

Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA “TULLIO LEVI-CIVITA”

CORSO DI LAUREA IN INFORMATICA



**Integrazione di processi PowerAutomate
all'interno di applicazioni aziendali**

Tesi di laurea

Relatore

Prof. Tullio Vardanega

Laureando

Silvio Nardo

ANNO ACCADEMICO 2023-2024

Sommario

Il presente documento descrive il lavoro svolto dal laureando Silvio Nardo durante il periodo di stage presso l'azienda Wintech S.p.A. di Padova.

Esso è diviso in quattro capitoli: "Contesto aziendale" descrive l'azienda ospitante, con particolare attenzione ai servizi e prodotti offerti e alle metodologie lavorative; "Progetto di stage" narra il rapporto che l'azienda ha con gli stage universitari, descrivendo i diversi progetti proposti con particolare dettaglio allo stage da me svolto; In "Svolgimento stage" sono contenute le informazioni relative alle attività da me svolte durante il percorso di stage con spiegazione dei risultati raggiunti; "Verifica retrospettiva" infine analizza le conoscenze acquisite durante questo percorso e il loro rapporto con quelle fornite dall'università nel corso di laurea da me frequentato.

Lo stage si è svolto in conclusione del percorso di studi della laurea triennale in Informatica ed ha avuto la durata di circa trecentoventi ore.

L'obiettivo dello stage è stato compiere un'analisi al fine di valutare l'applicabilità delle pratiche DevOps a progetti aziendali realizzati con gli strumenti Power Automate e Power Apps.

Le soluzioni individuate durante le attività di ricerca sono state integrate ai processi aziendali mediante fasi di sviluppo collaborativo e individuale.

Convenzioni tipografiche

Gli acronimi e i termini di uso non comune menzionati vengono definiti nel glossario, situato alla fine del presente documento.

I termini in lingua straniera sono evidenziati con il carattere *corsivo*.

Indice

1	Svolgimento stage	1
1.1	Analisi	1
1.1.1	Requisiti	1
1.1.2	Ambiente di lavoro	3
1.1.3	Tecnologie oggetto di stage	3
1.1.4	Tecnologie strumentali	7
1.1.5	Microsoft Solutions	8
1.1.6	DevOps	9
1.1.7	Applicazioni aziendali	9
1.1.8	Angular	11
1.2	Progettazione	11
1.3	Programmazione	11
1.4	Verifica e Validazione	11
1.5	Risultati raggiunti	11
1.5.1	Qualitativamente	11
1.5.2	Quantitativamente	11
	Glossario	12
	Bibliografia	13

Elenco delle figure

1.1	Moduli formativi per Power Automate.	4
1.2	Esempio di un flusso Power Automate istantaneo.	5
1.3	Elemento Tree View per la gestione gerarchica dei componenti.	6
1.4	Esempio interfaccia di Jenkins.	8
1.5	Interfaccia grafica di Report di visita.	10

Elenco delle tabelle

1.1	Tabella dei requisiti progettuali.	2
-----	--	---

Capitolo 1

Svolgimento stage

In questo capitolo vengono descritte tutte le attività da me svolte durante lo *stage*, divise nelle sezioni “Analisi”, “Progettazione”, “Programmazione” e “Verifica e validazione”, in modo da fornire una panoramica chiara e strutturata del lavoro svolto, evidenziando il processo seguito e le competenze acquisite in ciascuna fase.

Esse non sono intese come completamente sequenziali bensì, fin dalla prima fase, sono presenti tutte le attività correlate in modo da coprire l'intero periodo di *stage*.

1.1 Analisi

In questa sezione sono presenti tutte le attività analitiche da me svolte e il suo scopo è descrivere le modalità con cui ho compreso i bisogni, i requisiti e le tecnologie del mio progetto di *stage*.

Essendo quest'ultimo focalizzato per la maggior parte su attività legate alla ricerca e all'esplorazione tecnologica, la fase di analisi rappresenta la parte più corposa delle mie attività.

1.1.1 Requisiti

I requisiti del mio *stage*, derivanti in parte dalle dichiarazioni aziendali presenti nel documento “Progetto Formativo” generato all'inizio del suo svolgimento, e in parte conseguenza dell'analisi dei requisiti avvenuta in corrispondenza con la presenza di specifiche necessità progettuali, sono divisi in categorie:

- O - requisiti obbligatori, vincolanti in quanto obiettivi primari richiesti dall'azienda.
- D - requisiti desiderabili, non strettamente necessari ma dal riconoscibile valore aggiunto.
- F - requisiti facoltativi / opzionali, rappresentanti un valore aggiunto non strettamente competitivo.

