegarsi con un endpoint di servizio, mettere in esecuzione il codice o modificare i dati.

Di seguito verranno descritti quelli che sono i termini chiave per comprendere i concetti contenuti nel servizio Logic Apps.

### Workflow

Un flusso di lavoro è un insieme di fasi che definiscono un'attività o un processo specifico. Ogni flusso di lavoro inizia con un singolo trigger, seguito da una o più azioni.

### Trigger

Il passo iniziale di un qualsiasi flusso di lavoro è sempre un trigger, che stabilisce la condizione per l'esecuzione di tutte le azioni successive in quel flusso di lavoro. Un evento di attivazione può essere, ad esempio, la ricezione di un'e-mail nella casella di posta o l'identificazione di un nuovo file in un account di archiviazione.



### Action

Ogni fase di un flusso di lavoro che segue l'innesco viene definita azione.

Ogni azione di un flusso di lavoro esegue un'operazione.

### Operazioni integrate

Un'azione o un trigger integrato è un'attività nativa di Azure Logic Apps.

Ci sono operazioni che non fanno parte di alcun sistema mentre altre, invece, si trovano all'interno di specifici servizi; per questo motivo, in alcuni casi, prima di poterne usufruire, è necessario creare una connessione e autenticarsi.

Tramite le operazioni integrate è possibile controllare la pianificazione o la struttura del flusso di lavoro, eseguire il proprio codice, gestire e modificare i dati, inviare o ricevere richieste a un endpoint e svolgere altre attività nel flusso di lavoro.

### Gestire i connettori

Per poter accedere ad una applicazione o ad un servizio è necessario un *connettore gestito*. Questo non è altro che un proxy o wrapper realizzato per un'API REST.

Come per le operazioni integrate, per poter utilizzare un connettore gestito è richiesta una connessione dal proprio flusso di lavoro e l'autenticazione della propria identità. Microsoft si occupa di pubblicare, ospitare e gestire questi connettori.

### Account di integrazione

Quando si desidera definire e memorizzare artefatti B2B (Business-To-Business) da utilizzare nei flussi di lavoro, si crea un account di integrazione in Azure. I flussi di lavoro possono utilizzare questi artefatti B2B dopo aver creato e collegato un account di integrazione all'applicazione logica.

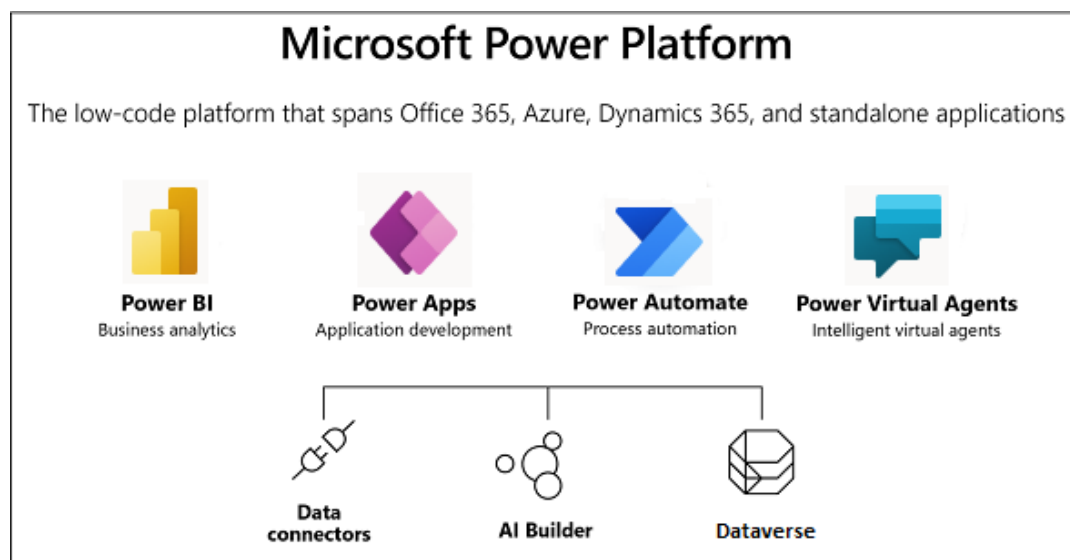


FIGURE 2.1: Microsoft Power Platform.

## Come funzionano le Logic Apps

Per capire il funzionamento delle app per la logica si parte dal primo elemento facente parte del workflow che è stato analizzato sopra, ossia il trigger. In seguito al soddisfacimento di una condizione infatti, il trigger viene attivato e questo permette di mettere in esecuzione il flusso. Quando si parla di *soddisfacimento di una condizione* si intende ad esempio il verificarsi di un evento o magari dati che soddisfano un determinato requisito.

Nel caso in cui si volesse decidere ogni quanto eseguire un flusso, è possibile pianificare l'esecuzione tramite appositi trigger che offrono opzioni di pianificazione.

Una volta che il flusso è iniziato, si possono verificare elaborazioni o modifiche sui dati che passano attraverso il flusso di lavoro. Queste operazioni sono gestite dalle azioni del flusso.

## Perché usare le app per la logica

Quando viene creata un'app per la logica il processo è piuttosto semplice e non richiede di scrivere codice elaborato.

Nel caso in cui fosse necessario scrivere del codice, è possibile utilizzare Azure Functions per creare frammenti di codice che possono essere eseguiti dal flusso di lavoro. Utilizzando l'azione *Inline Code*, è possibile anche creare snippet di codice da eseguire nel flusso di lavoro. Infine, viene messo a disposizione **Azure Event Grid** per monitorare, instradare e pubblicare eventi.

Come è stato ampiamente spiegato nell'introduzione, il vantaggio di utilizzare Azure Logic Apps sta proprio nel fatto di poter creare servizi e applicazioni "serverless", perciò ci si può concentrare esclusivamente sulla logica e sulle funzionalità aziendali. Infatti, questa tipologia di servizi consente di creare applicazioni cloud complesse con un codice minimo e riescono ad adattarsi automaticamente alle esigenze aziendali [6].

### 2.2.1 I connettori di Azure Logic Apps

Azure Logic Apps mette a disposizione centinaia di connettori attraverso i quali è possibile accedere con rapidità e facilità a dati, eventi e risorse di altre applicazioni, servizi, sistemi, protocolli e piattaforme quando si progettano flussi di lavoro.

E' importante sottolineare come tutto questo sia reso possibile senza la necessità di creare complesse righe di codice ma semplicemente inserendo un connettore nel flusso; sarà poi quest'ultimo che metterà a disposizione varie funzioni predefinite che possono essere sfruttate come azioni lungo il processo.

Nel caso in cui tra i tanti connettori già messi a disposizione da Microsoft non vi fosse quello necessario a collegarsi con la risorsa a cui si desidera accedere, è possibile comunicare con il servizio utilizzando la procedura HTTP generica o creando un nuovo connettore personalizzato.

### Cosa sono i connettori

Dal punto di vista tecnico un connettore è un proxy o un wrapper attorno a un'API che il servizio sottostante utilizza per comunicare con Azure Logic Apps.

Nel caso in cui si volesse accedere ad un'operazione del flusso, sotto forma di azione o di trigger, potrebbe essere necessaria una connessione al servizio o al sistema sottostante, ad esempio per autenticare l'accesso a un account utente e questo viene reso possibile proprio dal connettore[7].

Si va ora ad analizzare più nello specifico i trigger e le azioni.

### Triggers

La fase iniziale di qualsiasi processo è sempre definita da un trigger, che è l'evento che dà inizio al flusso di lavoro. I trigger possono avere diversi schemi di attivazione e a seconda dello schema si può comprendere il modo in cui il trigger monitora e risponde agli eventi.

Gli schemi più frequenti che un trigger può seguire sono lo schema polling o lo schema push, ma a volte sono presenti entrambe le varianti.

- I *trigger polling* monitorano un servizio o un sistema specifico su base regolare per vedere se ci sono nuovi dati o un evento. Questi trigger avviano ed eseguono una nuova istanza del flusso di lavoro ogni volta che sono disponibili nuovi dati o si verifica un evento specifico. Questa nuova istanza può quindi utilizzare i dati forniti come input.
- I *trigger push* monitorano i nuovi dati o eventi. Questi trigger costruiscono ed eseguono una nuova istanza del flusso di lavoro ogni volta che sono disponibili nuovi dati o si verifica un evento. Questa nuova istanza può quindi utilizzare i dati forniti come input.

Un esempio di modello polling potrebbe essere il seguente: nel caso in cui si volesse creare un flusso di lavoro che faccia qualcosa quando un file viene caricato su OneDrive, bisogna impostare come primo passaggio un trigger OneDrive denominato *Quando un file viene creato o modificato*. Utilizzando questo tipo di trigger quindi, si può pianificare ogni quanto verificare gli eventi di caricamento.

Una funzionalità molto utile dei trigger è il fatto di trasmettere i dati lungo tutto il flusso di lavoro e questo fa sì che le azioni successive possano accedere a quei dati e usufruirne. Ad esempio, si vuole utilizzare il trigger di Office 365 Outlook *Quando arriva una nuova e-mail* per avviare un flusso di lavoro ogni volta che si riceve una nuova e-mail. Questo trigger può essere configurato per trasmettere il contenuto di ogni nuova e-mail, come il mittente, l'oggetto, il corpo, gli allegati e così via. Altre azioni del flusso di lavoro possono essere utilizzate per elaborare queste informazioni [7].

### Actions

Un'azione è un'operazione che si svolge dopo il trigger e svolge un'attività nel processo di lavoro. All'interno del flusso di lavoro sono disponibili diverse azioni. Ad esempio, si può iniziare il flusso di lavoro con un trigger SQL che identifica i dati

di un nuovo cliente in un database SQL. Dopo il trigger, il flusso di lavoro può includere un'azione SQL che recupera i dati dei clienti. Dopo l'azione SQL, il flusso di lavoro può includere un'altra azione che elabora i dati, non necessariamente SQL [7].

### Categorie di connettori

All'interno delle Logic Apps esistono due categorie di connettori: i connettori predefiniti, che vengono eseguiti in modo nativo nel runtime di Azure Logic Apps e i connettori gestiti, che vengono distribuiti da Microsoft e vanno a fornire ai servizi cloud trigger e azioni [7].

### Configurazione della connessione

Molti connettori gestiti richiedono la connessione iniziale al servizio o al sistema di destinazione prima di poterne utilizzare i trigger o le azioni nel flusso di lavoro. Quindi bisogna convalidare la propria identità con le credenziali dell'account e con altri dettagli prima di poter effettuare una connessione all'interno della finestra di progettazione del flusso di lavoro dell'applicazione logica. Ad esempio, bisogna fare l'accesso all'account di posta elettronica di Office 365 Outlook prima che il flusso di lavoro possa accedervi e utilizzarlo [7].

### Sicurezza e crittografia della connessione

Le informazioni di configurazione della connessione, come l'indirizzo del server, il nome utente e la password, le credenziali e i segreti, sono crittografate e conservate nell'ambiente sicuro di Azure. Vi sono diversi controlli di accesso e soltanto i clienti che possiedono i diritti sulle risorse di connessione possono utilizzare queste informazioni.

Finché il servizio o il sistema di destinazione lo consente, le connessioni stabilite possono accedervi. Azure Logic Apps aggiorna i token di accesso a tempo indeterminato per i servizi che utilizzano connessioni OAuth di Azure AD, come Office 365 e Dynamics. Altrimenti potrebbero esserci dei fornitori che potrebbero imporre delle limitazioni su quanto tempo Logic Apps può utilizzare un token prima di doverlo aggiornare. Alcune azioni, come la modifica della password, rendono tutti i token di accesso non validi [7].

### Comportamento ricorrente

I trigger predefiniti ricorrenti, come il *Recurrence trigger*, vengono eseguiti in modo nativo nel runtime di Azure Logic Apps.

Quando si distribuisce o si salva l'app per la logica, viene immediatamente eseguita la ricorrenza; per questo motivo è consigliato impostare una data e un'ora di inizio per la prima ricorrenza [7].

### Connettori predefiniti nelle app per la logica di Azure

Con i connettori predefiniti, è possibile controllare la pianificazione e la struttura dei flussi di lavoro, eseguire codice, gestire o manipolare dati ed eseguire altre attività nei flussi di lavoro.

Tutti i connettori standard vengono eseguiti in modo nativo nel runtime dell'Azure Logic Apps e alcuni non hanno bisogno di creare una connessione per poter essere utilizzati. Solitamente una parte di questi connettori non è collegata a un servizio, sistema o protocollo specifico; cosa che li differenzia dai connettori gestiti. Per questo motivo si può attivare il flusso in due modi: o viene utilizzato il trigger *Recurrence* che fa iniziare il flusso tramite una precedente pianificazione o semplicemente si può lasciare che il flusso di lavoro attenda il trigger *Request*.

A seconda che si stabilisca un flusso di lavoro per app logiche di consumo in Azure Logic Apps multi-tenant o un flusso di lavoro per app logiche standard in Azure Logic Apps single-tenant, varia il numero e la tipologia di connettori integrati. Azure Logic Apps offre una versione integrata accanto alla versione gestita per un numero minore di servizi, sistemi e protocolli.

Inoltre, alcuni connettori predefiniti con attributi specifici nei flussi di lavoro dell'app per la logica standard, sono indicati in modo informale come provider di servizi. Alcuni connettori predefiniti offrono solo un metodo di autenticazione della connessione a un servizio remoto. Per la logica Azure, tutte le connessioni predefinite vengono eseguite nello stesso processo del runtime dell'applicazione [8].

### Connettori integrati generali

È possibile utilizzare i seguenti connettori integrati per eseguire attività generali, ad esempio:





			
<b>HTTP</b>	<b>Richiesta</b>	<b>Batch</b> (solo app per la logica di consumo)	
<b>Ricorrenza pianificata</b> : attiva un flusso di lavoro in base alla ricorrenza specificata.  <b>Finestra scorrevole</b> (solo app per la logica di consumo): attiva un flusso di lavoro che deve gestire i dati in blocchi continui.  <b>Ritardo</b> : sospende il flusso di lavoro per la durata specificata.  <b>Ritardo fino a</b> : sospende il flusso di lavoro fino alla data e all'ora specificate.	Chiamare un endpoint HTTP o HTTPS utilizzando il trigger o l'azione HTTP.  Puoi anche utilizzare questi altri trigger e azioni HTTP integrati: <ul style="list-style-type: none"> <li>• HTTP + spavalderia</li> <li>• HTTP + webhook</li> </ul>	<b>Quando viene ricevuta una richiesta HTTP</b> : Attendi una richiesta da un altro flusso di lavoro, app o servizio. Questo trigger rende il tuo flusso di lavoro richiamabile senza dover essere controllato o sottoposto a polling in base a una pianificazione.  <b>Risposta</b> : Rispondere a una richiesta ricevuta dal trigger <b>Quando viene ricevuta una richiesta HTTP</b> nello stesso flusso di lavoro.	<b>Messaggi batch</b> : attiva un flusso di lavoro che elabora i messaggi in batch.  <b>Invia messaggi in batch</b> : chiama un flusso di lavoro esistente che attualmente inizia con untrigger di <b>messaggi in batch</b> .

FIGURE 2.2: Connettori integrati generali [8].

## Connettori integrati per servizi e sistemi specifici

È possibile utilizzare i seguenti connettori integrati per accedere a servizi e sistemi specifici. Nei flussi di lavoro dell'app per la logica standard, alcuni di questi connettori predefiniti sono anche noti informalmente come provider di servizi, che possono differire in qualche modo dai connettori gestiti [8].

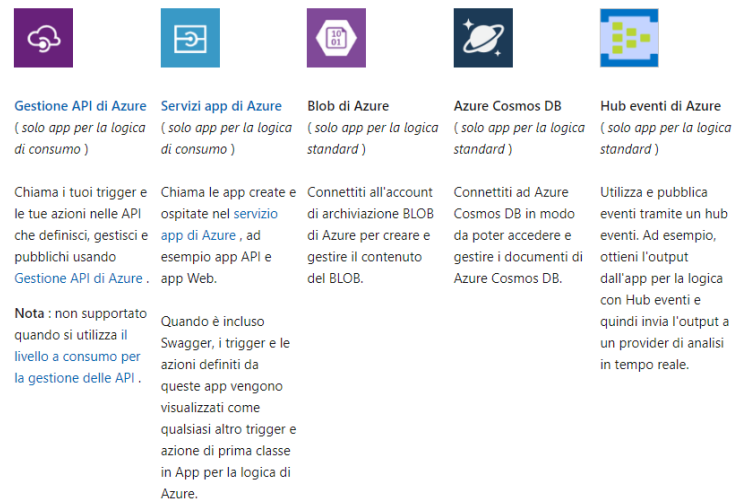


FIGURE 2.3: Connettori integrati per servizi e sistemi specifici pt.1 [8].



FIGURE 2.4: Connettori integrati per servizi e sistemi specifici pt.2 [8].

## Controllo del flusso di lavoro

App per la logica di Azure offre le seguenti azioni predefinite per strutturare e controllare le azioni nel flusso di lavoro:

### Controllo del flusso di lavoro

App per la logica di Azure offre le seguenti azioni predefinite per strutturare e controllare le azioni nel flusso di lavoro:



FIGURE 2.5: Controllo del flusso di lavoro [8].

## Gestire o manipolare i dati

Inoltre, Azure Logic Apps offre anche le seguenti azioni predefinite per l'utilizzo degli output di dati e dei relativi formati:

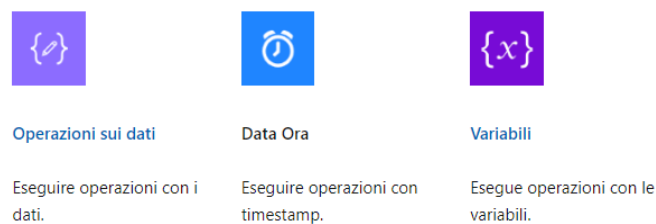


FIGURE 2.6: Connettori integrati generali [8].

### Connettori gestiti in App per la Logica di Azure

Nel caso in cui non siano disponibili i connettori integrati vengono messi a disposizione i connettori gestiti che consentono di collegarsi a servizi e sistemi. I connettori gestiti richiedono generalmente una connessione dal proprio flusso di lavoro e l'autenticazione. Questi trigger e azioni possono essere utilizzati per creare flussi di lavoro che combinano dati, applicazioni, servizi basati sul cloud e sistemi on-premise.

Come è stato detto anche per i connettori integrati, Azure Logic Apps offre una versione integrata accanto alla versione gestita per un numero minore di servizi, sistemi e protocolli. Inoltre, la quantità e la varietà di connettori integrati variano a seconda che si sviluppi un'applicazione logica di consumo per Azure Logic Apps multi-tenant o un'applicazione logica standard per Azure Logic Apps single-tenant. Nella maggior parte dei casi, la versione integrata offre prestazioni, funzionalità, prezzi superiori. Alcuni connettori integrati sono disponibili solo in un tipo di applicazione logica e non nell'altro.

Tutti i connettori gestiti in un'applicazione logica Standard sono organizzati nel gruppo Azure. I connettori gestiti in un'applicazione logica a consumo sono classificati come Standard o Enterprise.

- I connettori *Standard* forniscono l'accesso a servizi come Archiviazione BLOB di Azure, Office 365, SharePoint, Salesforce, Power BI, OneDrive e molti altri.
- I connettori *Enterprise* forniscono l'accesso ai sistemi aziendali, come SAP, IBM MQ e IBM 3270 a un costo aggiuntivo.

Alcuni connettori gestiti appartengono anche ai seguenti gruppi informali:

- I connettori locali forniscono l'accesso a sistemi locali come SQL Server, SharePoint Server, SAP, Oracle DB, condivisioni file e altri.
- I connettori di account di integrazione consentono di trasformare e convalidare XML, codificare e decodificare file flat ed elaborare messaggi business-to-business (B2B) utilizzando i protocolli AS2, EDIFACT e X12.
- I connettori dell'ambiente del servizio di integrazione sono progettati per essere eseguiti specificamente in un ISE e offrono vantaggi rispetto alle versioni non ISE [9].

### 2.2.2 Pianificazione per trigger ricorrenti nei flussi di lavoro di App per la logica di Azure

Come è stato già detto in precedenza, le app per la logica permettono di creare dei flussi di lavoro automatizzati e ricorrenti in base ad una pianificazione[10].

E' possibile creare dei flussi che iniziano tramite un *Recurrence trigger* o *Sliding Window trigger* che sono due esempi di trigger pianificati; tramite l'utilizzo di questi infatti, è possibile far eseguire le attività immediatamente, in un secondo momento oppure a intervalli ricorrenti.

Questo elenco mostra alcune attività di esempio che è possibile eseguire con i trigger pianificazione:



- Ottenere dati interni, ad esempio eseguire una stored procedure SQL ogni settimana.
- Ottenere dati esterni, come i bollettini meteorologici ogni 3 ore.
- Invio di dati del rapporto, ad esempio un'e-mail di riepilogo per tutti gli ordini superiori a un importo specifico nell'ultimo mese.
- Elaborazione dei dati, ad esempio comprimere le immagini caricate durante la settimana ogni settimana del mese durante le ore non di punta.
- Pulizia dei dati, ad esempio eliminare tutte le mail più vecchie di 6 mesi.
- Archiviazione dei dati, come ad esempio l'invio di documenti ad un servizio di backup alle 09:00 ogni giorno per i prossimi 3 mesi.

Si possono anche utilizzare le azioni integrate *Pianifica* per mettere in pausa il flusso di lavoro prima dell'esecuzione dell'azione successiva, ad esempio:

- Attendere fino a un giorno festivo per inviare un aggiornamento tramite e-mail.
- Ritardare il flusso di lavoro finché una chiamata HTTP non ha il tempo di terminare prima di riprendere e recuperare il risultato.

### Pianificazione dei trigger

I trigger *Ricorrenza* o *Finestra scorrevole* eseguono il flusso di lavoro in base all'intervallo di tempo e alla frequenza specificati, ad esempio il numero di secondi, minuti, ore, giorni, settimane o mesi. È inoltre possibile scegliere la data e l'ora di inizio, nonché il fuso orario. Di seguito vengono elencate le differenze tra i due trigger:

- **Ricorrenza:** esegue il flusso di lavoro a intervalli regolari a seconda della pianificazione scelta. Se ci sono, ad esempio, delle interruzioni o processi disabilitati, il trigger *Ricorrenza* non elabora le ricorrenze mancate, ma continua le ricorrenze con il successivo intervallo pianificato.  
Se si sceglie *Giorno* come frequenza, è possibile specificare le ore e i minuti dell'ora, ad esempio ogni giorno alle 18:00. Se si sceglie *Settimana* come frequenza, è possibile scegliere determinati giorni della settimana, come sabato. Si può anche specificare una data e un'ora di inizio insieme a un fuso orario per la pianificazione delle ricorrenze.
- **Sliding Window:** esegue il flusso di lavoro su base regolare e gestisce i dati in modo continuo. Se il trigger salta delle ricorrenze a causa di interruzioni o di flussi di lavoro disattivati, ad esempio, il trigger *Sliding Window* torna indietro ed elabora le ricorrenze mancate.  
Per posticipare ogni ricorrenza del flusso di lavoro, è possibile definire una data e un'ora di inizio, un fuso orario e una durata. Le pianificazioni avanzate, come ore specifiche del giorno, minuti dell'ora e giorni della settimana, non sono supportate da questo trigger [10].



## Chapter 3

# Dynamics 365 for Finance and Operations

### 3.1 Introduzione a Dynamics 365 for Finance and Operations

Le applicazioni aziendali Microsoft sono soluzioni intelligenti che forniscono una prospettiva completa delle operazioni di un'azienda. I prodotti Microsoft Dynamics 365, collegati da dati e intelligenza e supportati da Microsoft Power Platform, sono tra queste soluzioni.

Microsoft è riuscito ad unire il Customer Relationship Management (CRM) con l'Enterprise Resource Planning (ERP) nell'ambito di Dynamics 365 e Microsoft Power Platform. Tramite questa combinazione tecnologica ha fatto sì che i processi di creazione delle applicazioni e di distribuzione dei dati in tutte le applicazioni Dynamics 365 venissero semplificati.

Queste applicazioni creano interazioni continue, alimentate e informate da servizi intelligenti. Le aziende possono quindi offrire esperienze all'avanguardia e altamente personalizzate ai loro utenti, grazie a una visione a 360 gradi dei clienti e dei processi. Inoltre, essendo dotate di funzionalità analitiche, forniscono alle organizzazioni informazioni dettagliate su ogni area funzionale delle loro attività [11].

Le applicazioni aziendali di Microsoft [11] creano valore aggiunto ed effetti reali alle organizzazioni tramite:

- Le applicazioni multicanale che trasformano gli incontri una tantum in incontri ricorrenti.
- Servizi intelligenti che forniscono assistenza prescrittiva per migliorare i risultati aziendali.
- Piattaforma cloud che integra persone, processi e dati per una prospettiva a 360 gradi dei clienti e del business.
- Soluzioni flessibili che permettono alle organizzazioni di crescere grazie al cambiamento.

#### La trasformazione digitale

La trasformazione digitale [12] è l'integrazione della tecnologia digitale in tutte le aree di un'organizzazione. L'obiettivo della trasformazione digitale è cambiare il modo in cui le organizzazioni operano e forniscono valore ai loro clienti.

Molte organizzazioni sono riuscite a trasformarsi digitalmente proprio grazie alle applicazioni aziendali realizzate da Dynamics 365; infatti, queste applicazioni moderne e flessibili hanno permesso alle aziende di migliorare i loro risultati di business attraverso la tecnologia.

Inoltre, queste applicazioni sono state in grado di ridurre la complessità dei sistemi per la gestione delle relazioni con i clienti (CRM) e delle risorse aziendali (ERP).

Per creare innovazione e rivoluzionarsi le organizzazioni possono utilizzare le applicazioni aziendali Dynamics 365 per:

- Coinvolgere i clienti e costruire relazioni: ripensare il modo di interagire con i clienti creando esperienze personalizzate di marketing, vendita e assistenza clienti che utilizzano dati e intelligenza per migliorare ogni interazione.
- Migliorare le operazioni: migliorare il servizio, aumentare l'efficienza e ridurre i costi incorporando l'intelligence e le linee guida di prescrizione nei processi aziendali.
- Autorizzare i dipendenti: assumere, coinvolgere e liberare i migliori talenti affinché lavorino al meglio delle loro capacità con i dati e le informazioni disponibili sul posto di lavoro.
- Trasformare prodotti e servizi: utilizzare i dati come risorsa strategica per identificare nuove opportunità di mercato, sviluppare prodotti innovativi e creare esperienze eccezionali per i clienti con una visione completa dei clienti e delle operazioni.

### Finance and Operations apps

Le applicazioni di Dynamics 365 for Finance and Operations [13] si occupano di operazioni aziendali quali la gestione dell'inventario, l'acquisto e la vendita di merci e il controllo di tutte le attività finanziarie. Queste applicazioni fanno parte dell'Enterprise Resource Planning. Per la gestione delle operazioni quotidiane all'interno di un'azienda vengono utilizzate le 4 applicazioni seguenti:

- Dynamics 365 Commerce - Fornisce una soluzione multicanale completa che integra back-office, in-store, call center e interazioni digitali. Commerce consente di incrementare i ricavi grazie a una maggiore produttività del personale, di fidelizzare il marchio attraverso incontri personalizzati con i clienti, di ottimizzare le operazioni per ridurre i costi e di incrementare l'efficienza della supply chain.
- Dynamics 365 Finance - Contribuisce all'automazione e alla modernizzazione dei processi finanziari globali. Per alimentare la crescita aziendale, monitorare le prestazioni in tempo reale, prevedere i risultati futuri e prendere decisioni basate sui dati. Il settore finanziario può essere utilizzato per utilizzare l'intelligenza artificiale per guidare le decisioni finanziarie strategiche, integrare e automatizzare le operazioni finanziarie, minimizzare le spese operative e ridurre la complessità e il rischio finanziario globale.
- Dynamics 365 Human Resources - Automatizza diversi processi di assunzione del personale e snellisce diverse attività di registrazione.

La conservazione dei dipendenti, l'amministrazione dei benefit, la formazione, le valutazioni delle prestazioni e la gestione delle modifiche sono esempi di queste procedure. Human Resources fornisce anche una struttura per i dipendenti delle risorse umane per gestire le aree di supervisione.

- Dynamics 365 Supply Chain Management - Aiuta a trasformare le operazioni di produzione e supply chain. Utilizza l'intelligenza artificiale e l'Internet delle cose (IoT) per ottenere informazioni predittive e intelligence per la pianificazione, la produzione, l'inventario, il magazzino e la gestione dei trasporti. Il Supply Chain Management ha il potenziale per aumentare l'efficienza operativa, la qualità dei prodotti e la redditività. La gestione della catena di approvvigionamento può essere utilizzata per innovare con processi produttivi intelligenti, aggiornare la gestione del magazzino, ottimizzare le prestazioni di produzione, massimizzare la durata delle risorse e automatizzare e snellire la catena di approvvigionamento.

## 3.2 Ambiente di sviluppo

### 3.2.1 Visual Studio 2019

Per dare la possibilità agli sviluppatori di modificare il codice sorgente è stato introdotto un nuovo ambiente di sviluppo integrato (IDE) chiamato: *Visual Studio Code* [14].

Visual studio è un programma molto creativo che contiene un numero elevato di funzionalità; infatti, oltre a permettere di scrivere, debuggare e compilare codice, fornisce compilatori, strumenti di completamento del codice e anche finestre di grafica.

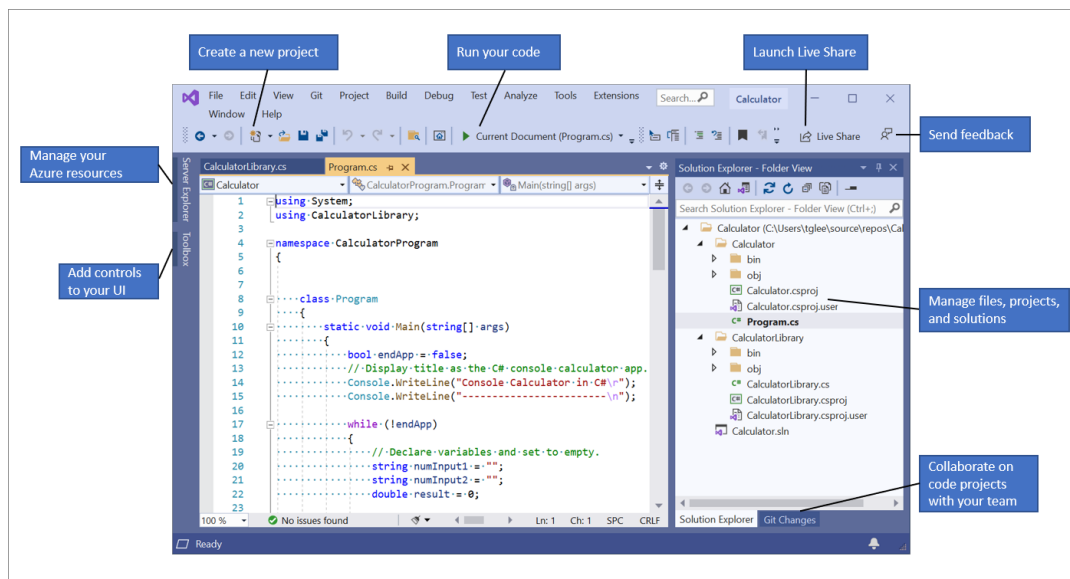


FIGURE 3.1: Visual Studio [14].

Nella figura 3.1 si possono evidenziare gli elementi principali, come:

- il *Solution Explorer*, posizionato in alto a destra nella schermata; permette di raggruppare i file in soluzioni e progetti, dà inoltre la possibilità di visualizzare, gestire e modificare i file.

- la *finestra di progettazione*, posizionata nella parte centrale della schermata; questa mostra il contenuto dei file e permette di modificare, scrivere codice o disegnare un'interfaccia utente

Il linguaggio del codice utilizzato per l'implementazione è il C++.

### 3.3 Stato dell'arte

#### 3.3.1 Panoramica delle organizzazioni e delle gerarchie organizzative

All'interno di una società, per poter raggiungere un determinato obiettivo aziendale, vengono creati dei gruppi di lavoro, detti organizzazioni[15]. Le organizzazioni possono essere di tre tipi:

1. persone giuridiche,
2. squadre,
3. unità operative.

#### Persone giuridiche

Si parla di *Legal Entity* o *Persona Giuridica*[16] quando ci si riferisce ad un'organizzazione che ha la possibilità di stipulare contratti legali e di rendicontare tutte le sue prestazioni. Le società sono proprio un esempio di legal entity e a ciascuna di esse viene associato un Company ID (o DataAreaID) che funge da meccanismo di protezione dei dati, in quanto ogni azienda ha la possibilità di accedere solo ai dati relativi alla propria società.

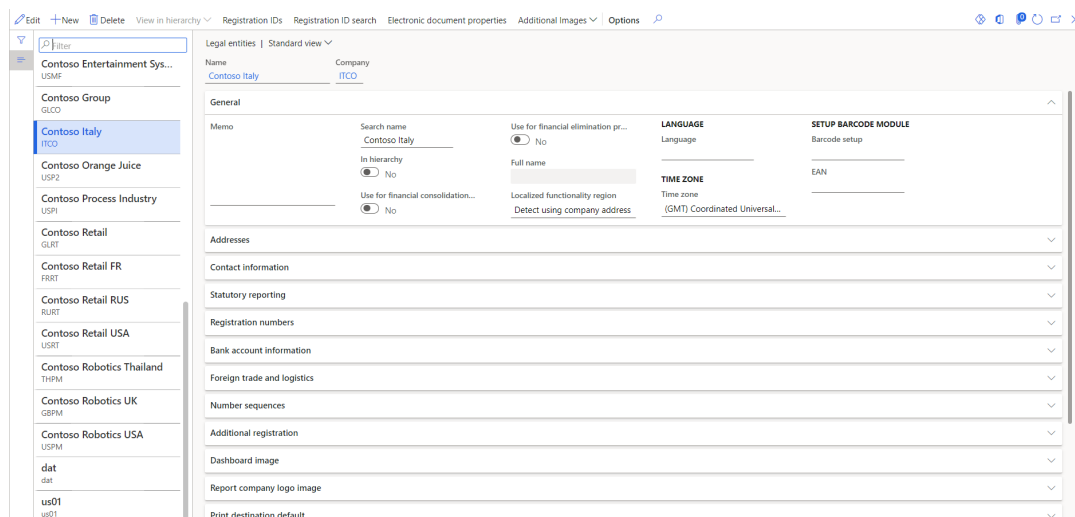


FIGURE 3.2: Persona Giuridica [16].

#### Squadre

Una squadra è un'organizzazione in cui i membri condividono una responsabilità, un interesse o un obiettivo comuni. I team non possono essere usati nelle gerarchie organizzative.

**Unità operative**

All'interno di un'azienda è necessario suddividere il monitoraggio delle risorse e dei processi. Per far sì che questo avvenga, vengono create le *operating units*, ossia organizzazioni in grado di massimizzare le performance di un'impresa. Ci sono diverse tipologie di operating units:

Tabella unità operative		
Unità operativa	Descrizione	Obiettivo
<b>Cost center</b>	Un'unità operativa in cui i manager sono responsabili delle spese preventivate ed effettive.	Utilizzato per la gestione e il controllo operativo dei processi aziendali che si estendono a più entità legali.
<b>Business unit</b>	Un'unità operativa semi-autonoma creata per raggiungere obiettivi aziendali strategici.	Si usa per il reporting finanziario basato su settori o linee di prodotto che l'organizzazione serve indipendentemente dalle entità legali.
<b>Value stream</b>	Un'unità operativa che controlla uno o più flussi di produzione.	Comunemente utilizzato nella produzione snella per controllare le attività e i flussi necessari per fornire un prodotto o un servizio ai consumatori.
<b>Department</b>	Un'unità operativa che rappresenta una categoria o una parte funzionale di un'organizzazione che svolge un compito specifico, come le vendite o la contabilità.	Si usa per fare report sulle aree funzionali. Un reparto può avere la responsabilità dei profitti e delle perdite e può essere costituito da un gruppo di centri di costo.
<b>Retail channel</b>	Un'unità operativa che rappresenta un negozio di mattoni e malta, un negozio online o un call center.	Viene utilizzata per la gestione e il controllo operativo di uno o più punti vendita all'interno di una o più entità legali.

TABLE 3.1: Le diverse tipologie di unità operative, una loro descrizione e il loro obiettivo.[15]



### 3.3.2 Scritture contabili

All'interno di Dynamics 365 Finance si possono esplorare i concetti relativi alle scritture contabili. L'obiettivo principale di una scrittura contabile è la registrazione di una transazione all'interno di un giornale contabile.

Un giornale [17] è un foglio di lavoro utilizzato per registrare transazioni all'interno del conto di contabilità generale o altri conti. Ogni qual volta vengono registrate nuove righe di transazioni all'interno del giornale, vengono create nuove voci all'interno dei conti di contabilità generale.

I giornali, se correttamente configurati, consentono di gestire le registrazioni in modo rapido ed efficiente.

Per utilizzare un giornale, è necessario prima creare un nome per il giornale e configurare il libro mastro, il giornale e le registrazioni del modulo General ledger.

Esistono diversi tipi di giornali, ad esempio ci sono giornali creati manualmente e altri generati automaticamente dal sistema. Inserendo il nome del giornale si indica il tipo di transazione, mentre il tipo di giornale va ad indicare dove può essere utilizzato un giornale di registrazione.

Prima di poter utilizzare i giornali di registrazione è necessario configurare varie componenti.

#### Elabora una registrazione a giornale

In un giornale generale [17], si inseriscono le informazioni relative alla transazione, come la data di registrazione, l'importo e i conti primari e di compensazione a cui la transazione deve essere registrata (Figura 3.3). Per creare una registrazione bisogna inserire le seguenti informazioni:

- Data di contabilizzazione
- Conto contabile
- Importo a debito o credito
- Conto di contropartita (può essere inserito sulla stessa riga o su più righe)

Nella pagina General journal processing (Figura 3.4) possono essere inseriti nuovi giornali di registrazione generale e possono essere visualizzati i giornali di registrazione esistenti.

**Journal voucher**

Display journal lines: All

List General Invoice Payment Payment fee Fixed assets Remittance History

+ New - Delete Voucher Financial dimensions Sales tax Functions

✓ Date	Voucher	Company	Account type	Account	Description	Debit	Credit	Offset company	Offset account type	Offset account	Offset transaction
2/20/2015	GNUL000003	usmf	Ledger	602180-002-026--	Benefits	2,751.84		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-002-025--	Benefits	305.76		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-002-024--	Benefits	4,892.16		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-002-023--	Benefits	19,262.88		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-002-022--	Benefits	3,363.36		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-001-026--	Benefits	4,127.76		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-001-025--	Benefits	458.64		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-001-024--	Benefits	7,338.24		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-001-023--	Benefits	28,894.32		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602180-001-022--	Benefits	5,045.04		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602120-002-026--	Payroll Tax	2,201.47		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602120-002-025--	Payroll Tax	244.61		usmf	Bank	USMF OPER	
2/20/2015	GNUL000003	usmf	Ledger	602120-002-024--	Payroll Tax	3,913.73		usmf	Bank	USMF OPER	

Account name: Pension/Profit-Sharin... Offset account name: Operating account - U... Calculated sales tax amount: 0.00 Actual sales tax amount: 0.00

CURRENCY: DEBIT 480,168.00 CREDIT 480,168.00 BALANCE 0.00 REPORTING CURRENCY: DEBIT 480,168.00 CREDIT 480,168.00 BALANCE 0.00

VOUCHER 480,168.00 480,168.00 0.00 480,168.00 480,168.00 0.00 Budget check results

FIGURE 3.3: Giornale contabile.

**General journal processing**

Summary

General journals

My not posted 0

My posted in current period 3

All not posted 0

My posted today 0

All posted today 0

My in review 0

My posted in previous period 0

All posted in previous period 0

Filter

Post Lines Validate Delete Open lines in Excel Workflow

Journal batch...	Name	Description	Log	Created by	In use by
00459	WFGenJm	WF General Journal		SARA	
00471	WFGenJm	WF General Journal		SARA	kweekley
00619	GenJm	General Journal		kweekley	

See more

Links

Analysis

Trial balance

Voucher transactions

Audit trail

Setup

Ledger calendars

Ledger currency exchange rates

Ledger reasons

Ledger account alias

Journal names

Journal descriptions

FIGURE 3.4: General Journal Processing [17].

L'intestazione del giornale è composta da diverse schede. Una volta che il giornale è stato registrato la casella controllo Posted viene selezionata automaticamente. Tramite il campo Show vengono filtrati i vari giornali (Figura 3.5).

**General journals**

Show: Posted Show user-created only

List General Setup Blocking Financial dimensions History

Journal batch...	Name	Description	Posted	Posted on	Log	L.	In use by	Reversing entry	Reversing date
00001	GenJm	Payroll - Jan	✓	11/28/2016 10:14:22 PM					
00002	GenJm	Payroll - Feb	✓	11/28/2016 10:22:01 PM					
00032	GenJm	Payroll - Mar	✓	11/28/2016 10:40:59 PM					
00043	GenJm	Payroll - Apr	✓	11/28/2016 10:44:20 PM					
00044	GenJm	Payroll - May	✓	11/28/2016 10:48:52 PM					
00045	GenJm	Payroll - Jun	✓	11/28/2016 10:51:52 PM					
00046	GenJm	Payroll - Jly	✓	11/28/2016 10:55:21 PM					
00047	GenJm	Payroll - Aug	✓	11/28/2016 10:59:03 PM					

FIGURE 3.5: General Journals [17].

La scheda General [18] mostra diverse informazioni, come descrizione, nome del giornale di registrazione, livello di registrazione, livello di dettaglio, limiti di riga e se il giornale di registrazione è per una registrazione di storno.

La scheda Setup mostra il tipo di conto, controlla se gli importi includono l'imposta sulle vendite e offre la possibilità di sostituire il tasso di cambio con un tasso fisso nel caso in cui utilizzi più valute. Permette inoltre, di impostare il conto di contropartita predefinito per un conto utilizzando la pagina dei nomi di registrazione. Creando un conto di contropartita predefinito, questo campo viene compilato automaticamente senza specificare il conto ogni volta.

La scheda Blocking mostra se questo giornale è stato bloccato da un errore, che attiverà l'opzione Bloccato dal sistema, o se è bloccato da un altro utente. In alternativa, puoi rendere privato il journal per un utente o un gruppo di utenti.

È possibile utilizzare le dimensioni finanziarie predefinite nella scheda Financial Dimensions. La cronologia mostra chi ha eseguito la manutenzione del diario.