Essi sono:

Codice	Descrizione
O1	Comprensione e mappatura delle funzionalità possibili tramite l'adozione dell'applicazione Microsoft Power Automate.
O1.1	Analisi approfondita riguardo ai flussi Power Automate ed individuazione delle caratteristiche, positive e negative, della sua adozione in azienda.
O1.2	Sviluppo collaborativo sui flussi Power Automate.
O1.3	Produzione ed esposizione al <i>tutor</i> aziendale e al <i>team</i> di sviluppo di una presentazione riguardo ai risultati della mia analisi sull'adozione di Power Automate in azienda.
O2	Comprensione ed utilizzo della tecnologia Microsoft Power Apps e relative fasi di sviluppo collaborativo.
O2.1	Comprensione di Power Apps affiancando il <i>team</i> di sviluppo fino al raggiungimento della comprensione dei metodi lavorativi e dei prodotti aziendali sviluppati con tale strumento.
O2.2	Identificazione e definizione di un metodo adatto allo sviluppo collaborativo di applicazioni Power Apps.
O2.3	Ottenimento di avanzamenti nello stato dei lavori di applicazioni aziendali realizzate con Power Apps.
O3	Individuazione ed implementazione di un efficace sistema di versionamento per progetti realizzati utilizzando tecnologie Power Automate e Power Apps
O4	Comprensione delle metodologie DevOps e realizzazione della documentazione relativa alla sua applicabilità su tecnologie Power Automate e Power Apps.
D1	Collaborazione con gli altri stagisti universitari e individuazione di soluzioni collaborative relative ai bisogni progettuali comuni.
D2	Realizzazione di PoC a supporto delle soluzioni individuate durante le fasi di ricerca.
D2.1	Realizzazione di PoC riguardo allo sviluppo di flussi Power Automate approvativi.
D2.2	Realizzazione di PoC riguardo all'integrazione tra flussi Power Automate ed elementi esterni tramite le chiamate HTTP .
D2.3	Realizzazione di PoC riguardo al processo DevOps di <i>build</i> su progetti realizzati con tecnologie Power Automate/Power Apps.
D2.4	Realizzazione di PoC riguardo all'applicazione di analisi statica del codice su progetti realizzati con tecnologie Power Automate/Power Apps.
D2.5	Realizzazione di PoC riguardo al processo DevOps di <i>test</i> su progetti realizzati con tecnologie Power Automate/Power Apps.
D3	Esecuzione di <i>meeting</i> e scambio di documenti con gli altri stagisti universitari al fine di comprendere l'applicabilità delle fasi di DevOps in ambito Sistemi
F1	Esplorazione della tecnologia Angular mediante realizzazione di un progetto di <i>test</i>
F2	Integrazione di un progetto Angular con le fasi di DevOps mediante un <i>server</i> Jenkins.
F3	Realizzazione ed esposizione di una presentazione finale, in collaborazione con gli altri stagisti, riguardo tutto il lavoro fatto durante lo <i>stage</i> .

Tabella 1.1: Tabella dei requisiti progettuali.

1.1.2 Ambiente di lavoro

Durante i primi giorni dello *stage* sono stato introdotto all'ambiente di lavoro e alle tecnologie di comunicazione e collaborazione.

Consequentemente ho analizzato e utilizzato il sistema di messaggistica basato su Microsoft Teams e Outlook e ho compreso la struttura di condivisione dei dati, utilizzata in azienda, basata sullo strumento Microsoft SharePoint.

Esso è una piattaforma *software* in grado di organizzare dati sottoforma di *file* e strutture tabellari chiamate "Liste", al fine di gestire il materiale condiviso dall'azienda e dai singoli *team* tramite un sistema di accessi e autorizzazioni.

Tramite questi strumenti ho studiato i documenti aziendali a me forniti in modo da comprendere i principali processi produttivi di Wintech.

Tali documenti comprendono i "Documenti di sviluppo sicuro" e includono:

- Agile e SCRUM: descrizione delle metodologie Agile e SCRUM, spiegazione dei ruoli necessari e delle cerimonie previste.
- Presentazione sviluppo sicuro: presentazione PowerPoint che descrive i processi aziendali atti a migliorare la qualità dei prodotti realizzati automatizzando fasi ripetitive e rispettando criteri di sicurezza.
- Modelli di sviluppo sicuro: elenco dettagliato dei documenti di sviluppo sicuro i quali descrivono come applicare automazioni processuali (per esempio i processi di *build* e *deploy*) in modo sicuro e normato.
- Politiche di sviluppo sicuro: strategie e normative aziendali definite al fine di garantire sicurezza e qualità nei processi e nel ciclo di vita del *software*.
- Piani di progetto degli altri stagisti: piani formativi degli altri due stagisti che nel mio stesso periodo hanno effettuato lo *stage* universitario in Wintech.