Finance rileva il periodo fiscale quando si specifica la data nella prima nota. Se un periodo è chiuso o sospeso, l'utente riceverà un messaggio di errore che lo informa che non può utilizzare la data in quel periodo.

### **Libro mastro**

Ogni persona giuridica ha un proprio libro mastro, che può essere collegato a un piano dei conti. È possibile collegare più libri mastri allo stesso piano dei conti, in modo da poterlo condividere con più di una persona giuridica. È inoltre necessario specificare quale struttura o quali strutture di conto devono essere collegate al libro mastro per ogni persona giuridica.

Le transazioni vengono registrate in quello che viene definito giornale di registrazione o giornale di registrazione generale.

La registrazione all'interno del giornale avviene tramite il metodo della partita doppia e avviene in ordine cronologico[19][20].

### **Partita Doppia**

All'interno di un'azienda quando vi sono degli scambi tra clienti e fornitori è necessario regolarli per legge e per farlo si utilizza la contabilità generale. Quest'ultima è fondamentale per poter determinare il patrimonio netto al termine dell'esercizio.

Il metodo della partita doppia entra in gioco proprio nel momento in cui deve essere applicata la contabilità generale.

Tramite il metodo della partita doppia vengono registrate le operazioni aziendali in modo simultaneo su due conti; si basa, infatti, sul principio della duplice rilevazione simultanea.

Il fine primario della partita doppia è andare a controllare i movimenti monetari e finanziari relativi alla gestione e, allo stesso tempo, calcolare il reddito di un determinato periodo amministrativo.

Gli eventi amministrativi si dividono in due aspetti fondamentali che sono quello monetario-finanziario o numerario, e quello economico. Prendendo come esempio

la vendita di un bene, questa va a generare un incasso ossia un credito: in questo caso si parla di aspetto numerario; ovviamente, si sta generando anche un ricavo che rappresenta l'aspetto economico. Tramite il metodo della partita doppia è possibile registrare la duplice dimensione degli eventi aziendali ossia quella finanziaria e quella economica.

Un'operazione aziendale per essere contabilizzata deve coinvolgere un soggetto interno all'azienda e uno esterno, deve essere supportato da un documento giustificativo e deve aver generato una variazione di cassa.

La struttura della partita doppia è costituita da due colonne (Dare e Avere). Le registrazioni devono essere inserite all'interno di esse e la loro somma si deve annullare. Quindi, se tutto è compilato in maniera corretta, si dovrebbe avere un saldo pari a zero, ossia la registrazione "quadra". Gli strumenti necessari alla compilazione in partita doppia sono i seguenti:

- Libro Giornale: all'interno di esso vengono raccolte in ordine cronologico tutte le singole righe di registrazione in partita doppia e documenta gli avvenimenti salienti dal punto di vista contabile, amministrativo e fiscale.
- Piano dei Conti: rappresenta l'elenco di tutti i conti impiegati dall'azienda per le registrazioni in partita doppia ed è contenuto all'interno del Piano dei Conti, il quale si suddivide in tre sezioni:
  - Stato Patrimoniale: indica lo stato del patrimonio aziendale e si organizza in Attivo e Passivo.

STATO PATRIMONIALE	
ELEMENTI PATRIMONIALI ATTIVI (immobili, crediti, cassa)	PATRIMONIO NETTO
	ELEMENTI PATRIMONIALI PASSIVI (debiti)

FIGURE 3.6: Schema di stato patrimoniale.

- Conto Economico: contiene i conti relativi alle attività economiche dell'azienda ripartiti a loro volta in Costi e Ricavi.

CONTO ECONOMICO	
COSTI	RICAVI

FIGURE 3.7: Schema di conto economico.

- Nota Integrativa: contiene specifiche informazioni relative ai criteri di valutazione adottati, al contenuto delle voci di Stato Patrimoniale e Conto

Economico, e, infine, ad altri elementi caratterizzanti l'impresa (dipendenti, tipologie di azioni, e così via).

Come detto precedentemente per poter registrare una transazione economica si utilizza la ripartizione in due colonne: "DARE", in cui vengono inserite sia le variazioni positive dei conti economici sia le variazioni finanziarie attive; "AVERE", in cui vengono inserite le variazioni economiche negative dei conti economici e le variazioni finanziarie passive. Le regole di registrazione per il metodo della partita doppia sono le seguenti:

- Vanno eseguite almeno due annotazioni contemporaneamente.
- Le annotazioni vanno effettuate in due o più conti a sezioni divise o accostate.
- Le annotazioni vanno effettuate in sezioni opposte.
- Il totale dei valori registrati in Dare deve essere uguale al totale dei valori in Avere.

Il risultato è che i valori di ogni colonna finiscono per sommarsi tra loro. Quindi si avrà un prospetto contabile che consente di rilevare operazioni di segno opposto [21].

CONTO	
Denominazione del conto	
Sezione DARE (sinistra)	Sezione AVERE (destra)

FIGURE 3.8: Schema di conto.

### 3.3.3 Le dimensioni finanziarie

Per poter riassumere, registrare e organizzare le transazioni di un'azienda vengono utilizzati i conti di contabilità generale (General Ledger Accounts), o conti GL.

Il piano dei conti è strutturato su più livelli e normalmente si utilizza una codifica parlante per rappresentarli.

Ogni forma di attività, passività, patrimonio netto, reddito o costo ha un conto assegnato e questi conti sono tenuti all'interno della contabilità generale (general ledger)[22].

Le organizzazioni per poter gestire le varie transazioni hanno bisogno di diversi conti di contabilità generale. Questi sono classificati come conti di stato patrimoniale o conto economico. I conti nel bilancio comprendono tutto ciò che è indicato nel bilancio, comprese le attività, le passività e il patrimonio netto. I saldi a conto economico sono indicati come conti di conto economico. Questi conti contengono proventi e costi da operazioni e ricavi, spese, profitti e perdite da attività non operative[23].

Per poter ottenere informazioni utili dalla transazione che è stata registrata vengono utilizzate delle dimensioni aziendali strategiche dette dimensioni finanziarie. Queste vengono scelte a seconda delle esigenze e degli obiettivi dell'azienda. Esistono diverse dimensioni finanziarie nei servizi di dynamics 365 disponibili nel sistema e queste si dividono in:

1. dimensioni personalizzate, i cui valori vengono scelti e inseriti direttamente dall'utente;
2. dimensioni supportate da entità, i cui valori vengono inseriti in entità presenti all'interno del sistema.

Tra le dimensioni già presenti a sistema si può trovare:

Business unit	Cost center	Department
Retail channel	Value stream	Vendor
Vendor group	Customer	Customer group
Legal entity	Bank account	Bank group
Sales agreement	Purchase agreement	Item

Le dimensioni finanziarie vengono utilizzate di solito per rappresentare delle persone giuridiche.

#### Dimension attribute

Un *dimension attribute* [24] o più semplicemente *dimension* è un'informazione addizionale che l'utente può aggiungere alle combinazioni di conti del libro mastro. Il dimension attribute viene creato a partire da classi di entità già presenti nel sistema come ad esempio, department o centro di costo.

A ciascuna dimension vengono associati diversi valori che possono derivare o da un'entità già esistente nel sistema oppure da una creata dall'utente stesso. Ad ogni dimensione inoltre, corrisponde un riferimento ad una tabella del sistema.

Ad esempio, per l'entità *Customer* viene utilizzato il riferimento alla tabella *CustTable*. Come riferimento alle entità definite dall'utente, viene invece utilizzata la tabella *DimensionFinancialTag*. Infine, la tabella *DimensionAttribute* contiene i metadati relativi a ciascuna dimensione.

Per comprendere appieno quanto spiegato sopra verrà presentato un esempio di seguito: vi sono due dimensioni, la dimensione *Customer* e la dimensione *LicensePlate*. Per quanto riguarda la prima, va ad indicare i clienti già presenti all'interno dell'applicazione, mentre la seconda riguarda la nuova lista creata appositamente dall'utente.

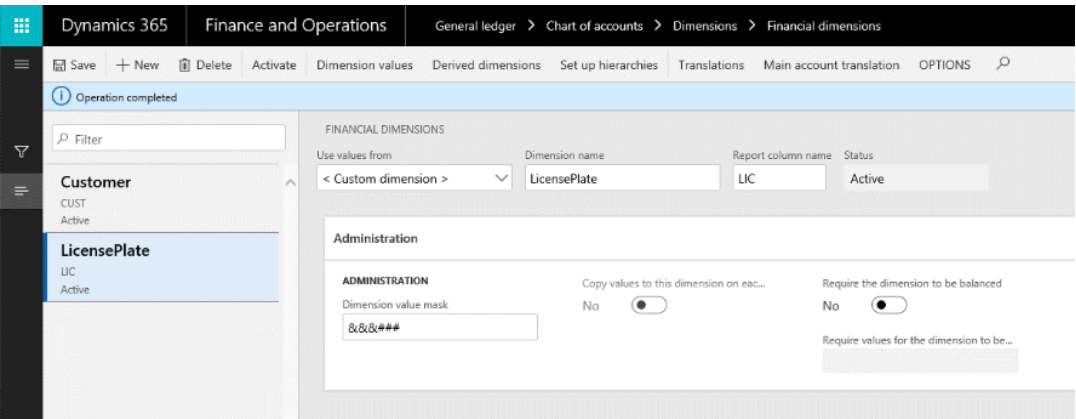


FIGURE 3.9: Dimension Attribute

Nella figura seguente si può vedere come vengono memorizzati i dati delle dimensioni all'interno della tabella *DimensionAttribute*.

	NAME	REPORTCOLUMNNAME	TYPE	VIEWNAME	RECID
1	Customer	CUST	0	DimAttributeCustTable	22565420939
2	LicensePlate	LIC	1	DimensionFinancialTag	22565420940

FIGURE 3.10: Query SQL che indica le informazioni di base relative a ciascuna dimensione

Per sapere se la dimensione deriva da un'entità già presente nel sistema oppure da un elenco realizzato appositamente dall'utente bisogna vedere il valore dell'attributo *Type* in figura. Dato che una singola entità può avere più scopi è possibile associare ad essa più dimensioni, una per ciascuno scopo.

Dimensions attribute values

Quando si parla dei valori di un dimension attribute si intende un'istanza della dimensione che fa parte del dimension framework. Tramite la proprietà *ViewName* della riga della tabella *DimensionAttribute* è possibile determinare i valori della singola dimensione. Inoltre, per poter visualizzare i valori delle dimensioni finanziarie è necessario selezionare il pulsante *Dimension values* dalla pagina *Financial Dimension*. Nella figura 3.11 viene riportato un esempio delle dimension values relative alla dimensione finanziaria *BusinessUnit*[23]:

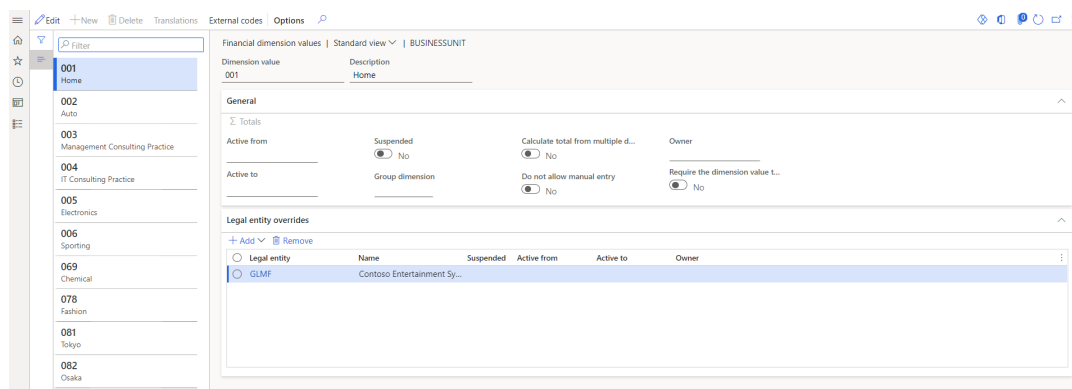


FIGURE 3.11: Elenco di valori fornito da BusinessUnit

I valori delle dimensioni finanziarie vengono archiviati nella tabella `DimensionAttributeValue`.

### Default Dimensions

All'interno della pagina Financial Dimension si possono trovare più di 30 dimensioni standard e le dimensioni vengono filtrate basandosi sulle strutture di conto effettivamente attive associate al libro mastro della società specificata. Le strutture dei conti possono essere gestite all'interno della pagina *Ledger*[25].

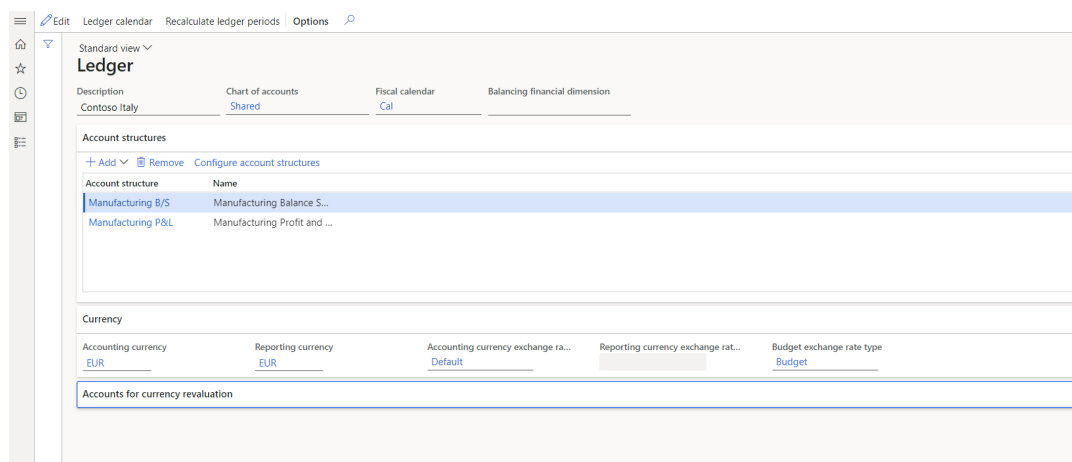


FIGURE 3.12: Pagina Ledger contenente le account structure relative alla company ITCO.

Le strutture dei conti possono avere dimensioni diverse e si possono visualizzare accedendo alla pagina *Ledger > Configure account structure*.

Nell'esempio riportato di seguito si può vedere una struttura dei conti con 3 dimensioni ossia MainAccount, BusinessUnit e Department.



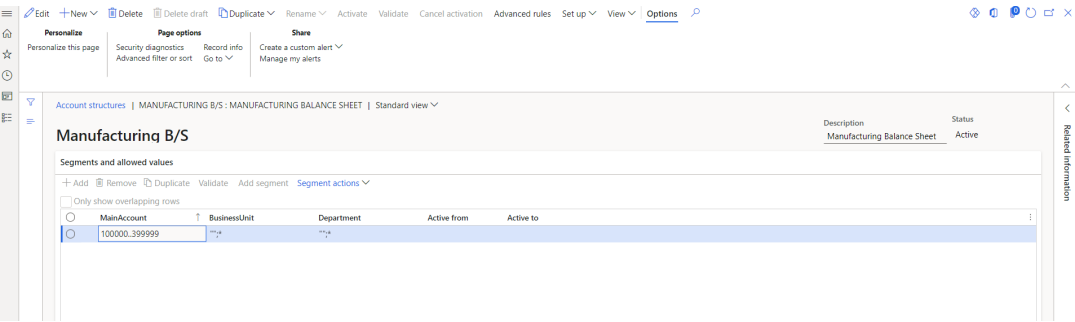


FIGURE 3.13: Esempio di un account structure.

Per poter memorizzare i valori che vengono associati alle dimensioni c'è bisogno di una tabella separata dalla tabella primaria che fa riferimento a tali valori di dimensione. Ad esempio, la tabella `LedgerJournalTable` ha una colonna `DimensionDefault` che contiene un riferimento di chiave esterna a un record nella tabella `DimensionAttributeValueSet`. Questo record è il record padre che rappresenta l'insieme di valori che viene mostrato.

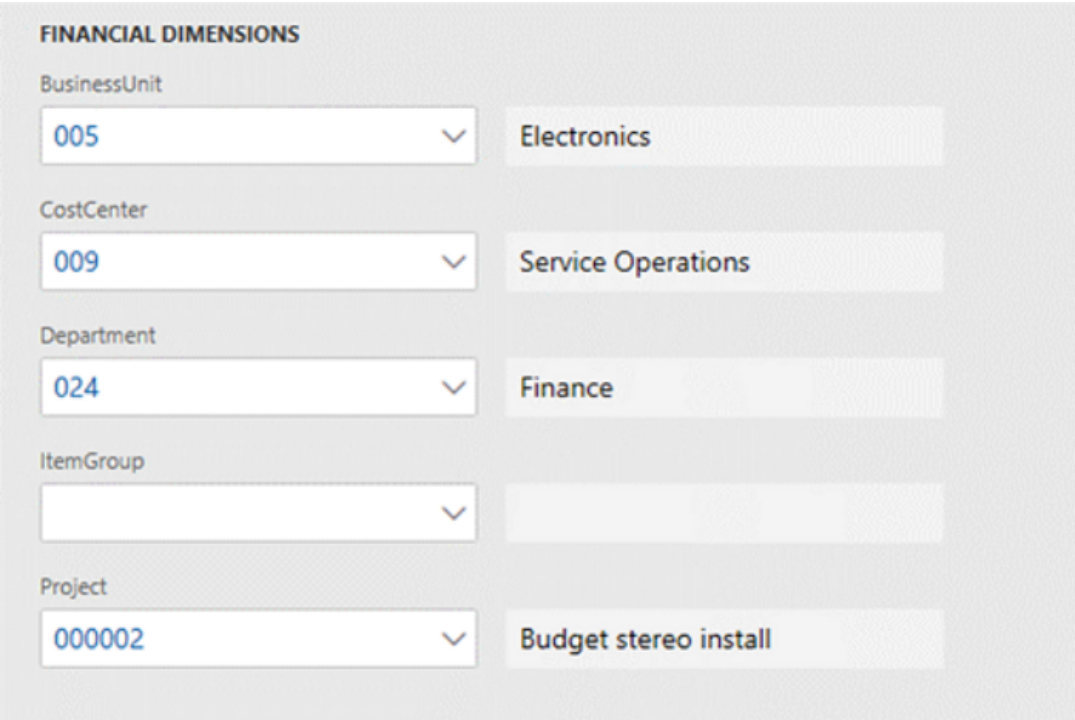


FIGURE 3.14: FinancialDimensions.

Ogni valore viene archiviato come riga separata nella tabella DimensionAttribute-ValueSetItem e dispone della stessa chiave esterna del record padre.

	DIMENSIONATTRIBUTEVALUE	DISPLAYVALUE	SETITEMRECID	DIMENSIONATTRIBUTE	ENTITYINSTANCE	ATTRIBUTEVALUERECD
1	22565423191	024	68719495445	22565421196	22565423101	22565423191
2	22565424661	005	68719495443	22565421195	22565423082	22565424661
3	22565425154	009	68719495444	22565421197	22565423086	22565425154
4	52565491825	000002	68719495446	22565421190	5637145326	52565491825

FIGURE 3.15: Tabella DimensionAttributeValueSetItem.

Per creare le ledger account dimensions le dimensioni predefinite vengono solitamente combinate tra loro. Queste combinazioni vengono poi utilizzate all'interno dei giornali di registrazione contabili. Quando l'utente inserisce per la prima volta un conto avrà il seguente campo:

FIGURE 3.16: Casella Account.

Nel momento in cui l'utente clicca sulla freccia a discesa visualizzerà i possibili segmenti.

FIGURE 3.17: Combinazione dimensioni finanziarie.

Dopo aver immesso i segmenti, che in questo caso sono il MainAccount e la dimensione BusinessUnit, il framework delle dimensioni salva la combinazione e quindi la convalida, in base ai vincoli[25] [24] [26].

### Main Account

Un Main Account[27] è un conto presente all'interno della contabilità generale. Viene utilizzato per registrare transazioni finanziarie, saldi o totali relativi ad attività, passività, ricavi, spese e patrimonio netto. Nella figura 3.18 viene visualizzata la pagina principale dei main accounts:

The screenshot displays the 'Main Account' configuration interface. On the left, a list of accounts is shown, including '110110. Bank Account - USD'. The main area is divided into several sections: 'General' (with tabs for Totals, Assign template, Additional consolidation accounts, and Currency code validation), 'IDENTIFICATION' (Main account, Name, Main account category), 'RELATED ACCOUNTS' (Offset account, Opening account), 'ADMINISTRATION' (Do not allow manual entry, Active from), 'CONSOLIDATION' (Default consolidation account), and 'CURRENCY' (Default Currency, Validate currency code, Foreign currency revaluation, Exchange rate type). The 'Main account category' is set to 'CASH', and the 'Main account type' is set to 'Asset'.

FIGURE 3.18: Main Account.

Quando viene creato un main account è necessario specificarne la tipologia:

The screenshot shows a dropdown menu for selecting the 'Main account type'. The menu is open, displaying a list of options: 'Asset', 'Profit and loss', 'Revenue', 'Expense', 'Balance sheet', 'Liability', 'Equity', 'Total', 'Reporting', and 'Common'. The 'Asset' option is currently selected.

FIGURE 3.19: Main Account Type.

All'interno del piano dei conti si possono trovare diversi tipi di *main account*.