Relativamente a quest'ultimo punto, nei primi giorni ho approfondito il lavoro svolto dagli altri stagisti tramite appositi *meeting* nei quali mi hanno descritto i risultati ottenuti fino a quel momento. Essi, avendo iniziato lo svolgimento del progetto circa una settimana prima, mi hanno esposto, mediante apposite presentazioni PowerPoint, le proprie ricerche riguardanti l'utilizzo dello strumento Git e dello studio avvenuto riguardo la possibilità di integrare tra loro gli strumenti Planner e Taiga.

1.1.3 Tecnologie oggetto di stage

In questo sottocapitolo vengono descritti i metodi di apprendimento e le nozioni apprese durante la fase di analisi delle tecnologie su cui si basa la ricerca del mio progetto di *stage*.

Dopo aver compreso le tecnologie e i principali processi aziendali, ho partecipato ad un *meeting* con il *tutor* aziendale al fine di discutere il mio progetto di *stage*.

I requisiti scaturiti da tale incontro sono stati:

- Autoapprendimento dello strumento Power Automate.
- Realizzazione di un PoC che testasse la possibilità di realizzare un flusso approvativo Power Automate.
- Testare le funzionalità disponibili con le licenze di utilizzo *standard*.

Power Automate

Ho pertanto studiato approfonditamente tali tecnologie con l’ausilio delle numerose guide e *tutorial* offerti da Microsoft. Essi sono direttamente accessibili dalla *home* di Power Automate e Power Apps e sono divisi in moduli testuali corredati da immagini, dalla durata e argomenti specifici.

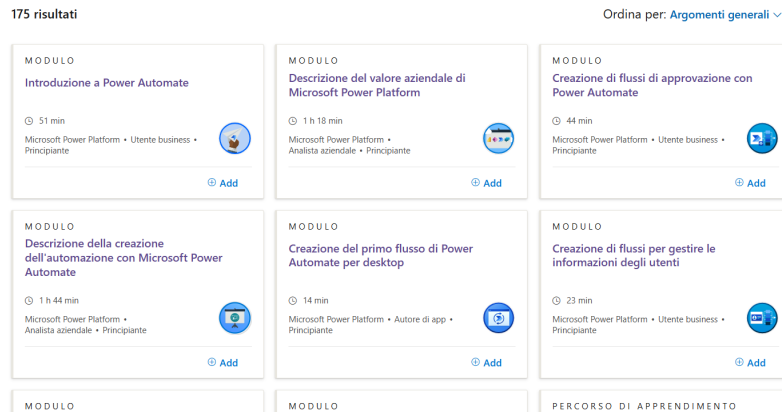


Figura 1.1: Moduli formativi per Power Automate.

Fonte: <https://learn.microsoft.com/it-it/training/browse/?products=power-automate>.

[//learn.microsoft.com/it-it/training/browse/?products=power-automate](https://learn.microsoft.com/it-it/training/browse/?products=power-automate).

Lo studio di tali moduli mi ha permesso di comprendere il funzionamento e le *features* principali di Power Automate:

Esso è formato da una pagina *web* che offre controllo sui flussi di automazione creati: è possibile modificarli e visualizzarne i dettagli, le esecuzioni e le statistiche. È possibile creare dei nuovi flussi partendo da altri progetti pubblici, o a noi condivisi, e modelli offerti da Microsoft da adattare alle proprie esigenze.

La modifica di un flusso non avviene mediante la scrittura di codice tramite un linguaggio di programmazione bensì tramite una composizione “a blocchi” personalizzabili collegati tra loro ciascuno avente proprietà e attributi definiti.

Essi sono selezionabili da una lista di blocchi relativi ciascuno a una funzionalità specifica di un servizio Microsoft.

Tali blocchi possono essere “Trigger” o “Azioni” ed entrambi sono necessari per la creazione e il funzionamento di un flusso.

In ogni flusso è presente uno e un solo Trigger il quale rappresenta il suo punto di partenza nonché la condizione che scatuisce la sua esecuzione.

Esistono tre tipologie principali di Trigger le quali determinano la tipologia stessa di ogni flusso:

- Automatico: per esempio “SharePoint - Quando viene creato un elemento”.
- Istantaneo: per esempio “Attiva manualmente un flusso”.
- Pianificato: per esempio “Ricorrenza”, il quale attiva il flusso periodicamente.

Ad ogni Trigger possono essere collegate, in serie o in parallelo, una moltitudine di Azioni, ciascuna responsabile di uno specifico compito, per esempio sono presenti le

azioni “Inizializza variabile”, “Avvia e attendi un’approvazione”, “Teams - Crea una chat” e “Outlook - Invia un messaggio di posta elettronica”.

Sono inoltre presenti azioni dedicate alla gestione logica dei flussi come “Condizione” e “Do until”. La prima rappresenta la struttura di controllo rappresentata nei classici linguaggi di programmazione con “if”, responsabile della ramificazione dell’esecuzione del flusso in base a una condizione specifica.

La seconda rappresenta la struttura di controllo rappresentata nei classici linguaggi di programmazione con “Do while”, responsabile della ripetizione condizionata di un insieme di azioni garantendone sempre la prima esecuzione. Successivamente è emersa,

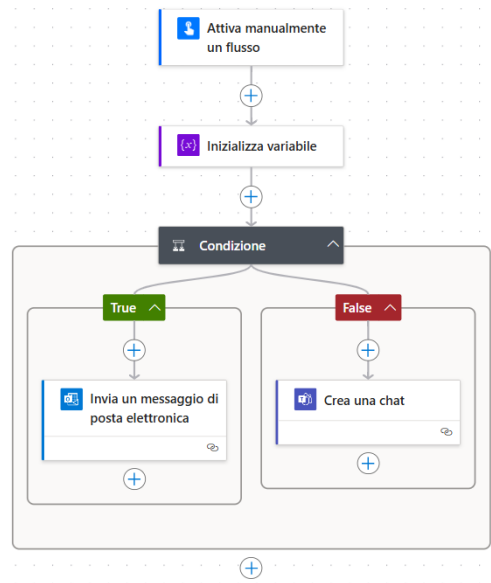


Figura 1.2: Esempio di un flusso Power Automate istantaneo.

da parte del tutor aziendale, la necessità di integrare i flussi Power Automate con il *software* gestionale WOW e gli altri prodotti aziendali al fine di poter integrare le funzionalità desiderate con libertà mantenendo coordinate le diverse parti del prodotto. Sono emersi quindi due requisiti: il primo, utile per apprendere le tecnologie in oggetto, è relativo alla creazione di un flusso che automatizzi un processo approvativo.

Il secondo, più concreto e integrabile con i prodotti aziendali, è basato sull’applicazione, ai flussi Power Automate, di chiamate [HTTP](#) (Hypertext Transfer Protocol): in italiano “protocollo di trasferimento ipertestuale” è un protocollo di rete, ovvero un insieme di regole formalmente descritte che definiscono le modalità di comunicazione tra due o più apparecchiature elettroniche, usato come principale sistema per la trasmissione d’informazioni sul *web*.

Esse permettono la comunicazione e lo scambio di dati tra flussi Power Automate e altri flussi o applicazioni: esiste infatti la possibilità di richiamare lo specifico Trigger “Alla ricezione di una richiesta HTTP” il quale genera un personale URL (Uniform Resource Locator), ovvero una sequenza di caratteri che identifica univocamente l’indirizzo di una risorsa su una rete di *computer*.

In seguito è possibile utilizzare la corrispondente azione “Response” al fine di rispondere al chiamante con l’*output* della richiesta.

Power Apps

I flussi Power Automate portano grande vantaggio in termini di efficienza dei processi aziendali e possono aumentare il numero di funzionalità di un'applicazione.

Per questo motivo il loro utilizzo assume maggiore significato quando associato alla realizzazione di applicazioni Power Apps, le quali sono pensate per integrare nativamente i flussi creati.

Il metodo di apprendimento con cui ho analizzato e studiato Power Apps deriva non solo da attività individuali tramite i moduli didattici offerti da Microsoft nell'apposito sito *web*, simili a quelli per Power Automate, ma soprattutto grazie alla collaborazione con un membro del *team* di sviluppo che ho affiancato. Egli è il responsabile in azienda di tutti i loro prodotti realizzati utilizzando le tecnologie oggetto del mio *stage* e assieme le abbiamo attentamente analizzate e modificate al fine di migliorarle ed applicarci le soluzioni da me individuate relativamente alle pratiche *DevOps* richieste.