- Profitti e perdite, ricavi e spese: sono utilizzati come conti di registrazione. I risultati di fine anno derivano dalla somma di tutti i conti profitti e perdite, ricavi e spese;
- Bilancio, attività, passività e patrimonio netto: sono dei conti di transazione utilizzati appositamente per registrare quello che possiede o che deve possedere una persona giuridica;
- Totale: sono dei conti utilizzati per aggiungere intervalli di account;
- Reporting: sono dei conti utilizzati per il reporting del rendiconto finanziario.

### 3.3.4 Business Events

Quando si esegue un processo aziendale c'è la possibilità di generare i cosiddetti business events[28]. Tramite questi meccanismi i sistemi esterni sono in grado di ricevere notifiche da parte di applicazioni finanziarie e operative e, una volta ricevuta la notifica, rispondono ai business events tramite delle business actions. Queste ultime si dividono in azioni del flusso di lavoro o azioni non del flusso di lavoro ma, in entrambi i casi, possono entrambe generare dei business events che verranno poi sfruttati come strumento di integrazione o notifica da parte dei sistemi esterni.

#### Catalogo eventi aziendali

Per visualizzare i business events disponibili nell'istanza utilizzata è necessario cliccare su: System administration > Set up > Business events, in questo modo si accede al catalogo. Dal catalogo è possibile filtrare i business events per categoria, ID evento aziendale e nome.

Per identificare l'origine di un evento aziendale si fa riferimento alla categoria, ad esempio, nella categoria Workflow si possono trovare tutti i business events che hanno origine dal sistema del flusso di lavoro. Le categorie prendono i nomi direttamente dai moduli di origine dei business events. Per comprendere al meglio il contesto in cui opera il business event e il business event stesso, nel catalogo è presente una descrizione.

Inoltre, all'interno del catalogo è possibile visualizzare l'elenco dei campi dati che verranno poi inviati nell'evento.

Nel caso in cui, un sistema di integrazione esterno richieda per un determinato evento lo schema del payload, è possibile scaricare il file JSON (Figura 3.20).

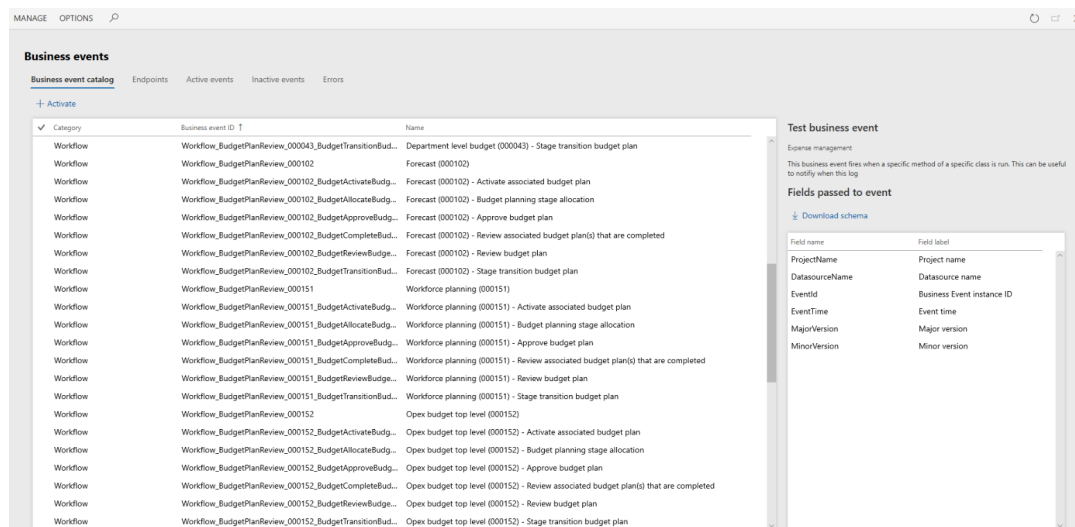


FIGURE 3.20: BusinessEventCatalog.

#### Attivazione dei business events

L'attivazione dei business events [28] avviene cliccando il pulsante "Attiva" direttamente dal catalogo. E' possibile specificare per quale persona giuridica attivare il business event oppure attivarlo per tutte le persone giuridiche lasciando il campo "Persona giuridica" vuoto.

Solo se il business event è attivo è possibile assegnargli degli endpoint.

Ovviamente, durante l'esecuzione dei processi aziendali, nel momento in cui si verifica un business event, il sistema andrà ad eseguire le elaborazioni solo per i business events che sono stati attivati in precedenza.

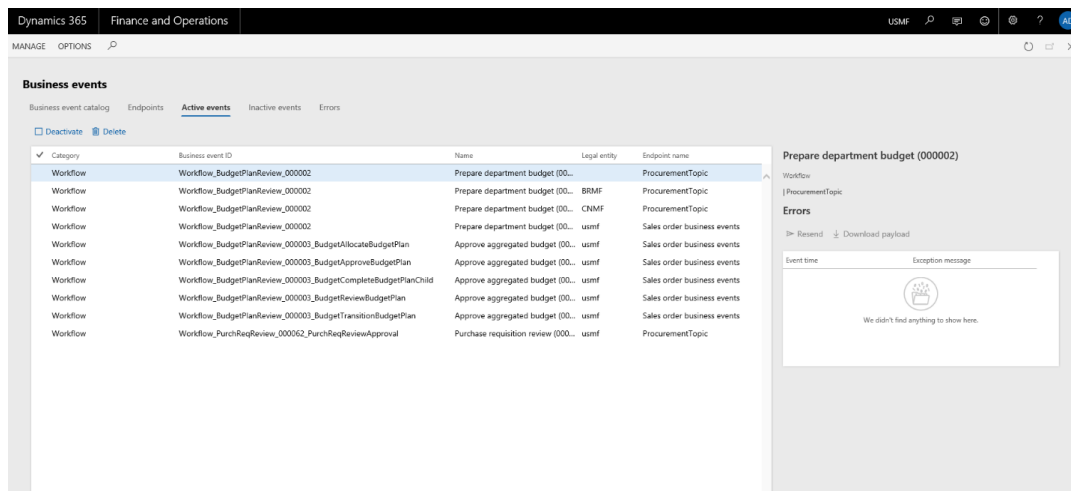


FIGURE 3.21: Business events activation.

Se necessario è possibile disattivare i business events così da non far avvenire più la loro elaborazione.

## Errori

Nel caso in cui si verificano degli errori[28] un business event potrebbe non essere consegnato all'endpoint, il sistema proverà a ripetere la consegna diverse volte, nel caso in cui l'errore permanga il business event verrà salvato in un registro degli errori.

## Endpoint

Gli eventi aziendali vengono inviati a diverse destinazioni, per poterle gestire vengono utilizzati gli *endpoint*[29]. Ne esistono diversi tipi tra cui:

- Azure Service Bus Queue
- Azure Service Bus Topic
- Azure Event Grid
- Azure Event Hub
- Azure Blob Storage
- HTTPS
- Microsoft Power Automate
- Dataverse

In particolare, prendendo in esame gli endpoint di Microsoft Power Automate essi vengono utilizzati per le sottoscrizioni che sono create e inviate direttamente da un flusso in Power Automate. Inoltre, l'endpoint viene generato all'interno della pagina Business Event nella scheda Endpoint.

**Workflow business events**

All'interno dei flussi di lavoro è possibile creare diversi business events in vari punti del flusso in esame.

In Power Automate, tramite endpoint, è possibile configurare e usare un business event utilizzando il trigger *Quando si verifica un evento aziendale*, presente nel connettore finanziario e operativo.

## Chapter 4

# Metodologia

### 4.1 Overview del progetto

Il progetto in esame nasce dall'esigenza di creare un sistema di integrazione che ha come obiettivo finale il trasferimento di un file di paghe e note spese da OneDrive a Dynamics 365 for Finance and Operations. Una volta avvenuta l'importazione, è stata generata una scrittura contabile in grado di registrare le transazioni contenute nel file in esame.

Per realizzare la migrazione del file è stata utilizzata la piattaforma di Azure Integration Services, *Azure Logic Apps*. Infatti, avendo come mezzo l'app per la logica di Microsoft, è stato possibile creare la connessione necessaria a far comunicare OneDrive e Dynamics 365.

La progettazione del sistema di integrazione è stata divisa nei seguenti passi:

1. analisi dei file inviati dall'azienda cliente;
2. selezione dei dati ritenuti di primaria importanza per il processo e conseguente mappatura;
3. creazione della prima logic app necessaria all'importazione del file di paghe/note spese;
4. implementazione del codice all'interno di Visual Studio per l'elaborazione del file e la successiva creazione del giornale contabile;
5. progettazione della seconda logic app per la ricezione dei dati processati e per la notifica della corretta creazione del giornale.

Per poter comprendere al meglio tutti i passi del processo è stato realizzato un BPMN (Business Process Modeling Notation), così da avere una rappresentazione visiva di più facile interpretazione. Il BPMN (Figura 4.1) è composto da una pool rappresentante il processo e tre lane che mostrano gli ambienti di sviluppo: OneDrive, Azure Logic Apps e Dynamics 365 for Finance and Operations.

1. Il processo inizia tramite un trigger; perciò, quando viene creato un file su OneDrive il token comincia a muoversi.  
La prima cosa che si osserva è l'esecuzione del flusso di lavoro all'interno dell'applicazione per la logica; tramite opportune azioni e connettori è resa possibile l'importazione del file all'interno di OneDrive.

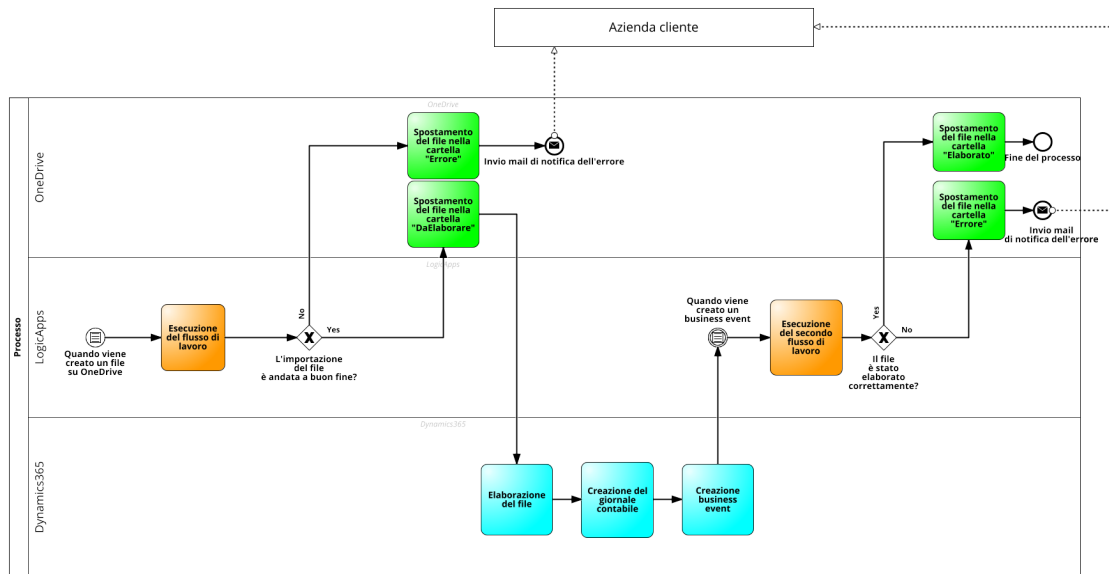


FIGURE 4.1: Schema del processo.

2. Dopo aver verificato se effettivamente l'importazione è avvenuta correttamente, si sposta il file dalla cartella di partenza "Files" alla cartella "DaElaborare"; altrimenti, se c'è stato qualche errore, il file viene spostato nella cartella "Errore" e l'utente viene avvisato tramite una mail.
3. Una volta che il file è stato importato all'interno di Dynamics 365, si procede con la sua elaborazione e avviene la creazione del giornale contabile adibito a contenere le informazioni del file di paghe/note spese.  
In seguito alla generazione della scrittura contabile, viene creato un business event che, svolgendo il ruolo di endpoint, diventerà il trigger per la seconda logic app.
4. L'evento che dà la possibilità al processo di proseguire è, appunto, la creazione del business event che permette al secondo flusso di lavoro di avere inizio. Tramite le azioni che fanno parte del flusso è possibile ottenere i dati elaborati su Dynamics 365.
5. Dopo aver verificato che la creazione del giornale contabile sia andata a buon fine si sposta il file nella cartella "Elaborato"; altrimenti, se si è verificato un errore, il file viene spostato nella cartella "Errore" e viene avvisato l'utente tramite una mail.



## Chapter 5

# Sperimentazione

### 5.1 Analisi dei file inviati dall'azienda cliente

Il primo passo è stato analizzare i file d'esempio inviati dallo studio paghe delle aziende clienti in modo da selezionare le principali informazioni per l'implementazione del sistema di integrazione. Dopodiché sono stati mappati i dati tramite apposite tabelle.

1. Il primo file messo a disposizione dall'azienda è stato un file a lunghezza fissa al cui interno si possono trovare informazioni riguardanti:

- anno,
- mese,
- azienda,
- sede,
- gruppo,
- sottogruppo,
- analitico,
- descrizione conto,
- dare,
- avere.

Questo rappresenta il file di paghe e note spese in seguito importato in Dynamics 365.



PNC_2021_09_001173_05092021_161303 - Blocco note di Windows		
File Modifica Formato Visualizza ?		
20210911730000 00	0010010001Salari	00000003592,5200000000000,000
20210911730000 00	0010010002Stipendi	00000001850,3700000000000,000
20210911730000 00	0010010031Ferie	00000000075,9400000000000,000
20210911730000 00	0010010033R.o.l.	00000000002,0900000000000,000
20210911730000 00	0010060001Compensi Stagisti/Tirocinanti	000000000500,0000000000000,000
20210911730000 00	0010130010Malattia c/inps	00000000034,4500000000000,000
20210911730000 00	0010150001Trattamento integr.reddito DL.3/2020	000000000279,4500000000000,000

FIGURE 5.1: File paghe/note spese da importare.

2. Dopodiché, è stato esaminato un secondo file contenente le indicazioni sull'anagrafica dei conti. All'interno di esso è stato possibile leggere i valori di gruppo, sottogruppo e analitico e la relativa descrizione del conto da essi identificato.

Gruppo	Sottogruppo	Analitico	Descrizione
1 - RETRIBUZIONI	1 - Retribuzioni	1	Salari
1 - RETRIBUZIONI	1 - Retribuzioni	2	Stipendi
1 - RETRIBUZIONI	1 - Retribuzioni	31	Ferie
1 - RETRIBUZIONI	1 - Retribuzioni	32	Permessi
1 - RETRIBUZIONI	4 - Compensi ass.partecipazione	3	Trattamento Fine Mandato

FIGURE 5.2: Anagrafica dei conti.

3. Infine, è stato sottoposto ad analisi un file excel contenente la descrizione dei raggruppamenti contabili.

ANNO	MESE	AZIENDA	SEDE	GRUPPO	SOTTO_GRUPPO	ANALITICO	DESCRIZIONE CONTO	DARE	AVERE
2021	9	1173	0	1	1	1	1 Salari	3592,52	0
2021	9	1173	0	1	1	1	2 Stipendi	1850,37	0
2021	9	1173	0	1	1	1	31 Ferie	75,94	0
2021	9	1173	0	1	1	1	33 R.o.I.	2,09	0
2021	9	1173	0	1	1	6	1 Compensi Stagisti/Tirocinanti	500	0
2021	9	1173	0	1	1	13	10 Malattia c/inps	34,45	0
2021	9	1173	0	1	1	15	1 Trattamento integr.reddito DL.3/2020	279,45	0
2021	9	1173	0	1	1	19	1 INPS -Contributi c/dipendente	0	523,85
2021	9	1173	0	1	1	23	1 Irpef 1001 - Lav.dipendente	0	848,52
2021	9	1173	0	1	1	25	16 Arrotondamento attuale	2,55	0
2021	9	1173	0	1	1	28	1 NETTO IN BUSTA	0	4965
2021	9	1173	0	11	1	1	1 Contributi c/ditta su retribuzioni	1677,52	0
2021	9	1173	0	11	1	1	2 Contributi c/ditta su ferie	23,98	0
2021	9	1173	0	11	1	1	4 Contributi c/ditta su R.o.I.	0,65	0
2021	9	1173	0	11	1	1	15 Contributi c/ditta Add.ASpl	25,9	0
2021	9	1173	0	11	1	1	39 Restituzione contr.Add. ASpl	0	111

FIGURE 5.3: File Excel dati azienda cliente.

## 5.2 Creazione delle tabelle

Per la mappatura dei dati contenuti all'interno dei file sono state create delle apposite tabelle e delle form per l'inserimento delle informazioni da parte dell'utente.

Sono state create sei tabelle:

- LedgerImpPayrollDecodeAccount\_ALTC,
- LedgerImpPayrollDecodeAnalytic\_ALTC,
- LedgerImpPayrollDecodeCompany\_ALTC,
- LedgerImpPayrollDecodeGroup\_ALTC,
- LedgerImpPayrollDecodeSubGroup\_ALTC,
- LedgerImpPayrollStatusTable\_ALTC.

Di seguito verranno analizzate le principali.

### *LedgerImpPayrollDecodeCompany\_ALTC*

La tabella *LedgerImpPayrollDecodeCompany\_ALTC* contiene i dati che sono stati inseriti direttamente dall'utente all'interno della form relativa. Questa tabella verrà sfruttata in seguito all'interno della Logic App per ricavare il Company Id. I campi della tabella sono:

- *TrackType*: tipo di tracciato (paghe/note spese),
- *ExCompanyId*: codice dell'azienda, contenuto nel nome del file,
- *CompanyId*: codice della società.

Table browser: LedgerImpPayrollDecodeCompany\_ALTC

Standard view ▾

Filter

✓	CompanyId	ExCompan... ↑	TrackType	recVersion	Partition	RecId
	ITCO ▾	1173	Payroll ▾	1	5637144576	5637146076

FIGURE 5.4: LedgerImpPayrollDecodeCompany\_ALTC.

### *LedgerImpPayrollDecodeAnalytic\_ALTC*

La tabella rappresenta l'anagrafica del dettaglio analitico contenuta nel file. I campi sono i seguenti:

- *ExtGroup*: codice gruppo esterno,
- *ExtSubGroup*: codice sottogruppo esterno,
- *ExtAnalytic*: codice analitico,
- *Name*: nome del conto.

Table browser: LedgerImpPayrollDecodeAnalytic\_ALTC

My view ▼

Filter

✓	ExtAnalytic	ExtGroup ↑	ExtSubGroup	Name	dataAreald	recVersion	Partition	RecId
	1	1	1	Salari	itco	1	5637144576	5637144577
	0	11	15	Credito	itco	816792134	5637144576	5637144576

FIGURE 5.5: LedgerImpPayrollDecodeAnalytic\_ALTC.

### *LedgerImpPayrollDecodeAccount\_ALTC*

La tabella rappresenta la decodifica dei conti. I campi della tabella sono:

- *LedgerImpPayrollDecodeAnalytic*: tabella contenente gruppo, sottogruppo e analitico del conto,
- *LedgerDimension*: ledger dimension del conto.

Company Account

Standard view ▼

Filter

○	↺	Reference	Main account
		11 15 0	130400
✓		1 1 1	602110 <span>▼</span>

FIGURE 5.6: LedgerImpPayrollDecodeAccountSimpleList\_ALTC.

Il Main Account è stato inserito manualmente, selezionando nel menu a tendina una dimensione finanziaria corrispondente alla voce "credito" e una corrispondente alla voce "salari".

***LedgerImpPayrollStatusTable\_ALTC***

La tabella è necessaria a memorizzare i dati del file importato.  
Verrà spiegato all'interno nel paragrafo 5.3 come sarà valorizzata.  
I campi della tabella sono:

- *Filename*: nome del file,
- *Register*: registro all'interno del quale vengono salvati gli errori che si verificano durante l'elaborazione del giornale,
- *FileStatus*: stato di elaborazione del file,
- *TrackType*: tipo di tracciato (paghe/note spese),
- *IncomingFile*: campo di tipo memo in cui viene memorizzato il contenuto del file,
- *ProcessingStatus*: stato di elaborazione del giornale,
- *CompanyId*: codice della società.

Table browser: LedgerImpPayrollStatusTable\_ALTC

Standard view

CompanyId	Filename	FileStatus	IncomingFile	ProcessingStatus	TrackType	dataAreaid	recVersion	Partition	RecId
ITCJ	PNC_2021_09_001173_1 - Copia.txt	Processed	20210911730000 00	Elaborato	Payroll	ITCO	1423945804	5637144576	5637276576

FIGURE 5.7: LedgerImpPayrollStatusTable\_ALTC.

Sulla tabella LedgerImpPayrollStatusTable\_ALTC è stata implementata una DataEntity (LedgerImpPayrollStatusTable\_ALTCEntity) dove sono stati mappati i campi descritti in precedenza.

### 5.3 Creazione della prima logic app per l'importazione del file

Il primo obiettivo da realizzare è stato il trasferimento del file di paghe e note spese da una directory presente su OneDrive a Dynamics 365 FO.

Per fare ciò è stato creato un flusso di lavoro tramite Logic Apps, in cui sono state inserite apposite azioni in grado di richiamare delle entità dati realizzate all'interno di Dynamics 365 FO; proprio queste data entities hanno permesso di creare il collegamento con l'app per la logica.

Di seguito si entrerà nel dettaglio della progettazione.

Per la creazione della prima Logic App, adibita all'importazione del file di paghe, è stato realizzato un flusso di lavoro costituito da dieci passi. Di seguito verranno descritti uno ad uno per comprenderne a fondo il funzionamento.