Power Apps offre la possibilità di creare nuove applicazioni e visionare le applicazioni create, o a noi condivise, con la possibilità di apportare modifiche. È inoltre possibile visualizzare informazioni e statistiche relative a tutte le applicazioni in nostro possesso. Lo sviluppo di tali applicazioni non avviene esclusivamente mediante codice di programmazione bensì dall'utilizzo di componenti, *standard* o *custom*, che vengono manualmente posizionati sull'interfaccia, creando in questo modo una struttura gerarchica ad albero chiamata "Tree View".

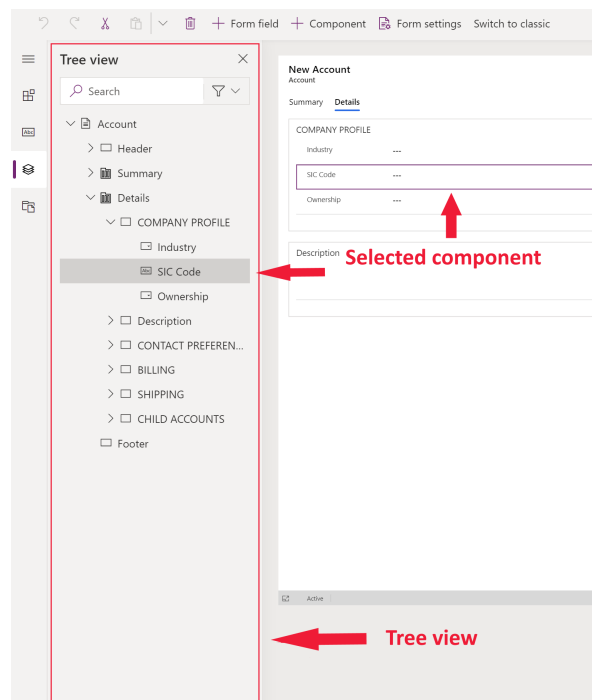


Figura 1.3: Elemento Tree View per la gestione gerarchica dei componenti.

Fonte: <https://learn.microsoft.com/en-us/power-apps/maker/model-driven-apps/using-tree-view-on-form>.

Al fine di modificare i parametri che lo compongono, ciascun componente può essere personalizzato tramite un'apposita finestra dell'interfaccia grafica. Questo non comprende solo le peculiarità grafiche ma soprattutto il loro comportamento a seguito di azioni specifiche: per esempio è possibile creare un componente "Pulsante" e personalizzarne il campo "OnSelect" al fine di eseguire conseguentemente un flusso Power Automate e inviarne i dati di *output* ad un altro componente.

Tali istruzioni vengono definite utilizzando Microsoft Power Fx, il quale rappresenta un linguaggio di programmazione dichiarativo, fortemente tipizzato e con uso limitato di codice, usato in Microsoft Power Platform.

Quest'ultima piattaforma include un insieme di strumenti e tecnologie, compresi Power Automate e Power Apps, pensati per offrire la possibilità di sviluppare prodotti *software* con limitata necessità di scrivere, e conoscere, codice tramite linguaggi di programmazione.

1.1.4 Tecnologie strumentali

In questo sottocapitolo vengono descritti i metodi di apprendimento e le nozioni apprese durante la fase di analisi degli strumenti e delle tecnologie adottate al fine di perseguire gli obiettivi e soddisfare i requisiti di *stage*.

Jenkins

Al fine di applicare le automazioni necessarie per adottare al *software* le metodologie [DevOps](#), come richiesto dagli obiettivi di *stage*, ho individuato lo strumento Jenkins in quanto esso rispecchia perfettamente le mie necessità ed è inoltre già conosciuto e applicato dal resto del *team* di sviluppo rappresentando quindi una tecnologia consolidata che non necessita di ulteriore formazione all'interno dell'azienda.

Ho appreso le conoscenze necessarie per studiare e utilizzare tale strumento in totale autonomia mediante la consultazione delle guide e del numeroso materiale presente *online* comprese le pagine *web* ufficiali di Jenkins con gli annessi video *tutorial*.

Esso è uno strumento scritto in linguaggio di programmazione Java che viene eseguito all'interno di un *web server* e offre la possibilità di creare, gestire, eseguire e monitorare dei progetti Jenkins chiamati "Job", potendone visionare e registrare i *report* di *output*. Essi sono unità di lavoro configurabili che rappresentano un'attività o un progetto specifico che Jenkins esegue. Possono essere di diverso tipo ma tutti sono basati sull'esecuzione di uno *script* o una *pipeline* di automazione.

Una *pipeline* Jenkins è una sequenza di fasi chiamate "Stage", eseguite in serie o in parallelo, responsabili di specifiche operazioni definite dall'utente, per esempio *build*, *test*, *deploy*. Esse possono essere definite tramite il linguaggio di programmazione Groovy oppure tramite l'interfaccia grafica di Jenkins.

Questo strumento permette dunque di fornire servizi di integrazione continua ed è indicato per l'applicazione di alcune fasi di [DevOps](#) come Build e Test, il caricamento automatico dei file sul *repository* di versionamento desiderato e l'autenticazione automatica ai servizi Microsoft tramite comandi Power Platform CLI.

Quest'ultima è un'interfaccia a riga di comando che offre la possibilità di eseguire un insieme di comandi specifici atti all'autenticazione e alla gestione degli ambienti e delle Microsoft Solutions.

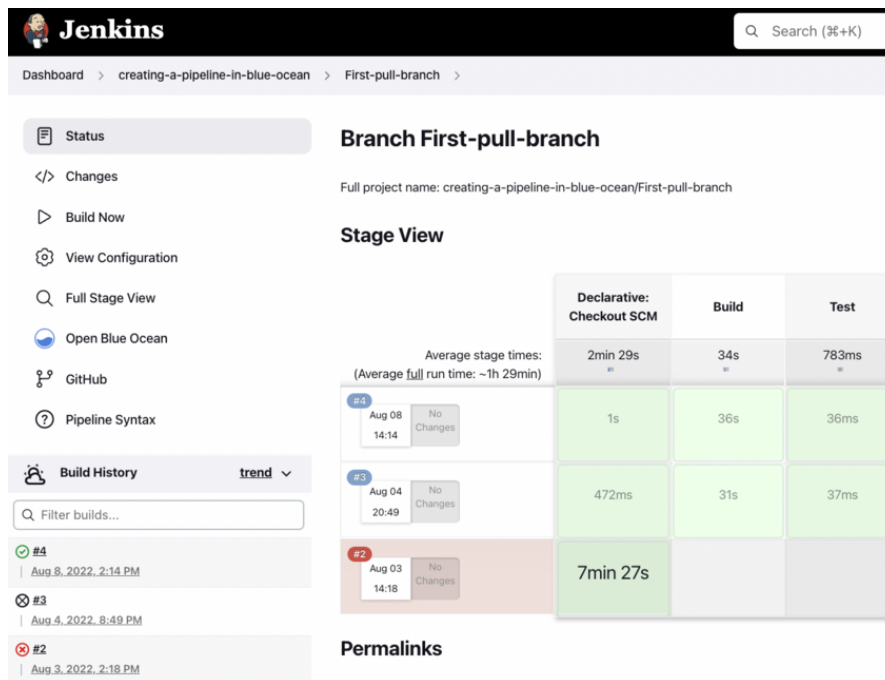


Figura 1.4: Esempio interfaccia di Jenkins.

Fonte: <https://commons.wikimedia.org/wiki/File:Updated-jenkins-view.png>.

1.1.5 Microsoft Solutions

Un'ulteriore oggetto di analisi e studio è stato l'utilizzo e la comprensione delle Microsoft Solutions. Esse fanno parte della *suite* Microsoft Power Platform e sono degli strumenti che servono a contenere tutti i flussi Power Automate e applicazioni Power Apps corrispondenti a uno stesso progetto, in modo da poterli sviluppare e distribuire in maniera organizzata e centralizzata.

Esistono due tipi di Soluzioni:

- **Gestite:** destinate prevalentemente per gli ambienti di produzione e *test*, non permettono la modifica degli elementi al suo interno. Per generare una Soluzione gestita è necessario esportare una soluzione non gestita.
- **Non gestite:** destinate prevalentemente per gli ambienti di sviluppo, permettono la modifica degli elementi al suo interno e la sua esportazione.

A differenza delle tecnologie precedentemente descritte, le Microsoft Solution non erano materia già affrontata in azienda, per cui il loro apprendimento è avvenuto in maniera totalmente autonoma mediante la documentazione presente nei siti *web* Microsoft dedicati.

Le Soluzioni preservano i loro dati, e quelli degli elementi in esse contenuti, nel sistema di archiviazione Microsoft Dataverse che funge da *database* centralizzato per applicazioni sviluppate con la Power Platform.

Tale adozione permette di usufruire di funzionalità altrimenti assenti. Le conseguenze del loro utilizzo e le motivazioni che mi hanno portato ad utilizzare questa tecnologia

sono materia di discussione presente nel capitolo 3.2 Progettazione.