- **Passo 1:** *'Quando viene creato un file su One Drive'*

Prima di tutto è stato inserito il trigger di 'One Drive for Business' che dà inizio al flusso di lavoro.

Nel campo 'Cartella' bisogna indicare il percorso per arrivare alla directory in cui vengono creati i file, nel caso in esame il percorso è: 'Documenti/Files'.

Quando viene creato un file su OneDrive

\* Cartella  
/Documenti/Files

Includi sottocartelle  
No

Deduci tipo di contenuto  
Si

Specificare la frequenza in base a cui verificare gli elementi  
5 Mese

Add new parameter

FIGURE 5.8: Trigger OneDrive for Business.

Perciò, dopo aver messo in esecuzione la logic app, è possibile creare il file all'interno della directory; questo non è altro che il trigger che permette al flusso di iniziare la sua esecuzione.

Nome	Stato
PNC_2021_09_001173_1	✓
InElaborazione	✓
Errore	✓
Elaborato	✓
DaElaborare	✓

FIGURE 5.9: Directory Files contenente il file di paghe e note spese

- **Passo 2 e 3:** "Inizializza variabile FileName" e "Inizializza variabile FileContent"

E' stata poi selezionata l'action DataOperation chiamata "Inizializza variabile" che permette di inizializzare due variabili di tipo stringa necessarie per gli step successivi del flusso: il nome del file (FileName) e il contenuto del file (FileContent). All'interno del campo **Valore** è stato possibile inserire come input il nome del file e il contenuto del file, i quali rappresentano gli output del passaggio precedente.

Figure 5.10 shows the configuration for the 'Inizializza variabile FileName' action. The 'Nome' field is set to 'FileName', the 'Tipo' is 'Stringa', and the 'Valore' field contains a dynamic content connector 'Nome file'.

FIGURE 5.10: Action per inizializzazione della variabile FileName.

Figure 5.11 shows the configuration for the 'Inizializza variabile FileContent' action. The 'Nome' field is set to 'FileContent', the 'Tipo' is 'Stringa', and the 'Valore' field contains a dynamic content connector 'Contenuto file'.

FIGURE 5.11: Action per inizializzazione della variabile FileContent.

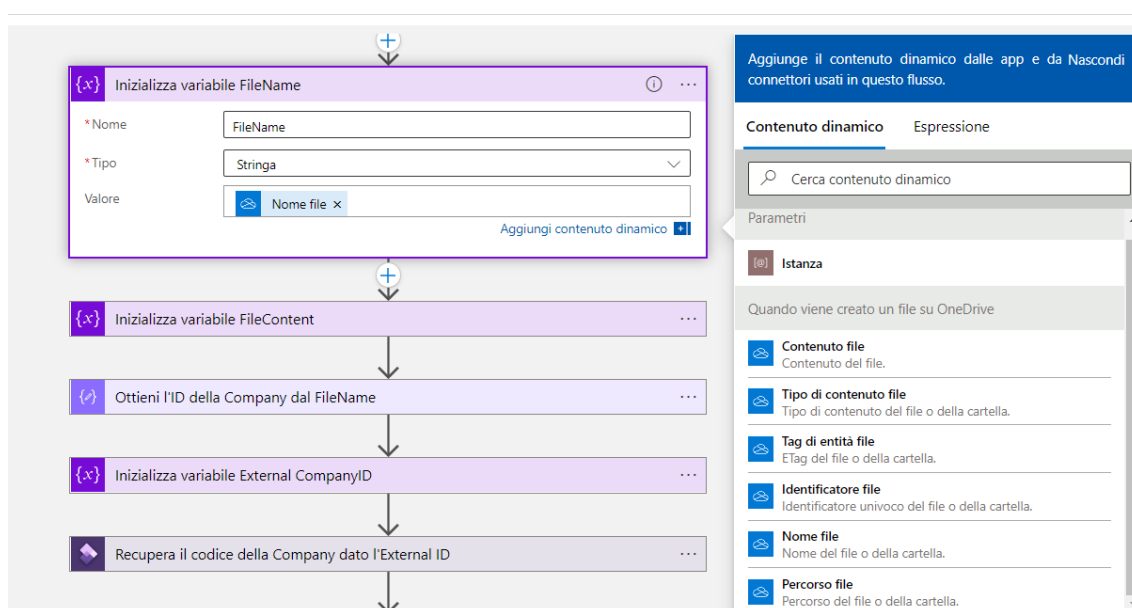


FIGURE 5.12: Lista valori da immettere.

- **Passo 4:** "Ottieni l'ID della Company dal FileName"

Tramite l'esecuzione di un'operazione sui dati quale la 'substring', si va a selezionare dal nome del file, l'ID della Company.

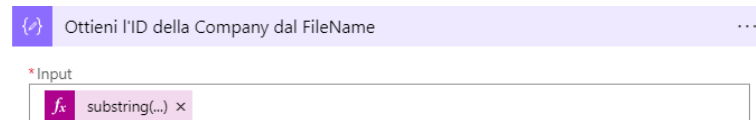


FIGURE 5.13: Operazione di substring per estrazione del nome.

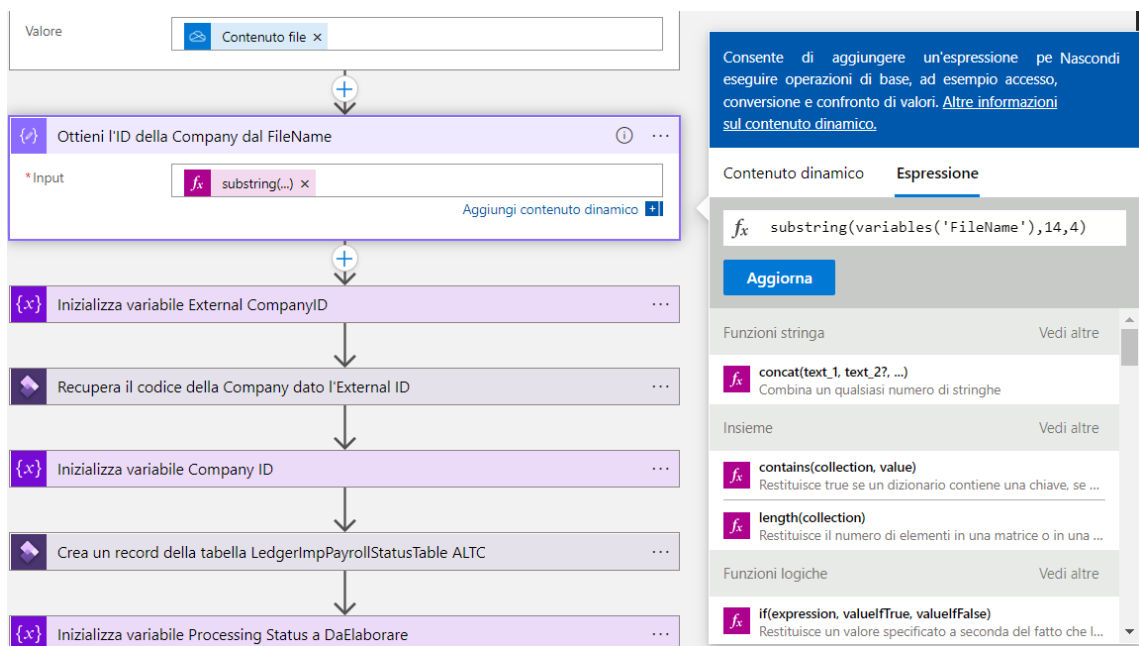


FIGURE 5.14: Substring applicata sul nome del file.



- **Passo 5:** "Inizializza variabile External Company ID"

Viene inizializzata una variabile di tipo stringa chiamata 'ExCompanyID' in cui viene salvato il valore della chiave esterna della società. Nel caso in esame questo id corrisponde al valore: 1173.

The screenshot shows a configuration window for the 'Inizializza variabile External CompanyID' action. It includes fields for 'Nome' (ExCompanyID), 'Tipo' (Stringa), and 'Valore' (Output).

FIGURE 5.15: Action per inizializzazione della variabile External CompanyId.

- **Passo 6:** "Recupera il codice della Company dato l'external ID"

A questo punto si ha la necessità di recuperare l'Id della Company dalla tabella "LedgerImpPayrollDecodeCompany\_ALTC", che è stata precedentemente compilata dall'utente.

La tabella in questione contiene:

- *TrackType*: il tipo di tracciato (paghe/note spese),
- *ExCompanyId*: il codice dell'azienda, contenuto nel nome del file,
- *CompanyId*: il codice della società.

Table browser: LedgerImpPayrollDecodeCompany\_ALTC

Standard view

Filter

✓	CompanyId	ExCompan... ↑	TrackType	recVersion	Partition	RecId
	ITCO	1173	Payroll	1	5637144576	5637146076

FIGURE 5.16: LedgerImpPayrollDecodeCompany\_ALTC.

Per poter estrapolare il *CompanyId* è stata utilizzata la seguente action, tramite la quale è possibile recuperare la riga di una tabella specificando l'ID di un oggetto appartenente alla stessa.

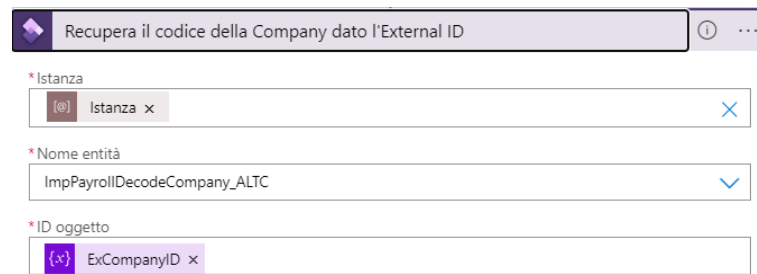
L'istanza si riferisce all'URL dell'istanza di destinazione dell'applicazione a cui deve connettersi il connettore; quindi il valore corrisponde all'URL senza il prefisso 'https://' o ne viene scelto uno dal menu a discesa. Il menu in questione elenca tutti gli ambienti distribuiti nel tenant di Azure Active Directory per l'account utente, usato per accedere al client specifico come Power Automate, Power Apps o App per la logica.

Il nome dell'entità si riferisce all'entità di dati da cui è necessario recuperare il record. Il menu a discesa mostra l'elenco delle entità di dati dall'ambiente di destinazione.

L'ID oggetto si riferisce ai campi delle chiavi primarie che devono essere specificati per identificare in modo univoco il record che deve essere recuperato. I valori devono essere specificati come un elenco di valori separati da virgole nell'ordine definito nell'entità.

Quindi, per prima cosa è stato creato il collegamento con Dynamics specificando nel campo **Nome entità** il nome della data entity creata all'interno di Dynamics: *ImpPayrollDecodeCompany\_ALTC* riferita alla tabella *LedgerImpPayrollDecodeCompany\_ALTC*.

Tramite questo collegamento è stato possibile recuperare il codice della company ITCO specificando nel campo **ID oggetto** l'external ID 1173.



Recupera il codice della Company dato l'External ID

\* Istanza  
[e] Istanza x

\* Nome entità  
ImpPayrollDecodeCompany\_ALTC

\* ID oggetto  
{\*} ExCompanyID x

FIGURE 5.17: Action per il recupero del codice della Company.

- **Passo 7:** "Inizializza variabile Company ID"

E' stata poi creata la variabile di tipo stringa 'Company' che contiene il Company ID ottenuto dal passaggio precedente, ossia: ITCO.

The screenshot shows a configuration box for the 'Inizializza variabile Company ID' action. It includes a 'Nome' field with the value 'Company', a 'Tipo' dropdown menu set to 'Stringa', and a 'Valore' field containing 'ID account società' with a small 'x' icon to its right.

FIGURE 5.18: Action per inizializzazione della variabile Company ID.

- **Passo 8:** "Crea un record della tabella LedgerImpPayrollStatusTable ALTC"

Questo è il passaggio fondamentale per l'importazione del file. Tramite questa action è stato possibile inserire una nuova riga all'interno della tabella *LedgerImpPayrollStatusTable\_ALTC*.

I campi della tabella sono:

- *Filename*: nome del file,
- *Register*: registro all'interno del quale vengono salvati gli errori che si verificano durante l'elaborazione del giornale,
- *FileStatus*: stato di elaborazione del file,
- *TrackType*: tipo di tracciato (paghe/note spese),
- *IncomingFile*: campo di tipo memo in cui viene memorizzato il contenuto del file,
- *ProcessingStatus*: stato di elaborazione del giornale,
- *CompanyId*: codice della società.

Table browser: LedgerImpPayrollStatusTable\_ALTC

Standard view

Filter

CompanyId	Filename	FileStatus	IncomingFile	ProcessingStatus	TrackType	dataAreaid	recVersion	Partition	RecId
ITCO	PNC_2021_09_001173_1 - Copia.txt	Processed	20210911730000 00	Elaborato	Payroll	ITCO	1423945804	5637144576	5637276576

FIGURE 5.19: LedgerImpPayrollStatusTable\_ALTC.

Come per il passo 6, per creare il collegamento con Dynamics è stato specificato nel campo nome entità, l'entità dati creata appositamente su Dynamics chiamata *ImpPayrollFiles*, relativa alla tabella *LedgerImpPayrollStatusTable\_ALTC*.

Una volta richiamata questa dataEntity è stato possibile aggiungere all'azione, come parametri, i campi della tabella "LedgerImpPayrollStatusTable\_ALTC". Perciò, ogni campo della tabella "LedgerImpPayrollStatusTable\_ALTC" viene riempito con le variabili valorizzate nei passi precedenti, quali: "FileName", "Company", "FileContent". Viene inoltre settato lo stato di elaborazione del file, "ProcessingStatus", come "DaElaborare".

Come si può vedere dalla figura 5.19, in questo modo è stata valorizzata la tabella.

Da sottolineare il fatto che il file di paghe e note spese è stato memorizzato all'interno del campo *IncomingFile* e in questo modo è avvenuta l'importazione all'interno di Dynamics 365 for Finance and Operations.

Crea un record della tabella LedgerImpPayrollStatusTable ALTC

\* Istanza  
[e] Istanza x

\* Nome entità  
ImpPayrollFiles

\* File Name  
{x} FileName x

Società  
{x} Company x

IncomingFile  
{x} FileContent x

ProcessingStatus  
DaElaborare

Company/Id  
{x} Company x

Add new parameter

FIGURE 5.20: Action per la creazione di un record della tabella LedgerImpPayrollStatusTable\_ALTC.

- **Passo 9:** "Inizializza variabile Processing Status a Da Elaborare"

Viene inizializzata una nuova variabile di tipo stringa "ProcessingStatus" e valorizzata come "DaElaborare", necessaria poi nel passaggio seguente.

The screenshot shows a configuration window for the 'Inizializza variabile Processing Status a DaElaborare' action. It has three fields: 'Nome' (Name) with the value 'ProcessingStatus', 'Tipo' (Type) set to 'Stringa' (String), and 'Valore' (Value) set to 'DaElaborare'.

FIGURE 5.21: Action per inizializzazione della variabile Processing Status.

- **Passo 10:** *Condizione*

E' poi necessario, tramite questa action, verificare se il ProcessingStatus (ossia lo stato di elaborazione del giornale) della tabella "LedgerImpPayrollStatusTable\_ALTC" corrisponde al ProcessingStatus, che è stato inizializzato al passaggio precedente.

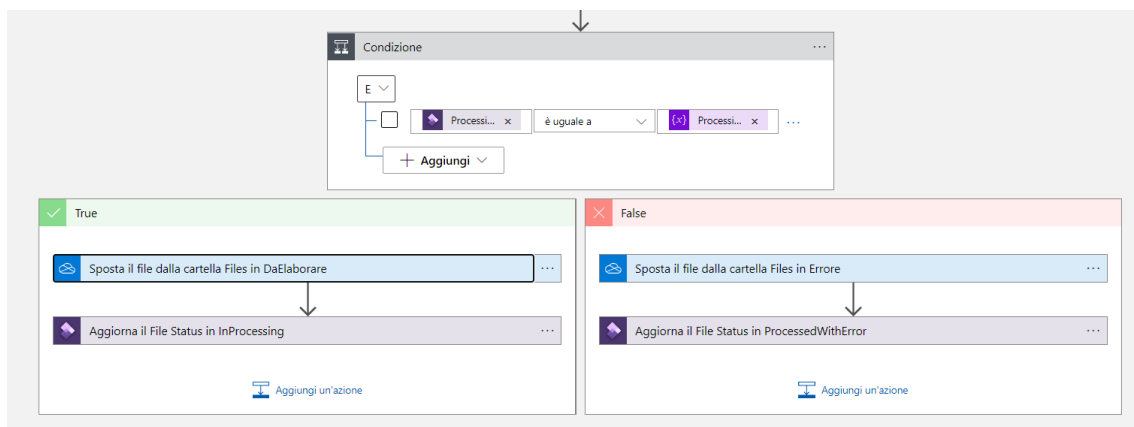


FIGURE 5.22: Action per verifica della condizione di uguaglianza.

Se la risposta è affermativa, il file viene spostato nella directory "InElaborazione" e il campo FileStatus (ossia, lo stato di processing del file) della tabella in esame viene settato a "InProcessing".

True

Sposta il file dalla cartella Files in DaElaborare

\* Percorso file: Percorso file x

\* Percorso file di destinazione: /Documenti/Files/InElaborazione/ Nome file x

Sovrascrivi: No

Aggiorna il File Status in InProcessing

\* Istanza: Istanza x

\* Nome entità: ImpPayrollFiles

\* ID oggetto: {x} Company x {x} FileName x

\* File Name: {x} FileName x

Società: {x} Company x

File Status: InProcessing

Add new parameter

FIGURE 5.23: Flusso in caso di risposta affermativa.

Se la risposta è negativa, il file viene spostato nella directory "Errore" e il campo FileStatus della tabella in esame viene settato a "ProcessedWithError".

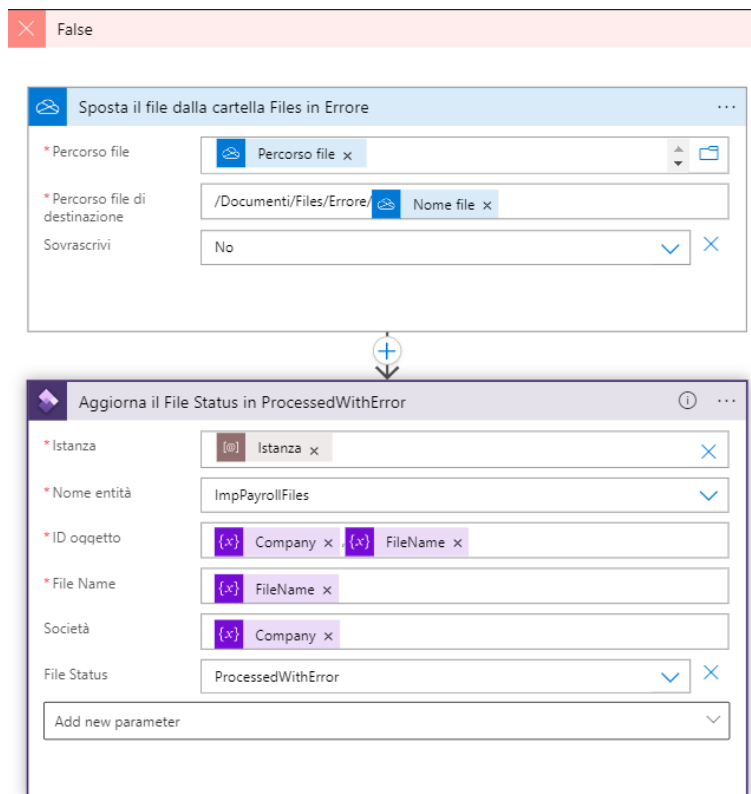


FIGURE 5.24: Flusso in caso di risposta negativa.

Infine, viene inviata una mail di notifica all'azienda cliente notificando l'errore.

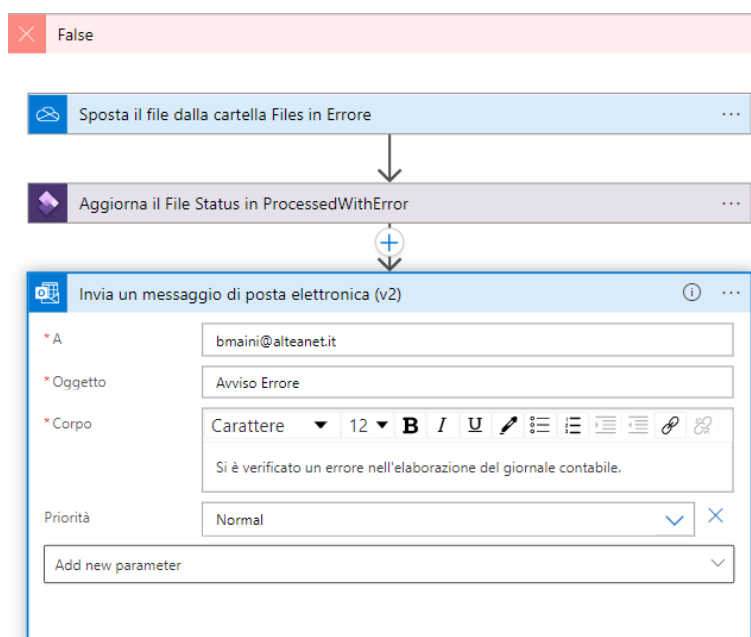


FIGURE 5.25: Flusso in caso di risposta negativa pt.2.