1.1.6 DevOps

Plan

Code

Build

Test

Release

Deploy

Operate

Monitor

1.1.7 Applicazioni aziendali

Oltre all'analisi avvenuta sulle tecnologie al fine di raggiungere gli obiettivi riguardanti la ricerca sull'effettiva applicabilità delle pratiche di [DevOps](#) ai progetti realizzati con Power Automate e Power Apps, è stato fin da subito definito che avrei alternato tali attività con fasi di sviluppo su applicazioni aziendali.

Questo approccio mi ha permesso di approfondire la conoscenza delle tecnologie in oggetto e di diventare operativo in modo più efficace.

Insieme al membro del *team* di sviluppo responsabile per la realizzazione delle applicazioni Power Apps e dei flussi Power Automate aziendali, ho affrontato problemi di sviluppo di vario genere in funzione delle necessità del momento, potendo analizzare diversi prodotti aziendali.

Le effettive attività di programmazione svolte su queste applicazioni sono presenti al capitolo 3.3 Programmazione.

I principali prodotti su cui abbiamo collaborativamente lavorato sono:

Report di visita

Principale applicazione su cui ho lavorato e con cui ho svolto attività di analisi al fine di comprendere le tecnologie utilizzate. Essa ha l'obiettivo di offrire la possibilità ad un dipendente dell'azienda, il quale dovesse trovarsi nella sede di un cliente, di registrare e condividere con i colleghi, in un ambiente condiviso, un *report* della visita al fine di tracciare l'esperienza con il cliente.

Essa offre quindi la possibilità di:

- Ricercare e selezionare una specifica azienda visualizzabile in un'apposita lista a schermo
- Visualizzare le sedi e le filiali dell'azienda selezionata
- Visualizzare le informazioni anagrafiche dei contatti dell'azienda selezionata
- Visualizzare i *report* di visita già creati come bozza
- Modificare eventuali bozze selezionate

- Creare un nuovo documento inserendo in un apposito *form* i dati relativi all'operatore, al cliente e alla visita specifica
- Salvare tale *report* come bozza o come documento definitivo
- Caricamento automatico dei dati generati nel *database* aziendale

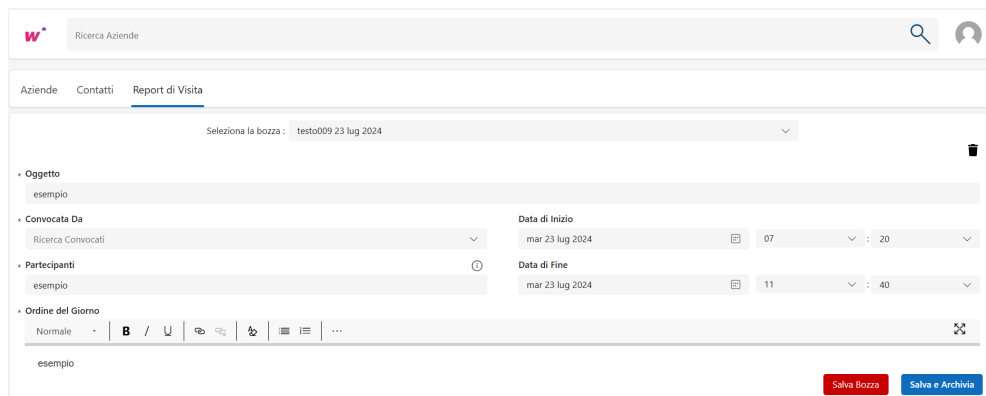


Figura 1.5: Interfaccia grafica di Report di visita.

Registro accessi

Applicazione realizzata al fine di gestire agevolmente i *check-in* e *check-out* lavorativi del personale.

Le funzionalità presenti sono:

- Identificazione del dipendente
- Selezione della sede, compresa la possibilità di selezionare lo *smart working* o lavoro fuorisede
- Selezione del *check-in*
- Visualizzazione di errore in caso di *check-in* in mancanza del precedente *check-out*
- Selezione del *check-out*
- Visualizzazione di errore in caso di *check-out* in mancanza del precedente *check-in*
- Visualizzazione di tutti i propri *check-in* e *check-out*

Non conformità

Nel momento in cui ho svolto lo *stage*, questa applicazione era nelle sue prime fasi di sviluppo e le sue funzionalità non erano state ancora definite pienamente.

I suoi obiettivi sono quelli di creare e gestire un flusso approvativo a più *step*, dove per ogni fase vengono notificati i corrispondenti incaricati, ai quali viene richiesta la compilazione di *form* relativi alla non conformità di un prodotto e alle conseguenti azioni da intraprendere.

1.1.8 Angular

Gli ultimi requisiti emersi durante lo svolgimento dello *stage* sono quelli relativi allo studio e all'applicazione di alcune fasi di [DevOps](#) a progetti realizzati con lo strumento Angular.

Esso è un *framework* per la realizzazione di applicazioni *web* tramite il linguaggio di programmazione TypeScript e rappresenta una delle principali tecnologie usate dai *team* di sviluppo di Wintech.

È per questo motivo che, nel momento in cui ho terminato le attività previste per il mio *stage* con alcuni giorni di anticipo, mi è stato affidato il compito di studiare e provare ad utilizzare le stesse loro tecnologie e metodologie di lavoro al fine di comprenderle al meglio ed integrarmi maggiormente nell'ambiente lavorativo.

Tramite il sito *web* ufficiale di Angular, ho seguito le guide fornite in modo da apprendere le basi e poter realizzare quanto richiesto soddisfacendo i requisiti assegnati.

1.2 Progettazione

In questa sezione sono presenti tutte le attività progettuali da me svolte e il suo scopo è descrivere le modalità con cui ho individuato le soluzioni ai bisogni progettuali in modo da soddisfarne i requisiti.

Microsoft Solutions

1.3 Programmazione

In questa sezione sono presenti tutte le attività da me svolte al fine di sviluppare e implementare le soluzioni individuate in fase di progettazione.

1.4 Verifica e Validazione

In questa sezione sono presenti tutte le attività da me svolte al fine di verificare il corretto funzionamento delle soluzioni sviluppate e il loro soddisfacimento dei requisiti progettuali.

1.5 Risultati raggiunti

1.5.1 Qualitativamente

1.5.2 Quantitativamente

Glossario

Continuous Deployment (CD) L'adozione di questo approccio consente di rilasciare nuove versioni del *software* in modo rapido e frequente, garantendo che le funzionalità siano disponibili per gli utenti finali in tempi brevi. Inoltre, il *team* di sviluppo non è più obbligato ad interrompere lo sviluppo per prepararsi ed effettuare i rilasci. Questi ultimi sono meno rischiosi poiché le modifiche apportate al prodotto sono tipicamente contenute ed è quindi più agevole identificare eventuali problemi. Infine il cliente ha la possibilità di fornire *feedback* costantemente potendo verificare ogni avanzamento. [12](#)

Continuous Integration (CI) Ogni integrazione viene verificata automaticamente attraverso l'esecuzione di *test* per rilevare rapidamente eventuali errori o conflitti nel codice. Il concetto della *Continuous Integration* è stato originariamente proposto come contromisura preventiva per il problema dell'"*integration hell*", ovvero le difficoltà dell'integrazione di porzioni di *software* sviluppati in modo indipendente su lunghi periodi di tempo e che di conseguenza potrebbero essere significativamente divergenti. [12](#)

DevOps Metodologia che enfatizza l'automazione, la condivisione di responsabilità e il miglioramento continuo, utilizzando strumenti e processi che supportano la [Continuous Integration](#), il [Continuous Deployment](#) e il monitoraggio costante dei sistemi. [2](#), [6](#), [7](#), [9](#), [11](#)

HyperText Transfer Protocol (HTTP) Protocollo a livello applicativo, ovvero il livello più alto definito dal modello OSI (Open Systems Interconnection), il quale rappresenta uno *standard* architetturale per reti di calcolatori. Tale livello è responsabile della gestione delle comunicazioni tra applicazioni, fornendo i servizi necessari per lo scambio di dati strutturati e significativi tra *client* e *server*. [2](#), [5](#)

Sistemi Sistemi S.p.A. è una società italiana partecipata con Wintech S.p.A. Essa possiede tecnologie ed ambienti di sviluppo dedicati al fine di creare soluzioni *software* gestionali e servizi per professionisti e imprese, soprattutto in ambiti relativi a studi professionali di commercialisti, consulenti del lavoro e avvocati, imprese e associazioni di categoria. [2](#)

Bibliografia

Siti web consultati

Atlassian, Scrum. URL: <https://www.atlassian.com/it/agile/scrum>.

Documentazione Jenkins. URL: <https://www.jenkins.io/doc/>.

Documentazione Power Apps. URL: <https://learn.microsoft.com/en-us/power-apps/>.

Documentazione Power Automate. URL: <https://learn.microsoft.com/en-us/power-automate/>.

Sistemi. URL: <https://www.sistemi.com/chi-siamo/>.

Wikipedia. URL: <https://it.wikipedia.org/>.

Wintech. URL: <https://www.wintech.it/>.