## 5.4 Implementazione

Tramite la logic app, descritta nel paragrafo precedente, è stato possibile importare il file di paghe memorizzandolo all'interno del campo "incomingFile" della tabella *LedgerImpPayrollStatusTable\_ALTC*.

In questo capitolo verrà illustrato e spiegato il codice che è stato implementato all'interno di Visual Studio per poter portare a termine i seguenti obiettivi:

- creare il giornale contabile contenente le informazioni del file di paghe e note spese;
- creare il business event che permette di trasferire i dati elaborati alla seconda Logic App che verrà descritta nel Capitolo 6.

### 5.4.1 Creazione SysOperationFramework

Per la realizzazione del giornale contabile si è deciso di utilizzare il *SysOperationFramework* [30] che permette di eseguire un processo in background. Nel caso in esame il processo in questione consiste nel processare il set di record contenuti all'interno della tabella *LedgerImpPayrollStatusTable\_ALTC*.

1. Per poter inviare i dati al processo bisogna realizzare una classe **Contract** che deve avere come attributo [DataContractAttribute]. Il nome della classe deve contenere, per convenzione, la parola 'Contract' al termine. Inoltre, al suo interno è necessario specificare le variabili e un metodo *parm* per ciascuna variabile.

```
[DataContractAttribute, SysOperationAlwaysInitializeAttribute]
class LedgerImpPayrollContract_ALTC extends SysOperationDataContractBase
    implements SysOperationInitializable
{
    private ProcessingStatus processingStatus;
    [DataMemberAttribute]
    public ProcessingStatus parmProcessingStatus(ProcessingStatus
        _processingStatus
        = processingStatus)
    {
        processingStatus = _processingStatus;
        return processingStatus;
    }

    public void initialize()
    {
        processingStatus = processingStatus::DaElaborare;
    }
}
```

2. Per poter configurare il *SysOperationFramework* è necessario creare una classe '**Controller**' che vada ad estendere la classe *SysOperationServiceController*.



Al suo interno viene creato un metodo 'main' che viene richiamato dalla voce di menu. All'interno del metodo 'main' viene chiamato "startOperation" che serve a richiamare il metodo della classe Service così da poter eseguire l'elaborazione.

```
class LedgerImpPayrollController_ALTC extends SysOperationServiceController
{
    public static void main(Args _args)
    {
        LedgerImpPayrollController_ALTC controller =
        new LedgerImpPayrollController_ALTC();
        controller.parmClassName(classstr(LedgerImpPayrollService_ALTC));
        controller.parmMethodName(methodstr(LedgerImpPayrollService_ALTC,
        createJournal));
        controller.parmExecutionMode(SysOperationExecutionMode::Synchronous);

        controller.startOperation();
    }
}
```

3. Il codice effettivamente necessario per l'elaborazione è contenuto all'interno della classe 'Service'; bisogna quindi creare una nuova classe che estenda la classe *SysOperationServiceBase*.

Anche in questo caso il nome della classe, per convenzione, deve terminare con la parola 'Service'.

```
class LedgerImpPayrollService_ALTC extends SysOperationServiceBase
{
}
```

All'interno di questa classe è stato creato il metodo pubblico che consente la creazione del giornale contabile e che accetta un unico parametro di tipo 'Contract'.

Dopo aver creato le classi Contract, Service e Controller bisogna creare un *action menu* che andrà a chiamare la classe Controller.

#### **All'interno della classe *LedgerImpPayrollService\_ALTC***

Nella classe Service sono stati implementati i seguenti metodi contenenti il codice necessario all'elaborazione del file:

- createJournal(),
- import(),
- processFile(),
- processRecord(),
- createDefaultDimensions().

Di seguito verranno analizzati uno ad uno.

### Metodo createJournal

Per registrare le transazioni contenute all'interno del file di paghe che è stato importato, è stato creato un giornale contabile su Dynamics 365 for Finance and Operations.

All'interno del giornale contabile è necessario inserire:

- data: anno, mese e data della transazione
- voucher
- company (ad esempio, ITCO)
- account type
- account
- descrizione: viene utilizzata per inserire i nomi dei conti coinvolti nella transazione
- debito
- credito
- currency

Di seguito viene presentato il codice scritto per la sua creazione.

Il primo metodo in esame è il *createJournal* a cui viene passato come parametro di input un'istanza della classe *Contract*.

```
public void createJournal(LedgerImpPayrollContract_ALTC _dataContract)
{
    LedgerImpPayrollStatusTable_ALTC    ledgerImpPayrollStatusTable;
    str                                   fileContent;
    LedgerImpPayrollBusinessEvent        ledgerImpPayrollBusinessEvent;
```

Tramite l'istruzione *while select* è possibile scorrere riga per riga la tabella *LedgerImpPayrollStatusTable* contenente i dati del file importato.

Il ciclo While continua ad iterare fin quando il campo *ProcessingStatus* della tabella *LedgerImpPayrollStatusTable* è uguale a 'DaElaborare'.

```
while select forupdate ledgerImpPayrollStatusTable
where ledgerImpPayrollStatusTable.ProcessingStatus
== _dataContract.parmProcessingStatus()
{
```

Se la condizione di uguaglianza è verificata allora si entra all'interno del try, dove troviamo un *ttsLevelChecking*. Questo controllo assicura che nessun record possa essere modificato o cancellato se non è stato selezionato per l'aggiornamento all'interno dello stesso ambito di transazione. Per garantire l'integrità si utilizzano le seguenti istruzioni:

- `ttsBegin`: segna l'inizio di una transazione. Questo assicura l'integrità dei dati e garantisce che tutti gli aggiornamenti eseguiti fino al termine della transazione siano coerenti,
- `ttsCommit`: segna la fine di una transazione.

Le istruzioni tra il `ttsBegin` e il `ttsCommit` permettono di:

- inizializzare la stringa `fileContent` in cui andremo a memorizzare il contenuto del file di paghe. Il file viene importato tramite il metodo `import` (che verrà in seguito analizzato nel dettaglio),
- settare ad "Elaborato" il campo `ProcessingStatus`, che serve a tenere traccia dello stato di processing del giornale contabile,
- aggiornare la tabella `ledgerImpPayrollStatusTable` in seguito alle modifiche apportate.

```
try
{
    ttsbegin;
    fileContent = ledgerImpPayrollStatusTable.IncomingFile;
    this.import(fileContent);
    ledgerImpPayrollStatusTable.ProcessingStatus =
    ProcessingStatus::Elaborato;
    ledgerImpPayrollStatusTable.update();
    ttscommit;
}
```

All'interno del `catch` invece viene richiamato il metodo `saveError` necessario a valorizzare il campo "Register".

```
catch
{
    //viene richiamato il metodo saveError che serve per
    //valorizzare il registro ('Register')
    this.saveError(ledgerImpPayrollStatusTable);
}

    ledgerImpPayrollBusinessEvent = ledgerImpPayrollBusinessEvent::
    newFromLedgerImpPayrollStatusRecId(ledgerImpPayrollStatusTable.RecId);
    ledgerImpPayrollBusinessEvent.send();
}
}
```

### Metodo import

Il metodo *import* ha come parametro di input la stringa *fileContent* che, come è stato specificato nel punto precedente, contiene il file di paghe che è stato importato tramite il flusso di lavoro delle Logic Apps.

Dopo aver trasformato il set di caratteri del file in una sequenza di byte e aver salvato questo flusso in memoria, viene richiamato il metodo *processFile*.

```
public boolean import(str _fileContent)
{
    boolean ret = true;

    System.Byte[] byteArray;
    System.IO.MemoryStream memoryStream;

    // codifica un set di caratteri in una sequenza di byte
    byteArray = System.Text.Encoding::UTF8.GetBytes(_fileContent);

    // la classe MemoryStream crea un flusso
    // il cui archivio di backup è la memoria
    memoryStream = new System.IO.MemoryStream(byteArray);

    // richiama la funzione per il processamento dello stream
    ret = this.processFile(memoryStream);

    memoryStream.Close();

    return ret;
}
```

### Metodo processFile

1. Il metodo *processFile*, come suggerisce il nome, è stato realizzato per processare il file di paghe, a cui viene passato il flusso di byte, *memoryStream*, creato nel metodo *import*.

```
protected boolean processFile(System.IO.Stream _memoryStream)
{
    LedgerJournalTable      ledgerJournalTable;
    LedgerJournalEngine      ledgerJournalEngine;
    boolean                  ret = true;
    TextStreamIo             textStreamIo;
    container                currentBuffer;
    str                      currentBufferStr;
```

2. Tramite il metodo *constructForRead* viene creata un'istanza della classe *TextStreamIo* che fornisce le funzionalità per leggere il flusso di testo. Il metodo *read* legge la riga completa successiva da un oggetto di tipo *TextIo* e ritorna un contenitore che contiene un singolo record. Questo contenitore viene salvato all'interno di *currentBuffer*. Ogni voce del contenitore equivale a un campo del record. La definizione del prossimo record completo è verificata dalle proprietà *inFieldDelimiter* e *inRecordDelimiter*.

```
textStreamIo = TextStreamIo::constructForRead(_memoryStream);
textStreamIo.inRecordDelimiter('\r\n');
textStreamIo.inFieldDelimiter('\r\n');

currentBuffer = textStreamIo.read();
```

3. Si è passati poi alla creazione della testata del giornale contabile[31]. La testata del giornale, come si può vedere anche dalla figura 5.26, deve contenere:

- numero di serie del giornale,
- nome del giornale,
- anno di pubblicazione,
- descrizione.

○	↺	Journal batch number	↑	Name	Year	Voucher to be reco...	Description
○		ITCO-000555		GL DAILY	2022	▼	General Ledger Daily

FIGURE 5.26: Esempio di testata del giornale contabile.

Quindi, dopo aver controllato che il giornale non sia già presente, vengono inserite le diverse variabili che compongono la testata.

Si può trovare inoltre un'istanza della classe *LedgerJournalEngine*, che permette di settare il tipo di *Journal*. La classe *LedgerJournalEngine* supporta le form del ledger journal con una varietà di funzioni.

```
//creazione della testata del giornale

if (!ledgerJournalTable.RecId)
{
    ledgerJournalTable.clear();

    // inserimento del nome del giornale
    ledgerJournalTable.initFromLedgerJournalName("GL DAILY");

    // inserimento del numero del giornale
    ledgerJournalTable.JournalNum =
    JournalTableData::newTable(ledgerJournalTable).nextJournalId();
    ledgerJournalTable.insert();

    // La classe LedgerJournalEngine supporta le form
    // del ledger journal con una varietà di caratteristiche
    ledgerJournalEngine = LedgerJournalEngine::construct(
    ledgerJournalTable.JournalType);

    // '.newJournalActive' un nuovo giornale è l'active record
    ledgerJournalEngine.newJournalActive(ledgerJournalTable);
}
```

4. Tramite il metodo *conPeek* viene recuperato un elemento specifico dal contenitore *currentBuffer* e viene indicata la posizione dell'elemento da recuperare. In questo caso viene preso il primo elemento ossia la prima riga del file.

```
while(!textStreamIo.status())
{
    currentBufferStr = conPeek(currentBuffer, 1);
```

5. Come è stato spiegato in precedenza, per applicare il metodo della partita doppia è necessario inserire i valori di debito e credito all'interno delle transazioni. L'ammontare del debito e del credito è stato estrapolato dal file posizionale messo a disposizione dall'azienda cliente, tramite la funzione *substring*.

```
AmountCur amountCurDebit = str2Num(strReplace
(subStr(currentBufferStr, 145, 15),',','.'));
AmountCur amountCurCredit = str2Num(strReplace
(subStr(currentBufferStr, 160, 15),',','.'));
```

6. Se sono valorizzati contemporaneamente i campi di debito e credito Dynamics 365 FO non creerà la riga della transazione. Per ovviare a questo problema si è deciso di procedere nel modo seguente: si è verificato, come prima cosa, se credito e debito fossero entrambi valorizzati.

Nel caso di risposta positiva, è stata richiamata la funzione `ProcessRecord` due volte, così da creare due righe separate del giornale, una per il debito e una per il credito. Al contrario, nel caso in cui non fossero valorizzati entrambi i campi, è stata creata una singola transazione in cui viene registrato o il valore del debito o il valore del credito.

Per facilitare la comprensione, è stato aggiunto il flag `'CreditDebit'` che permette di indicare se sono valorizzati entrambi i campi (`CreditDebit = Tutti`) o uno solo dei due (`CreditDebit = Debit` o `CreditDebit = Credit`).

```

        if (amountCurDebit && amountCurCredit)
        {
            ret = this.processRecord(currentBufferStr, ledgerJournalEngine,
            CreditDebit::Debit) && ret;
            ret = this.processRecord(currentBufferStr, ledgerJournalEngine,
            CreditDebit::Credit) && ret;
        }
        else
        {
            ret = this.processRecord(currentBufferStr,
            ledgerJournalEngine,
            CreditDebit::Tutti) && ret;
        }
        currentBuffer = textStreamIo.read();
    }
    return ret;
}

```

### Metodo `processRecord`

Questo metodo permette di processare il file di paghe riga per riga e i suoi parametri sono: `_currentBufferStr` ossia una riga del file, un'istanza del `ledgerJournalEngine` e il flag `creditDebit`.

```

protected boolean processRecord(str _currentBufferStr,
LedgerJournalEngine ledgerJournalEngine, CreditDebit _creditDebit)
{
    boolean                                ret = true;
    LedgerImpPayrollDecodeAccount_ALTC    ledgerAccount;
    LedgerJournalTrans ledgerJournalTrans;
    AmountCur amountCurDebit;
    AmountCur amountCurCredit;
}

```

Bisogna andare a valorizzare i campi gruppo, sottogruppo e analitico che andranno a definire il conto. Questi sono stati estrapolati dal file posizionale tramite il metodo `substring`.

```

ExtGroup extgroup = str2Int(subStr(_currentBufferStr, 46, 3));
ExtSubGroup extSubGroup = str2Int(subStr(_currentBufferStr, 49, 3));
ExtAnalytic extAnalytic = str2Int(subStr(_currentBufferStr, 52, 4));

```

Per l'inizializzazione delle variabili debito e credito si procede nel modo seguente:

Se il flag = Tutti, vengono inizializzati debito e credito.

```

if (_creditDebit == creditDebit::Tutti)
{
    amountCurDebit = str2Num(strReplace(subStr(_currentBufferStr, 145,
15),',','.'));
    amountCurCredit = str2Num(strReplace(subStr(_currentBufferStr,
160, 15),',','.'));
}

```

Altrimenti, se il flag = Debit, viene inizializzata la variabile debito mentre la variabile credito viene impostata pari a zero.

```

else
{
    if (_creditDebit == CreditDebit::Debit)
    {
        amountCurDebit = str2Num(strReplace(subStr
(_currentBufferStr, 145, 15),',','.'));
        amountCurCredit = 0;
    }
}

```

Oppure, se il flag = Credit, viene inizializzata la variabile credito mentre la variabile debito viene impostata pari a zero.

```

if(_creditDebit == CreditDebit::Credit)
{
    amountCurCredit = str2Num(strReplace(subStr
(_currentBufferStr, 160, 15),',','.'));
    amountCurDebit = 0;
}
}

```

Il metodo *initValue()* viene chiamato quando viene inserita una nuova riga nella tabella ledgerJournalTrans.

```

ledgerAccount = LedgerImpPayrollDecodeAccount_ALTC::getRecId(extgroup,
extSubGroup, extAnalytic);

ledgerJournalTrans.clear();

//inizializza nuova transazione
ledgerJournalTrans.initValue();
ledgerJournalEngine.initValue(ledgerJournalTrans);

```



Vengono poi impostate: la valuta, la data della transazione e il tipo di conto.

```
//valuta
ledgerJournalTrans.CurrencyCode = "EUR";

//data della transazione
ledgerJournalTrans.TransDate = DateTimeUtil::getSystemDate(
DateTimeUtil::getUserPreferredTimeZone());

//tipo di conto, in questo caso 'ledger'
ledgerJournalTrans.AccountType = LedgerJournalACType::Ledger;
```

Viene poi richiamato il metodo per la creazione della default dimension, a cui viene passato come parametro la value dimension pari a "001" che corrisponde al campo "Home" della financial dimension "Business Unit".

```
ledgerJournalTrans.DefaultDimension = this.createDefaultDimensions("001");
```

L' API *serviceCreateLedgerDimension()* è l'API più utilizzata per le dimensioni del libro mastro. Deve essere richiamata ogni volta che è necessario creare una nuova combinazione di conti contabili da un conto predefinito o da una combinazione di conti contabili esistente da zero a tre dimensioni predefinite. Se è necessario unire più di tre dimensioni predefinite, questa API può essere chiamata più volte. In questo caso, diverse fonti prendono il risultato della chiamata precedente come primo parametro per ogni chiamata successiva. La dimensione MainAccount viene recuperata solo dalla dimensione contabile fornita.

```
ledgerJournalTrans.LedgerDimension          =
LedgerDimensionFacade::serviceCreateLedgerDimension(
ledgerAccount.LedgerDimension, LedgerJournalTrans.DefaultDimension);

//impostazione del tipo di offset account in questo caso Ledger
ledgerJournalTrans.OffsetAccountType        = LedgerJournalACType::Ledger;

//inizializzazione ammontare del debito
ledgerJournalTrans.AmountCurDebit = amountCurDebit;

//inizializzazione ammontare del credito
ledgerJournalTrans.AmountCurCredit = amountCurCredit;
```

Se è tutto valido, vengono inseriti debito e credito all'interno della riga della transazione.

```
if (ledgerJournalTrans.validateWrite())
{
    ledgerJournalTrans.insert();
}
```

```

        ledgerJournalEngine.write(ledgerJournalTrans);
    }

    return ret;
}

```

### Metodo createDefaultDimension

E' stata creata la dimensione finanziaria, di cui si è parlato nel capitolo 3, passandogli la value dimension. Il dimension attribute, preso come esempio, è la "BUSINESS UNIT" il cui dimension attribute value è "001". Il metodo ritorna la default dimension.

```

protected DimensionDefault createDefaultDimensions(DimensionValue _value)
{
    DimensionAttribute          dimensionAttribute;
    DimensionAttributeValue     dimensionAttributeValue;
    DimensionDefault            defaultDimension;

```

Di seguito vengono spiegate alcune delle principali righe di codice.

- Il 'dimensionAttributeRecId' sarebbe da prelevare da un parametro ma non essendo disponibile su questo ambiente è stato inserito manualmente,
 

```
DimensionAttributeRecId dimensionAttributeRecId = 22565421195;
```
- La classe 'DimensionAttributeValueSetStorage' gestisce il salvataggio della tabella 'DimensionAttributeValueSet' e della tabella 'DimensionAttributeValueSetItem',
- La classe 'DimensionAttributeValueValidationRequest' viene utilizzata per catturare l'informazione che è richiesta per validare il dimension attribute values,
- La classe 'DimensionAttributeValueValidationStatusContract' è stata utilizzata come un data contract per la validazione del dimension attribute value.

```

DimensionAttributeValueSetStorage valueSetStorage =
new DimensionAttributeValueSetStorage();

```

```

DimensionAttributeValueValidationRequest
dimensionAttributeValueValidationRequest;

```

```

DimensionAttributeValueValidationStatusContract
dimensionAttributeValueValidationStatusContract;

```

Viene impostato il dimension attribute tramite il metodo find passando il dimensionAttributeRecId e il valore del dimensionAttribute.

```

    if (_value != '')
    {
        // viene impostato il dimension attribute cercandolo

```

```
// tramite il metodo find
dimensionAttribute = DimensionAttribute
::find(dimensionAttributeRecId);

// viene impostato il value del dimension attribute
// ottenendolo tramite il metodo find
dimensionAttributeValue = DimensionAttributeValue::
findByDimensionAttributeAndValue(
dimensionAttribute, _value);
```

La *DimensionAttributeValueValidationRequest* è utilizzata per catturare le informazioni necessarie per convalidare i dimension attribute values.

```
if (dimensionAttributeValue)
{
    dimensionAttributeValueValidationRequest =
    DimensionAttributeValueValidationRequest::
    newForDimensionAttributeValueRecId(
    dimensionAttributeValue.RecId,
    DateTimeUtil::getSystemDate(DateTimeUtil
    ::getUserPreferredTimeZone()));
```

La classe *DimensionAttributeValueValidation* gestisce la validazione e la memorizzazione dei valori degli attributi delle dimensioni.

```
dimensionAttributeValueValidationStatusContract =
DimensionAttributeValueValidation::getStatus(
dimensionAttributeValueValidationRequest);
```

Il metodo *addItem* aggiunge l'oggetto "DimensionAttributeValue" specificato all'insieme.

Se il valore "DimensionAttribute" è associato al valore già presente nell'insieme, l'indice verrà aggiornato per puntare al nuovo valore e verrà restituito lo stesso indice.

Se il valore "DimensionAttribute" non è ancora presente nell'insieme, verrà aggiunto alla fine e verrà restituito l'indice del nuovo valore.

L'ordinamento e l'indice dell'insieme sono stabili per un'istanza in esecuzione della classe "DimensionAttributeValueSetStorage", ma possono cambiare se l'istanza viene mantenuta e ricaricata.

Dopodichè, tramite il metodo *save()* viene salvato il value set, se non esiste ancora, e ritorna l'id del record del *DimensionAttributeValueSet* associato.

```
// metodo parm per lo stato di validazione della dimensione
if (dimensionAttributeValueValidationStatusContract.parmValidationStatus() ==
DimensionValidationStatus::Valid)
{
    valueSetStorage.addItem(dimensionAttributeValue);
    defaultDimension = valueSetStorage.save();
}
```

```

else
{
    dimensionAttributeValueValidationStatusContract.displayValidationMessages();
}

}
}

return defaultDimension;
}

```

Nelle figure 5.27 e 5.28 viene riportato il risultato finale della creazione della testata e del giornale contabile.

Standard view ▾

### General journals

Show

Not posted ▾ ☐ Show user-created only

List General Setup Customs declaration Blocking Financial dimensions History

Journal batch number	Name	Year	Voucher to be reco...	Description	Posted	Posted on
ITCO-000555	GL DAILY	2022	▾	General Ledger Daily		
ITCO-000556	GL DAILY	2022		General Ledger Daily		
ITCO-000557	GL DAILY	2022		General Ledger Daily		
ITCO-000558	GL DAILY	2022		General Ledger Daily		
ITCO-000559	GL DAILY	2022		General Ledger Daily		
ITCO-000560	GL DAILY	2022		General Ledger Daily		
ITCO-000561	GL DAILY	2022		General Ledger Daily		
ITCO-000693	GL DAILY	2022		General Ledger Daily		
ITCO-000786	GL DAILY	2022		General Ledger Daily		
ITCO-000787	GL DAILY	2022		General Ledger Daily		
ITCO-000788	GL DAILY	2022		General Ledger Daily		
ITCO-000789	GL DAILY	2022		General Ledger Daily		
ITCO-000790	GL DAILY	2022		General Ledger Daily		

FIGURE 5.27: Esempio di General Journal.

ITCO-000790: GL DAILY | Standard view ▾

### Journal voucher

Display journal lines

All ▾

All General Invoice Payment Payment fee Fixed assets Remittance History

+ New Delete Voucher Financial dimensions Sales tax Functions ▾

Date	Voucher	Company	Account type	Account	Description	Debit	Credit	Offset com...	Offset account type	Offset account	Offset transaction
8/23/2022	GL000002482	itco	Ledger	602110-001---		3,592.52		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			1,850.37		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			75.94		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			2.09		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			500.00		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			34.45		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			279.45		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger				523.85	itco	Ledger		
8/23/2022	GL000002482	itco	Ledger				848.52	itco	Ledger		
8/23/2022	GL000002482	itco	Ledger			2.55		itco	Ledger		
8/23/2022	GL000002482	itco	Ledger				4,965.00	itco	Ledger		

Account name Salaries - Other Expe... Offset account name Calculated sales tax amount 0.00 Actual sales tax amount 0.00

CURRENCY DEBIT CREDIT BALANCE REPORTING CURRENCY DEBIT CREDIT BALANCE

VOUCHER 688,071.81 688,071.81 0.00 688,071.81 688,071.81 0.00 Budget check results

JOURNAL 688,071.81 688,071.81 0.00 688,071.81 688,071.81 0.00 Budget check not performed

FIGURE 5.28: Esempio di giornale.

### 5.4.2 Creazione business event

#### LedgerImpPayrollBusinessEvent

Una volta che il giornale è stato creato bisogna far sì che Dynamics 365 for Finance and Operations comunichi nuovamente con la Logic App così da potergli trasferire i dati che sono stati elaborati. Per far ciò è stato necessario creare un Business Event.

Per implementare il business event [32] sono stati seguiti tre passi fondamentali, ossia:

1. creare il contratto,
2. creare l'evento,
3. aggiungere il codice che serve ad inviare l'evento.

Inoltre, è necessario implementare due classi:

- Business event – Questa classe estende la classe BusinessEventsBase. Va a supportare la creazione del business event, la costruzione del payload, e l'invio del business event,
- Business event contract – Questa classe estende la classe BusinessEventsContract. Definisce il payload dell'evento aziendale e consente di compilare il contratto in fase di esecuzione.

Il processo di implementazione di un'estensione della classe BusinessEventsBase [32] consiste nell'estendere la classe BusinessEventsBase e nell'implementare un metodo costruttore statico, un metodo privato new, metodi per mantenere lo stato interno e il metodo buildContract.

1. Prima di tutto viene estesa la classe **BusinessEventBase**. Importante sottolineare la presenza dell'attributo BusinessEvents . Questo attributo fornisce al framework degli eventi aziendali informazioni sul contratto, il nome e la descrizione dell'evento aziendale. Fornisce inoltre il modulo di cui fa parte l'evento aziendale. Le etichette devono essere definite per gli argomenti del nome e della descrizione.

```
[BusinessEvents(classStr(LedgerImpPayrollBusinessEventContract),
'LedgerImportStatusPostedBusinessEvent', '', ModuleAxapta::Ledger)]
```

```
class LedgerImpPayrollBusinessEvent extends BusinessEventsBase
{
    private RecId ledgerImpPayrollStatusRecId;
    private LedgerImpPayrollStatusTable_ALTC ledgerImpPayrollStatusTable;
```

2. E' stato implementato il metodo private parm per mantenere lo stato interno.

```
private RecId parmLedgerImpPayrollStatusRecId(RecId
    _ledgerImpPayrollStatusRecId
    = ledgerImpPayrollStatusRecId)
{
    ledgerImpPayrollStatusRecId = _ledgerImpPayrollStatusRecId;
    return ledgerImpPayrollStatusRecId;
}
```

3. E' stato implementato il metodo statico `newFrom<my_buffer>`. La parte `<my_buffer>` del nome del metodo è in genere il buffer della tabella utilizzato per inizializzare il contratto dell'evento aziendale. Nel caso in questione il nome del metodo sarà: `newFromLedgerImpPayrollStatusRecId`.

```
public static LedgerImpPayrollBusinessEvent
newFromLedgerImpPayrollStatusRecId(
    RecId _ledgerImpPayrollStatusRecId)
{
    LedgerImpPayrollBusinessEvent businessEvent = new
    LedgerImpPayrollBusinessEvent();
    businessEvent.parmLedgerImpPayrollStatusRecId(
        _ledgerImpPayrollStatusRecId);

    return businessEvent;
}
```

4. E' stato poi implementato un nuovo metodo privato. Questo metodo viene chiamato solo dal metodo del costruttore statico.

```
private void new()
{
}
```

5. Poi si può trovare il metodo `buildContract` che viene chiamato solo quando un evento aziendale è abilitato per un'azienda.

```
[Wrappable(true), Replaceable(true)]
public BusinessEventsContract buildContract()
{
    select ledgerImpPayrollStatusTable where
    ledgerImpPayrollStatusTable.RecId ==
    this.parmLedgerImpPayrollStatusRecId();
    return LedgerImpPayrollBusinessEventContract
    ::newFromLedgerImpPayrollStatus(
    ledgerImpPayrollStatusTable);
}
}
```

### LedgerImpPayrollBusinessEventContract

Una classe di contratto per eventi aziendali estende la classe `BusinessEventsContract`. Definisce e popola il payload dell'evento aziendale. Sebbene gli eventi aziendali siano leggermente diversi, il nucleo del contratto di evento aziendale rimane invariato. L'estensione della classe `BusinessEventContract`, la definizione dello stato interno, l'implementazione di un metodo di inizializzazione, l'implementazione di un metodo costruttore statico e l'implementazione di metodi `parm` per accedere allo stato del contratto fanno parte del processo di costruzione di un `BusinessEventContract`.

1. Estensione della classe **BusinessEventContract**. La classe deve avere l'attributo **DataContract**.

```
[DataContract]
class LedgerImpPayrollBusinessEventContract extends BusinessEventsContract
{
```

2. Sono state aggiunte variabili private per mantenere lo stato del contratto.

```
private Filename fileName;
private FileStatus fileStatus;
private TrackType trackType;
private ProcessingStatus processingStatus;
private CompanyId companyId;
```

3. Implementazione di un metodo di inizializzazione protetto. Il metodo *initialize* è responsabile dell'impostazione dello stato privato della classe del contratto dell'evento aziendale, in base ai dati forniti tramite il metodo del costruttore statico. Dovrebbe essere protetto in modo che un'estensione di classe Chain of Command (CoC) possa essere utilizzata per estendere la classe di contratto.

```
private void initialize(LedgerImpPayrollStatusTable_ALTC
    _ledgerImpPayrollStatusTable)
{
    fileName = _ledgerImpPayrollStatusTable.FileName;
    fileStatus = _ledgerImpPayrollStatusTable.FileStatus;
    trackType = _ledgerImpPayrollStatusTable.TrackType;
    processingStatus = _ledgerImpPayrollStatusTable.ProcessingStatus;
    companyId = _ledgerImpPayrollStatusTable.CompanyId;
}
```

4. Implementazione di un metodo di costruzione statico. Il metodo del costruttore statico chiama il metodo *initialize* per inizializzare lo stato della classe privata.

```
public static LedgerImpPayrollBusinessEventContract
newFromLedgerImpPayrollStatus(LedgerImpPayrollStatusTable_ALTC
    _ledgerImpPayrollStatusTable)
{
    var contract = new LedgerImpPayrollBusinessEventContract();
    contract.initialize(_ledgerImpPayrollStatusTable);

    return contract;
}

private void new()
{
}
```

5. Implementazione dei metodi **parm** per accedere allo stato del contratto. I metodi **parm** devono avere gli attributi **DataMember('<name>')** e **BusinessEventsDataMember('<description>')**. Il nome fornito nell'attributo **DataMember**

(in questo caso, 'FileStatus') sarà visibile ai consumatori del contratto dati. La descrizione fornita nell'attributo BusinessEventsDataMember sarà visibile nell'interfaccia utente (UI) del catalogo Business Events e utilizzata per descrivere i membri dati di questo contratto.

```
[DataMember('FileName'), BusinessEventsDataMember('FileName')]
public Filename parmFileName(Filename _fileName = fileName)
{
    fileName = _fileName;

    return fileName;
}

[DataMember('FileStatus'), BusinessEventsDataMember('FileStatus')]
public FileStatus parmFileStatus(FileStatus _fileStatus = fileStatus)
{
    fileStatus = _fileStatus;

    return fileStatus;
}

[DataMember('TrackType'), BusinessEventsDataMember('TrackType')]
public TrackType parmTrackType(TrackType _trackType = trackType)
{
    trackType = _trackType;

    return trackType;
}

[DataMember('ProcessingStatus'),
BusinessEventsDataMember('ProcessingStatus')]
public ProcessingStatus parmProcessingStatus(ProcessingStatus
_processingStatus = processingStatus)
{
    processingStatus = _processingStatus;

    return processingStatus;
}

[DataMember('CompanyId'), BusinessEventsDataMember('CompanyId')]
public CompanyId parmCompanyId(CompanyId _companyId = companyId)
{
    companyId = _companyId;

    return companyId;
}
}
```

6. Infine è necessario inviare il business event, il codice in questione è stato implementato all'interno della classe **LedgerImpPayrollService\_ALTC**, nel metodo *createJournal*.



```
ledgerImpPayrollBusinessEvent =
ledgerImpPayrollBusinessEvent::newFromLedgerImpPayrollStatusRecId(
ledgerImpPayrollStatusTable.RecId);
ledgerImpPayrollBusinessEvent.send();
```

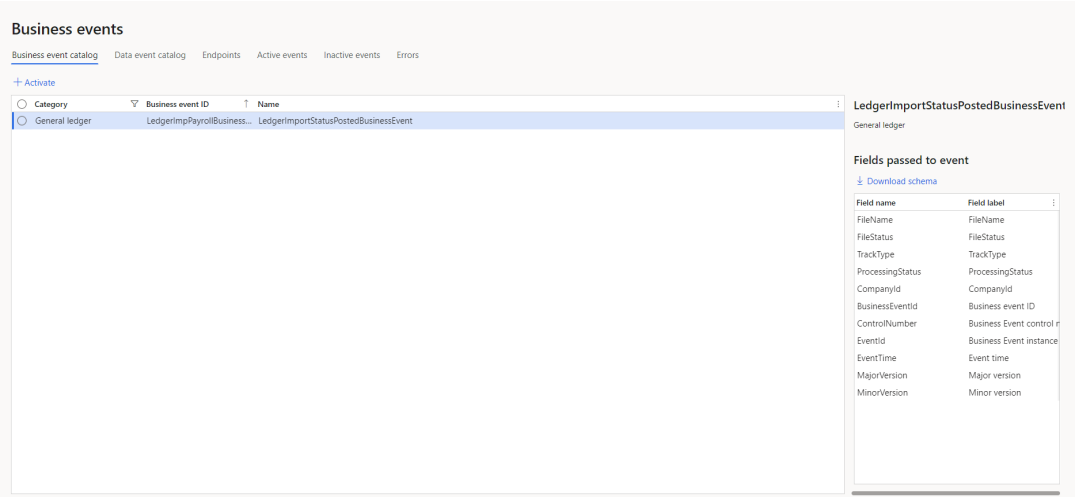


FIGURE 5.29: Business Event

Nella figura 5.29 viene visualizzato il business event creato all'interno di Dynamics365 FO nel business event catalog. Nel lato destro della pagina si possono visualizzare i campi che sono stati specificati nel codice.

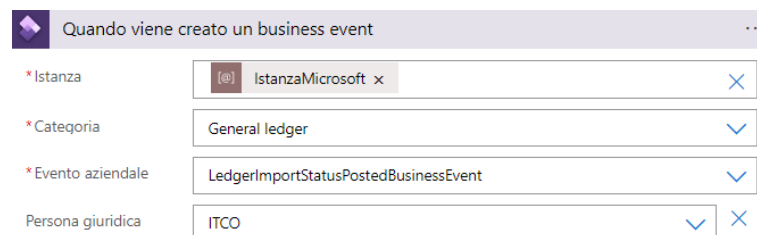
## 5.5 Creazione della seconda Logic App

Come è stato descritto nella sezione 5.1, gli eventi aziendali sono uno strumento che consente ai componenti esterni di ricevere notifiche direttamente da D365 Finance and Operations. I componenti esterni possono quindi eseguire attività aggiuntive in risposta all'evento aziendale attivato.

Perciò, una volta che il giornale contabile è stato creato all'interno di Dynamics 365 FO, il business event viene eseguito automaticamente e genera un carico di dati. A questo punto, la logic app riceve il carico di dati ed esegue le operazioni necessarie a spostare nuovamente il file una volta che è stato elaborato.

- **Passo 1:** *Quando viene creato un business event*

L'evento aziendale creato apparirà come trigger nella Logic App, in questo modo si ottengono le informazioni contenute all'interno della tabella Ledger-ImpPayrollStausTable\_ALTC.



The screenshot shows the configuration for a Logic App trigger titled "Quando viene creato un business event". The configuration includes the following fields:

Field	Value
* Istanza	IstanzaMicrosoft x
* Categoria	General ledger
* Evento aziendale	LedgerImportStatusPostedBusinessEvent
Persona giuridica	ITCO

FIGURE 5.30: Trigger creazione business event.

- **Passo 2: Analizza JSON**

Attraverso lo schema JSON viene importato il payload contenente i dati che sono stati ricevuti dal business event. All'interno del payload, infatti, vi sono i valori dei campi della tabella *LedgerImpPayrollStatusTable\_ALTC* aggiornati in seguito all'elaborazione del file.

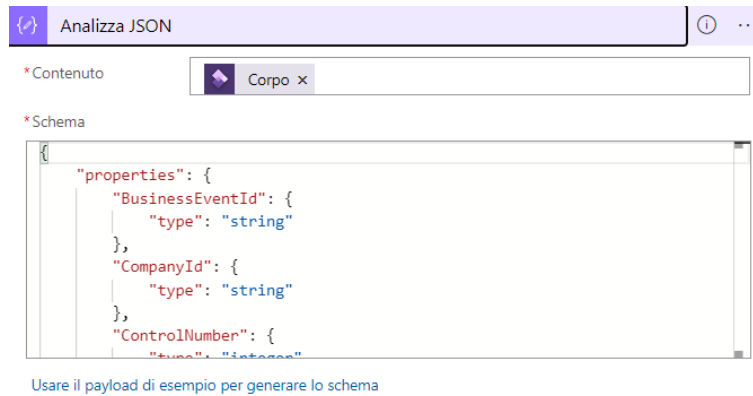


FIGURE 5.31: Schema JSON.

```
{
  "properties": {
    "BusinessEventId": {
      "type": "string"
    },
    "CompanyId": {
      "type": "string"
    },
    "ControlNumber": {
      "type": "integer"
    },
    "EventId": {
      "type": "string"
    },
    "EventTime": {
      "type": "string"
    },
    "FileName": {
      "type": "string"
    },
    "FileStatus": {
      "type": "string"
    },
    "MajorVersion": {
      "type": "integer"
    },
    "MinorVersion": {
      "type": "integer"
    }
  },
}
```

```

    "ProcessingStatus": {
      "type": "string"
    },
    "TrackType": {
      "type": "string"
    }
  },
  "type": "object"
}

```

- **Passo 3:** *Inizializza variabile ProcessingStatus a Elaborato*

Viene inizializzata la variabile Processing status che, come si è detto in precedenza, indica lo stato di elaborazione del giornale e viene posta pari a "Elaborato".

FIGURE 5.32: Action per inizializzazione del ProcessingStatus.

- **Passo 4:** *Se il Processing Status è uguale a Elaborato*

Tramite questa condizione viene verificato se lo stato di elaborazione del giornale è effettivamente pari ad "Elaborato".

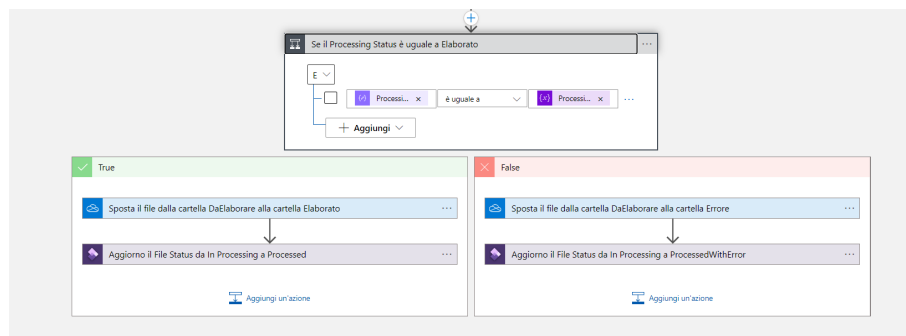


FIGURE 5.33: Condizione per il controllo degli errori.

Se la risposta è positiva, il file viene spostato nella cartella "Elaborato" tramite l'azione *Sposta o rinomina file in base al percorso*; quindi, viene specificato il percorso di origine del file e il percorso di destinazione.

Viene poi aggiornato il campo File Status della tabella *LedgerImpPayrollStatusTable* così da indicare che il file è stato processato e si è arrivati al termine dell'elaborazione.

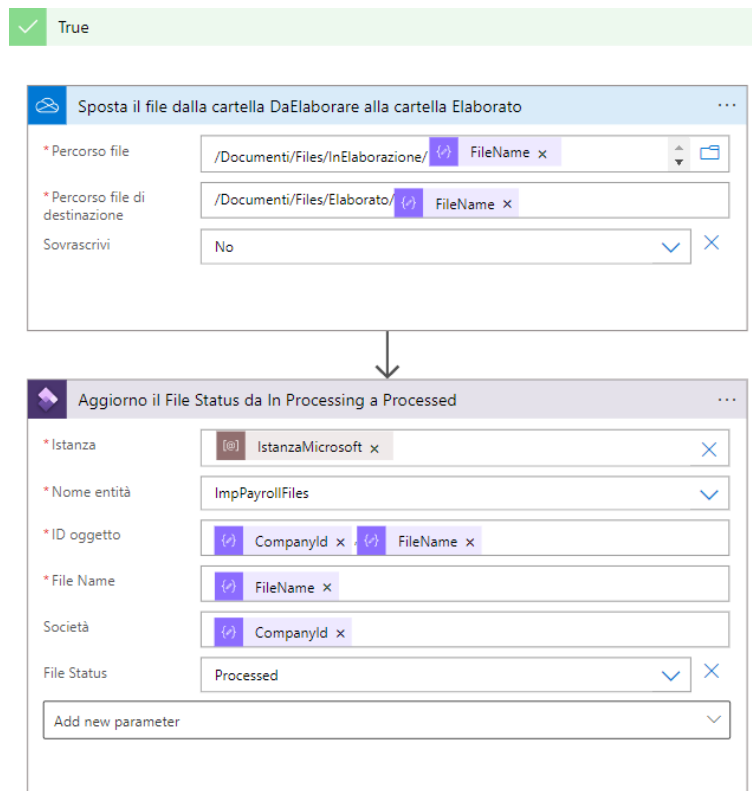


FIGURE 5.34: Ramo del flusso in caso di risposta affermativa.

Se la risposta è negativa, il file viene spostato nella cartella "Errore" tramite l'azione *Sposta o rinomina file in base al percorso*; quindi, come descritto al passaggio precedente, viene specificato il percorso di origine del file e il percorso di destinazione.

Viene poi aggiornato il campo File Status della tabella *LedgerImpPayrollStatusTable* così da indicare che il file è stato processato ma che si è verificato un errore.

False

Sposta il file dalla cartella DaElaborare alla cartella Errore

\* Percorso file: /Documenti/Files/InElaborazione/ FileName x

\* Percorso file di destinazione: /Documenti/Files/Errore/ FileName x

Sovrascrivi: No

Aggiorno il File Status da In Processing a ProcessedWithError

\* Istanza: IstanzaMicrosoft x

\* Nome entità: ImpPayrollFiles

\* ID oggetto: CompanyId x FileName x

\* File Name: FileName x

Società: CompanyId x

File Status: ProcessedWithError

Add new parameter

FIGURE 5.35: Ramo del flusso in caso di risposta negativa.

Infine, viene inviata una mail all'azienda cliente notificando l'errore.

False

Sposta il file dalla cartella DaElaborare alla cartella Errore

Aggiorno il File Status da In Processing a ProcessedWithError

Invia un messaggio di posta elettronica (v2)

\* A: bmaini@alteamet.it

\* Oggetto: Avviso Errore

\* Corpo: Carattere 12 **B** *I* U Si è verificato un errore nell'elaborazione della scrittura contabile.

Priorità: Normal

Add new parameter

FIGURE 5.36: Ramo del flusso in caso di risposta negativa pt.2.

## Chapter 6

# Risultati Sperimentali

In questo capitolo viene presentata una guida step by step per mettere in esecuzione il processo di importazione del file e creazione del giornale contabile.

### 1. Creazione del file all'interno di OneDrive.

Nome	Stato	Ultima modifica	Tipo	Dimensione
PNC_2021_09_001173_1	✓	02/03/2022 10:35	Documento di testo	264 KB
PNC_2021_09_001173_1 - Copia	✓	02/03/2022 10:35	Documento di testo	264 KB
InElaborazione	✓	30/08/2022 13:34	Cartella di file	
Errore	✓	24/05/2022 09:40	Cartella di file	
Elaborato	✓	30/08/2022 13:40	Cartella di file	
DaElaborare	✓	24/05/2022 09:36	Cartella di file	

FIGURE 6.1: Step 1 - Creazione del file in OneDrive.

### 2. Esecuzione della prima logic app.

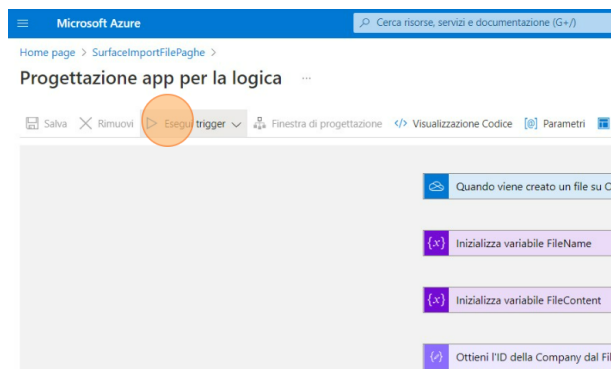


FIGURE 6.2: Step 2 - Esecuzione prima logic app.

3. Tutti i passi del flusso di lavoro sono stati eseguiti correttamente, il file è stato importato.

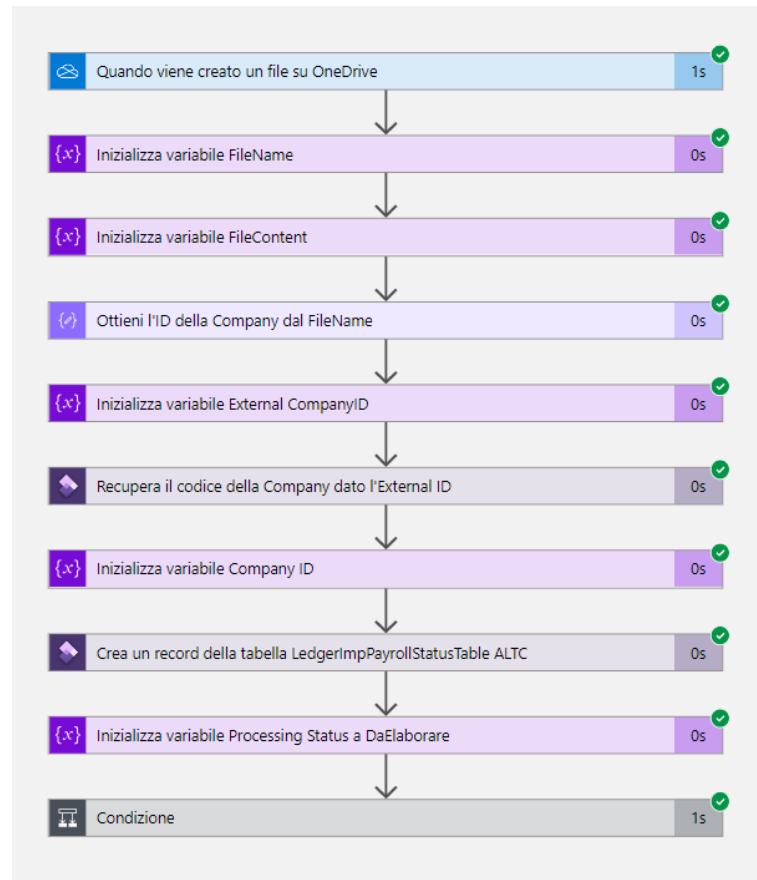


FIGURE 6.3: Step 3 - Corretta esecuzione della prima logic app.

4. All'interno della tabella *LedgerImpPayrollStatusTable\_ALTC* viene inserito il primo record, contenente tutte le informazioni del file importato. Da notare la valorizzazione del campo *FileStatus* = "InProcessing" e il campo *ProcessingStatus* = "DaElaborare".

Table browser: LedgerImpPayrollStatusTable\_ALTC

Standard view

CompanyId	Filename	FileStatus	IncomingFile	ProcessingStatus	TrackType	dataAreaId	recVersion	Partition	
ITCO	PNC_2021_09_001173_1 - Copia.txt	In Processing	20210911730000 00	Da Elaborare	Payroll	ITCO	1559763555	5637144576	56

FIGURE 6.4: Step 4 - LedgerImpPayrollStatusTable\_ALTC .



5. Dalla home page di Dynamics 365 for Finance and Operations si accede all'action menu item che permette di dare inizio all'elaborazione del file importato all'interno di Dynamics; quindi, viene creato il giornale contabile e il business event che metterà in esecuzione la seconda app per la logica.

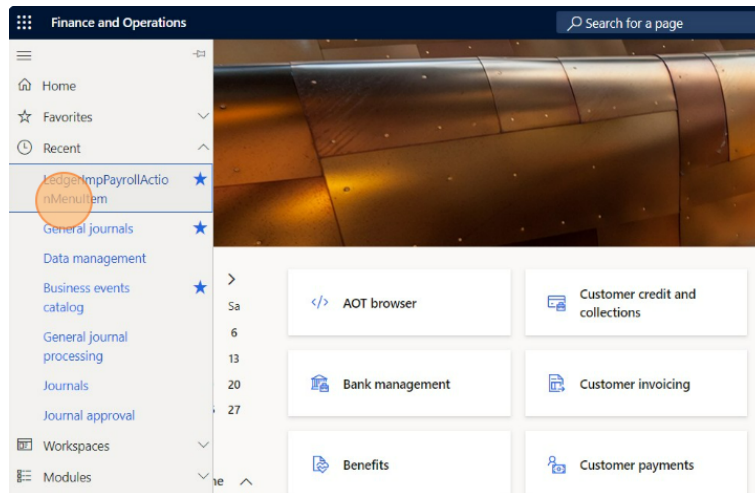


FIGURE 6.5: Step 5 - LedgerImpPayrollActionMenuItem.

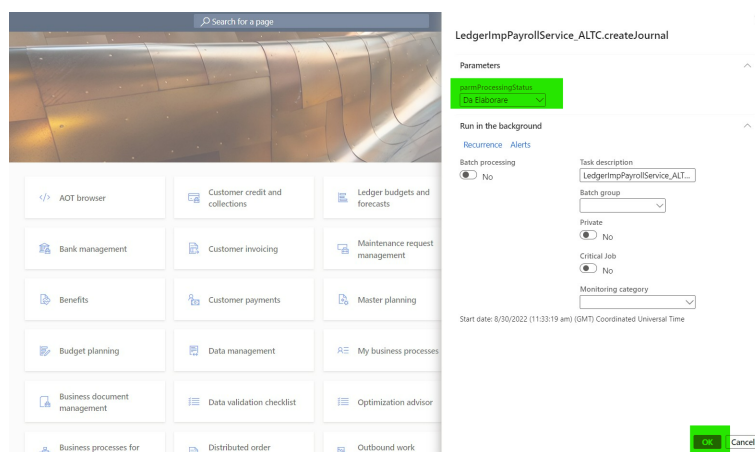


FIGURE 6.6: Step 5 - LedgerImpPayrollActionMenuItem pt.2.

6. Grazie alla generazione del business event nel passo precedente, la seconda logic app ha iniziato la sua elaborazione e tutti i passi del flusso sono stati eseguiti correttamente.

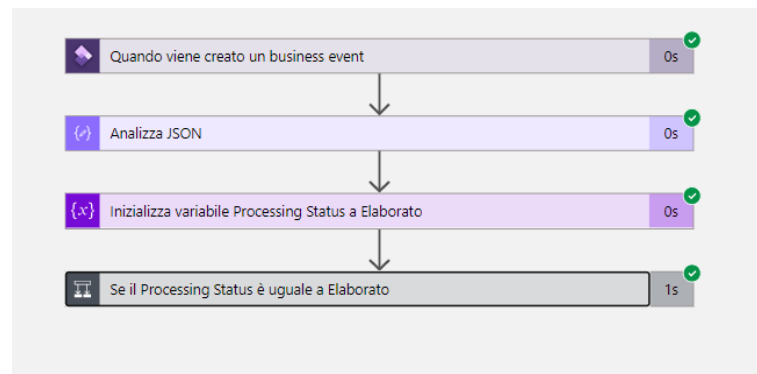


FIGURE 6.7: Step 6 - Corretta esecuzione della seconda logic app.

7. Tornando all'interno della tabella LedgerImpPayrollStatusTable\_ALTC, i campi FileStatus e ProcessingStatus sono stati aggiornati, assumendo rispettivamente i valori "Processed" e "Elaborato".

Table browser: LedgerImpPayrollStatusTable\_ALTC

Standard view

CompanyId	Filename	FileStatus	IncomingFile	ProcessingStatus	BackType	dataAreaId	recVersion	Partition	
ITCO	PNC_2021_09_001173_1 - Copia.bt	Processed	20210911730000 00	Elaborato	Payroll	Itco	488490282	5637144576	56

FIGURE 6.8: Step 7 - LedgerImpPayrollStatusTable\_ALTC.

8. Infine, è possibile visualizzare il giornale contabile generato al passo 5 cliccando su GeneralJournals all'interno della Home Page di Dynamics 365 FO.

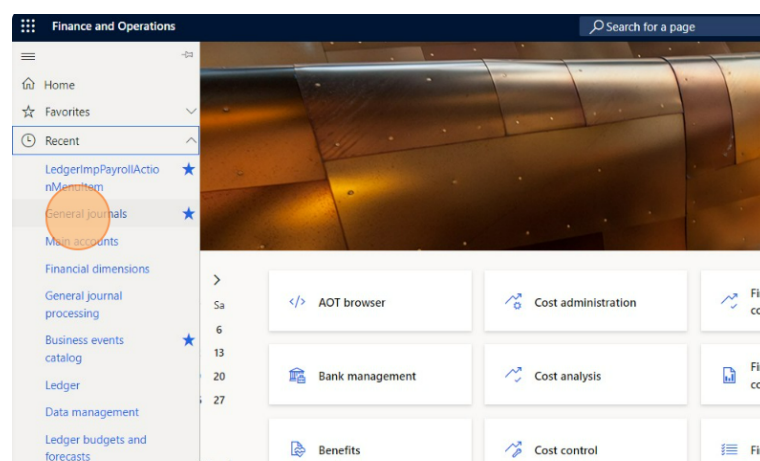


FIGURE 6.9: Step 8 - General Journals.

ITCO-000793 : GL DAILY | Standard view

Journal voucher

Display journal lines

All

ListGeneralInvoicePaymentPayment feeFixed assetsRemittanceHistory

+ NewDeleteVoucherFinancial dimensionsSales taxFunctions

	Date	Voucher	Company	Account type	Account	Description	Debit	Credit	Offset com...	Offset account type	Offset account	Offset transaction
	8/30/2022	GL000002485	itco	Ledger			1.34		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger			0.04		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger				98.25	itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger				279.45	itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger			279.45		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger			37.51		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger				37.51	itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger			37.51		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger				37.51	itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger			32.36		itco	Ledger		
	8/30/2022	GL000002485	itco	Ledger				10.46	itco	Ledger		

Account name

Offset account name

Calculated sales tax amount

Actual sales tax amount

Salaries - Other Expe...0.000.00

CURRENCY

	DEBIT	CREDIT	BALANCE
VOUCHER	688,071.81	688,071.81	0.00
JOURNAL	688,071.81	688,071.81	0.00

REPORTING CURRENCY

	DEBIT	CREDIT	BALANCE
VOUCHER	688,071.81	688,071.81	0.00
JOURNAL	688,071.81	688,071.81	0.00

Budget check results

Budget check not performed

FIGURE 6.10: Step 6 - Giornale contabile realizzato.



## Chapter 7

# Conclusione

L'obiettivo della presente tesi è stato quello di realizzare un sistema di integrazione che permettesse di importare un file di paghe e note spese da OneDrive a Dynamics 365 for Finance and Operations. In seguito, è stato implementato il codice per la creazione della scrittura contabile necessaria per la registrazione delle transazioni contenute nel file.

Nella prima parte del seguente elaborato sono stati affrontati i concetti teorici riguardanti i sistemi cloud-based scendendo nel particolare delle Azure Logic Apps. E' stato poi presentato un capitolo che ha portato avanti un'analisi concettuale dei principi alla base del sistema di integrazione riguardanti Dynamics 365 e Visual Studio .

Dopodiché, è stata inserita un'overview con la presentazione dello scopo del progetto, il funzionamento del processo e come sarebbe stato realizzato. Entrando poi nel particolare della sperimentazione vera e propria, è stata descritta l'analisi e la mappatura dei dati inviati dallo studio contabile delle aziende clienti.

In seguito, è stata descritta la progettazione delle applicazioni per logica e il codice necessario alla realizzazione della scrittura contabile.

L'utilizzo delle Logic Apps, per mettere in atto la migrazione del file, è stato l'elemento cardine al fine di realizzare il progetto. Sono state create due applicazioni per la logica adibite a due scopi differenti. La prima, contenente un flusso di lavoro in grado di inviare il file di paghe/note spese a Dynamics365, con apposita verifica di corretta importazione. La seconda, con all'interno il flusso di lavoro per la ricezione dei dati elaborati e il controllo della corretta creazione del giornale contabile, con eventuale notifica di errori.

Per quanto concerne la parte di creazione della scrittura contabile, è stata implementata tramite codice scritto in Visual Studio. Grazie al giornale contabile è stato dunque possibile registrare le transazioni contenute all'interno del file di note spese con il metodo della partita doppia.

Una volta implementato tutto il codice, e aver messo in esecuzione le due app per la logica, è stato possibile osservare il corretto funzionamento del processo.

Tramite questo lavoro di tesi sono stati messi in luce i vantaggi delle Azure Logic Apps, che permettono di automatizzare flussi di lavoro grazie a connessioni con altri sistemi tramite l'utilizzo di azioni e trigger. Ciò che di più interessante ne consegue è la possibilità di sviluppare rapidamente soluzioni di integrazione altamente scalabili per gli scenari aziendali.



## Chapter 8

# Bibliografia

- [1] Altea, Alterna Federation, **2022**, <https://alteafederation.it/at-a-glance/>.
- [2] Alterna, Alterna SRL, **2022**, <https://www.alternanet.it/soluzioni/digital-experience/smart-working/>.
- [3] A. Kumar, S. Mahendrakar, Serverless Integration Design Patterns with Azure: Build powerful cloud solutions that sustain next-generation products, **2019**, [https://books.google.it/books?hl=it&lr=&id=KB2IDwAAQBAJ&oi=fnd&pg=PP1&dq=logica+apps+microsoft&ots=AYpSVwktAu&sig=UPBTJgcbhZ4bgvX131CfGXyuK9w&redir\\_esc=y#v=onepage&q&f=false](https://books.google.it/books?hl=it&lr=&id=KB2IDwAAQBAJ&oi=fnd&pg=PP1&dq=logica+apps+microsoft&ots=AYpSVwktAu&sig=UPBTJgcbhZ4bgvX131CfGXyuK9w&redir_esc=y#v=onepage&q&f=false).
- [4] Noaa, What are Azure Logic Apps and how are they used?, **2015**, <https://www.netreo.com/cloud-automation/what-are-azure-logic-apps-and-how-are-they-used/>.
- [5] A. Velagana, Overview of Azure Logic Apps, **2022**, <https://mindmajix.com/overview-of-azure-logic-apps>.
- [6] Microsoft, Overview of Azure Logic Apps, **2022**, <https://docs.microsoft.com/en-us/azure/logic-apps/logic-apps-overview>.
- [7] Microsoft, Informazioni sui connettori in App per la logica di Azure, **2022**, <https://docs.microsoft.com/en-us/azure/connectors/apis-list>.
- [8] Microsoft, Connettori predefiniti nelle app per la logica di Azure, **2022**, <https://docs.microsoft.com/en-us/azure/connectors/built-in>.
- [9] Microsoft, Connettori gestiti in App per la logica di Azure, **2022**, <https://docs.microsoft.com/en-us/azure/connectors/managed>.
- [10] Microsoft, Pianificazioni per trigger ricorrenti nei flussi di lavoro di App per la logica di Azure, **2022**, <https://docs.microsoft.com/en-us/azure/logic-apps/concepts-schedule-automated-recurring-tasks-workflows>.
- [11] Microsoft, Introduzione a Microsoft Dynamics 365, **2022**, <https://docs.microsoft.com/it-it/learn/modules/dynamics-365-introduction/1-introduction>.
- [12] Microsoft, Digital Transformation, **2022**, <https://docs.microsoft.com/it-it/learn/modules/dynamics-365-introduction/2-digital-transformation-business-transformation>.
- [13] Microsoft, Microsoft Business Applications, **2022**, <https://docs.microsoft.com/it-it/learn/modules/dynamics-365-introduction/3-microsoft-business-applications>.
- [14] Microsoft, Welcome to the Visual Studio IDE, **2022**, <https://docs.microsoft.com/it-it/visualstudio/get-started/visual-studio-ide?view=vs-2019>.

- [15] Microsoft, Organizations and organizational hierarchies overview, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/fin-ops/organization-administration/organizations-organizational-hierarchies?toc=\%2Fdynamics365\%2Fcommerce\%2Ftoc.json>.
- [16] Microsoft, Create legal entities, **2022**, <https://docs.microsoft.com/en-us/dynamics365/commerce/channels-legal-entities>.
- [17] Microsoft, Process a journal entry, **2022**, <https://docs.microsoft.com/en-us/learn/modules/general-ledger-daily-procedures-dyn365-finance/3-process-journal-entry>.
- [18] Microsoft, Microsoft Dynamics 365 for Financials – Using General Journals, **2022**, <https://www.loganconsulting.com/blog/microsoft-dynamics-365-for-financials-using-general-journals/#:~:text=General\%20journals\%20are\%20used\%20to,entries\%20on\%20general\%20ledger\%20accounts..>
- [19] Microsoft, Perform general ledger daily procedures in Dynamics 365 Finance, **2019**, <https://docs.microsoft.com/en-us/learn/modules/general-ledger-daily-procedures-dyn365-finance/>.
- [20] TeamSystem, Libro Giornale o Libro Contabile: Cos'è?, **2022**, <https://www.teamssystem.com/store/dizionario/libro-giornale/>.
- [21] IONOS, Il metodo della partita doppia spiegato semplicemente, **2019**, <https://www.ionos.it/startupguide/gestione/il-metodo-della-partita-doppia-spiegato-semplimente/>.
- [22] NEX, Financial dimensions in Dynamics 365 for Finance and Operations, **2022**, <https://www.nexsoftsys.com/articles/financial-dimensions-in-dynamics-365-services.html>.
- [23] I. Brandon, D365 LEDGER ACCOUNT DIMENSIONS PART 1: TIPS TO OPTIMIZE FINANCIALS, **2022**, <https://www.instructorbrandon.com/d365-ledger-account-dimensions-part-1-tips-to-optimize-financials/>.
- [24] Microsoft, Ledger account combinations, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/financial/ledgeraccountcombinations>.
- [25] Microsoft, Default financial dimensions, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/financial/dimension-defaulting>.
- [26] githubUser, Default financial dimensions, **2022**, <https://github.com/MicrosoftDocs/Dynamics-365-Unified-Operations-Public/blob/main/articles/fin-ops-core/dev-itpro/financial/dimension-defaulting.md>.
- [27] Microsoft, Main account information, **2022**, <https://docs.microsoft.com/en-us/learn/modules/configure-chart-accounts-dyn365-finance/6-main-account-info>.
- [28] Microsoft, Business events overview, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/business-events/home-page>.
- [29] Microsoft, Manage business event endpoints, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/business-events/managing-business-event-endpoints?source=recommendations>.



- [30] P. Ramer, SysOperationFramework in Dynamics 365, **2021**, <https://dynamics365musings.com/sysoperation-framework-in-d365/#:~:text=SysOperation%20Framework%20in%20D365%20can,some%20process%20on%20a%20record..>
- [31] Microsoft, Understand the different types of journals and their setup, **2019**, <https://docs.microsoft.com/en-us/learn/modules/general-ledger-daily-procedures-dyn365-finance/>.
- [32] Microsoft, Business events developer documentation, **2022**, <https://docs.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/business-events/business-events-dev-doc>.